A Reproduced Copy

OF

(NASA-CR-151940) ANALYSIS OF STATIC PRESSURE DATA FROM 1/12-SCALE MODEL OF THE YF-12A. VOLUME 3: THE MARK IVS SUPERSONIC-HYPERSONIC ARBITRARY BODY PROGRAM, USERS MANUAL Final Report (Douglas 00/02

N86-71341

Unclas
00677

Reproduced for NASA

by the

NASA Scientific and Technical Information Facility

FFNo 672 Aug 65

N-155610

NASA CONTRACTOR REPORT

NASA CR–151940

ANALYSIS OF STATIC PRESSURE DATA FROM 1/12-SCALE MODEL OF THE YF-12A

Volume III – The MARK IVS Supersonic – Hypersonic Arbitrary Body Program – Users Manual

By D. N. Smyth and H. C. Loo

October 1981

Disribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.

LIBRARY COPY

SEP 9 1985

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA

APR 1986
RECEIVED
NASA STI FACILITY
ACCESS DEPT.

Prepared Under Contract No. NAS2-7585

by

Douglas Aircraft Company
3855 Lakewood Blvd
Long Beach, CA 90846

for

AMES RESEARCH CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA CONTRACTOR REPORT

NASA CR-151940

ANALYSIS OF STATIC PRESSURE DATA FROM 1/12-SCALE MODEL OF THE YF-12A

VOLUME III - THE MARK IVS SUPERSONIC-HYPERSONIC ARBITRARY BODY PROGRAM - USERS MANUAL

D. N. Smyth

H. C. Loo

DOUGLAS AIRCRAFT COMPANY

MCDONNELL DOUGLAS CORPORATION

OCTOBER 1981

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

TABLE OF CONTENTS

		Page
SECTION 1	INTRODUCTION	1
SECTION 2	PROGRAM DESCRIPTION	3
SECTION 3	INPUT-DATA INSTRUCTIONS	7
	INPUT TO MAIN-EXECUTIVE ROUTINE	8
	Executive-Flag Card	8
	System-Control Card	8
	GEOMETRY DATA PREPARATION	9
	Geometry Nomenclature	9
	Geometry-Data-Preparation Methods	10
	Input to Geometry Executive Control and Quadrilateral-Calculation Routine	13
	Geometry-Control Card	13
	Panel-Identification Card	14
	Scale-Factor Card	18
	Input to the Element-Reading Routine	18
	Element-Control Card	19
	Element-Data-Input Cards	23
	Elliptical-Cross-Section Data	24
	Ellipse-Generation-Control Card	26
	Cross-Section-Data Cards	27
	Parametric-Cubic-Input Data	30
	Parametric-Cubic Title Card	31
	Parametric-Cubic Boundary Data Card	33
	Aircraft-Geometry Option	34
	Aircraft Geometry Identification and Control	39
	Title and Identification Card	39
	Control-Flag Card	41
	Wing-Area Card	44
	Wings	44
	Wing Percent-Chord-Location Cards	47
	Airfoil Leading-Edge Coordinate Cards	47
	Wing Camber-Line Cards	47
	Wing Airfoil-Ordinate Cards	48

TABLE OF CONTENTS (CONTD)

	Page
Fuselage	48
Fuselage X-Station Cards	50
Fuselage-Camber Cards	50
Fuselage Cross-Section-Area Cards	51
Fuselage Y-Coordinate Cards	51
Fuselage Z-Coordinate Cards	51
Pods or Nacelles	52
Pod-Origin Cards	53
Pod X-Station Cards	54
Pod-Radii Cards	54
Fins	55
Fin Leading-Edge-Coordinate Card	56
Fin Percent-Chord-Location Card	56
Fin Airfoil-Ordinate Card	56
Canards or Horizontal Tails	57
Canard Leading-Edge-Coordinate Card	58
Canard Percent-Chord-Location Card	58
Canard Upper-Ordinate Card	59
Canard Lower-Ordinate Card	59
General-Airfoil Surfaces	60
General-Airfoil Control-Flag Card	62
Airfoil Percent-Chord-Location Cards	63
Airfoil-Orientation Cards	63
Airfoil Camber-Distribution Cards	64
Airfoil Thickness-Distribution Cards	64
AERODYNAMIC PROGRAM — INPUT DATA	66
Input to Aero-Executive Routine	66
Aero-System Title and Control Card	66
Aero-Flag Card	68
Flight-Condition Card	68
Reference-Dimension Card	69
α-β Cards	70
Flow-Field Program-Input Data	70

TABLE OF CONTENTS (CONTD)

Page

Flow-Field Data 71

Master-Directory Title Card. 71

Master-Directory Data-Set Card 71

α-β Set Directory-Table Card 72

Region-Directory-Table Card. 72

Surface-Data-Transfer Option. 73

Surface-Data Normalization 74

Flow-Field Control and Plane-Orientation Card. . 77

Hand-Loaded Surface-Pressure-Coefficient-Data Cards 78

Surface-Data-Coordinate Card. 78

Surface-Data-Property Card. 79

Surface-Data Panel-Selection Card. 79

Pressure-Calculation Program Input Data 80

Title Card and Basic Flags 81

Component-Organization Card. 81

Pressure-Method Cards. 82

Input-Pressure Option. 83

Surface-Property Access-Flag Card 84

Normalization-Data Cards. 86

Special C_p Interpolation at Previously-Stored Coordinates 86

Panel-Number Card. 87

Row-Sequence-Number Card 87

Special C_p-Interpolation Input-Coordination Cards. 87

Special-Routines-Option-Input Data. 88

Special-Option Section Card. 88

Summation-Routine Option. 89

Summation-Control Cards 89

Shear Option. 91

Fuselage 92

Wing 92

Vertical and Ventral Surfaces. 93

Elevon Hinge Moments 93

TABLE OF CONTENTS (CONTD)

Page

Shear-Option-Input Cards 94
Shear-Component-Description Card. 95
Shear-Station-Definition Cards. 95
Shear-Panel-Number-Definition Cards 95
Elevon-Definition Cards 96
Hinge-Line Coordinate Card 97
Panel-Number Card. 97
Automated Input Assembler for AERO Option 98

SECTION 4 ILLUSTRATED EXAMPLE. 100
GEOMETRY DEFINITION. 100
PRESSURE INPUT, FORCE CALCULATION, AND SUMMATIONS. 103

SECTION 5 DATA STORAGE TECHNIQUES. 111
UNIT 4 GEOMETRY DATA. 112
UNIT 9 FORCE-DATA 116
UNIT 10 FLOW-FIELD-DATA. 118
UNIT 15 COORDINATES AND POINTERS FOR Cp DATA 121
UNIT 16 Cp DATA. 124
UNIT 20 CASE DATA-INTEGRATED FORCES, MOMENTS, AND SHEARS . 124
UNIT 21 SURFACE-SPLINE PRESSURE DISTRIBUTION DATA. 125

APPENDIX A ILLUSTRATED EXAMPLE GEOMETRY DEFINITION. 129
INPUT
OUTPUT

APPENDIX B ILLUSTRATED EXAMPLE PRESSURE INPUT, FORCE CALCULATION, AND SUMMATIONS 150
INPUT
OUTPUT

APPENDIX C OVERLAY STRUCTURE AND EXECUTION DECK SET UP. 221

APPENDIX D PROGRAM LISTINGS 224

REFERENCES 428

LIST OF FIGURES

		Page
1.	Functional Organization of the MARK IVS Program	4
2.	Geometry Definitions Diagram	10
3.	Geometry Input and Usage Flow Chart	12
4.	Input Geometry Coordinate System	20
5.	Use of Small Paper Model in Visualizing Geometry Loading Procedure	22
6.	Parametric Cubic Patch Input	30
7.	Aircraft Geometry Option Data Flow Logic	40
8.	Wing With Streamwise Airfoils	45
9.	Points and Elements Generated for Wing Surface	46
10.	Fuselage Geometry Generation	49
11.	Pod or Nacelle Geometry	52
12.	Fin Geometry Generation	55
13.	Generation of Canard and Horizontal-Tail Geometry	58
14.	General Airfoil Coordinate System	60
15.	Configuration 101A Geometry Panels	101
16.	Configuration 101A Three View	102
17.	Configuration 101A Flow Regions	104
18.	Flow-Field Data-Storage Format	119

LIST OF TABLES

1.	Configuration 101A Geometry Panel Characteristics	106
2.	Configuration 101A Input Pressures	107
3.	Correspondence Between Original MARK IV Calculations and MARK IVS Input Pressures	108

LIST OF TABLES (CONTD)

Page

4. Configuration 101A Force Components 109

5. Configuration 101A Summation Data Sets. 110

ANALYSIS OF STATIC PRESSURE DATA FROM 1/12-SCALE MODEL OF THE YF-12A

VOLUME III - THE MARK IVS SUPERSONIC-HYPERSONIC ARBITRARY-BODY PROGRAM - USERS MANUAL

By D.N. Smyth and H.C. Loo

SECTION 1. INTRODUCTION

The computer program described in this manual is identified as the MARK IVS Mod 0 version of the Supersonic-Hypersonic Arbitrary Body Computer Program and was prepared to fulfill the requirements of a NASA study. Specifically, the primary objective of the subject study was to obtain the vehicle components of force and moment from the pressure data measured on a 1/12-scale wind-tunnel model of the YF-12A airplane. In addition, the shear, bending moment, and torque at various stations on the fuselage, wing, and vertical surfaces, as well as the elevon hinge moments were to be determined.

The two major requirements to be accomplished in the preparation of this computer program were the capabilities to consider a very complex geometrical configuration and to analyze a large amount of pressure data (approximately 540 orifice locations on the model for more than 1,600 test conditions). The first task was uniquely met by an existing computer program - the MARK IV Supersonic-Hypersonic Arbitrary Body Computer Program. (References 1, 2, and 3). This program is capable of analyzing completely arbitrary configurations and already had some capability to calculate total vehicle forces and moments from input pressure distributions.

The MARK IV, then, was selected as the starting point for the new program. Emphasis was placed on improving the efficiency of the input-pressure option and meeting the special requirements of the present study. These included adding new routines for the shear and moment calculations, creating new storage files for subsequent use with plotting programs, and devising some means of automatically inputting the vast amount of experimental data to be analyzed. Originally, these procedures were added directly to the MARK IV program via new input flags and additional overlay links. However, this resulted in a large, logically complex computer program that could possibly inhibit its use in the future. A single purpose computer program - input pressure option only, was then prepared by stripping out all the aerodynamic predictive methods producing a physically smaller source deck and a logically simpler code. The final result is the MARK IVS, for "MARK IV Stripped" Program.

The primary purpose of this manual is to provide instructions for the use of the MARK IVS Program. It is intended that this manual be used in conjunction with the MARK IV documentation (References 1, 2, and 3) and familiarity with the MARK IV (or the earlier MARK III) is almost a prerequisite. The present manual is provided mainly as a convenience so that, once a user is knowledgable of the concept, the instructions for the input pressure option are readily available in a format unencumbered by all of the other MARK IV options. In

this regard, an attempt has been made to make the MARK IVS Manual a stand-alone document. Input descriptions are given for a general application and also for the automated procedure set up for the analysis of the YF-12A wind-tunnel data. An example case illustrating the general input for a simple configuration is also presented.

SECTION 2. PROGRAM DESCRIPTION

The MARK IVS retains all of the arbitrary-body geometry input capabilities of the original program. The only aerodynamic method available pertains to the input pressure option and has general applicability in all speed ranges - subsonic, supersonic, and hypersonic. The shear option (the calculation of shear, bending moment, and torque at various vehicle stations) in its present form is applicable only to the YF-12A configuration. An automatic input feature has been set up specifically for the YF-12A; however, the procedure is general and could be easily adapted to other experimental configurations. Several of the optional features (e.g., pressure distribution along various surface contours) were dictated by requirements of this study and Reference 4 should be consulted for a complete discussion of the analysis. Similarly, the Mark IV documentation (Reference 2) should be referred to for a description of the mathematical techniques employed, such as the definition of the surface elements or the formulation of the surface spline.

The underlying concept used in the MARK IVS is simply stated:

> The configuration is modelled by a number of plane quadrilateral elements. Incremental forces and moments are determined for each element and summed over all elements to obtain the vehicle forces and moments.

The outstanding feature of the program is the capability to handle completely arbitrary shapes independently of the locations for the measured pressures. The functional organization of the MARK IVS is shown in Figure 1. Program execution is controlled by an executive routine. The input to the executive routine provides the order in which the Geometry and Aero options are to be performed. These options, in turn, use a small executive routine or function to further control the selection of sub-options.

All geometry input, element generation and storage is accomplished in the Geometry component of the program. Subroutine GEOM is the executive controller of this option. Geometry data may be input with the use of four methods; Surface Elements, Elliptical Cross-Sections, Parametric Cubics, and the Aircraft Geometry Program. A given vehicle may be defined by a combination of the methods selected at the discretion of the user.

The GEOM subroutine converts the input data into quadrilateral elements and stores them along with the original input surface element corner point coordinates on Unit 4 using mass storage (random access) techniques. The organization of the data on Unit 4, and all the other mass storage units, are described in Section 5.

Many other parts of the program make use of the geometry data stored on Unit 4. The vehicle is normally divided into a number of panels with each panel composed of a number of elements. Each panel is identified by a storage sequence number. The manner in which these panels are to be grouped together, in the other parts of the program, is determined by direct reference to the panel sequence number.

The AERO component of the program contains four major options; Flow Field Analysis, Inviscid Pressure, Special Routines, and Shear. The Automated

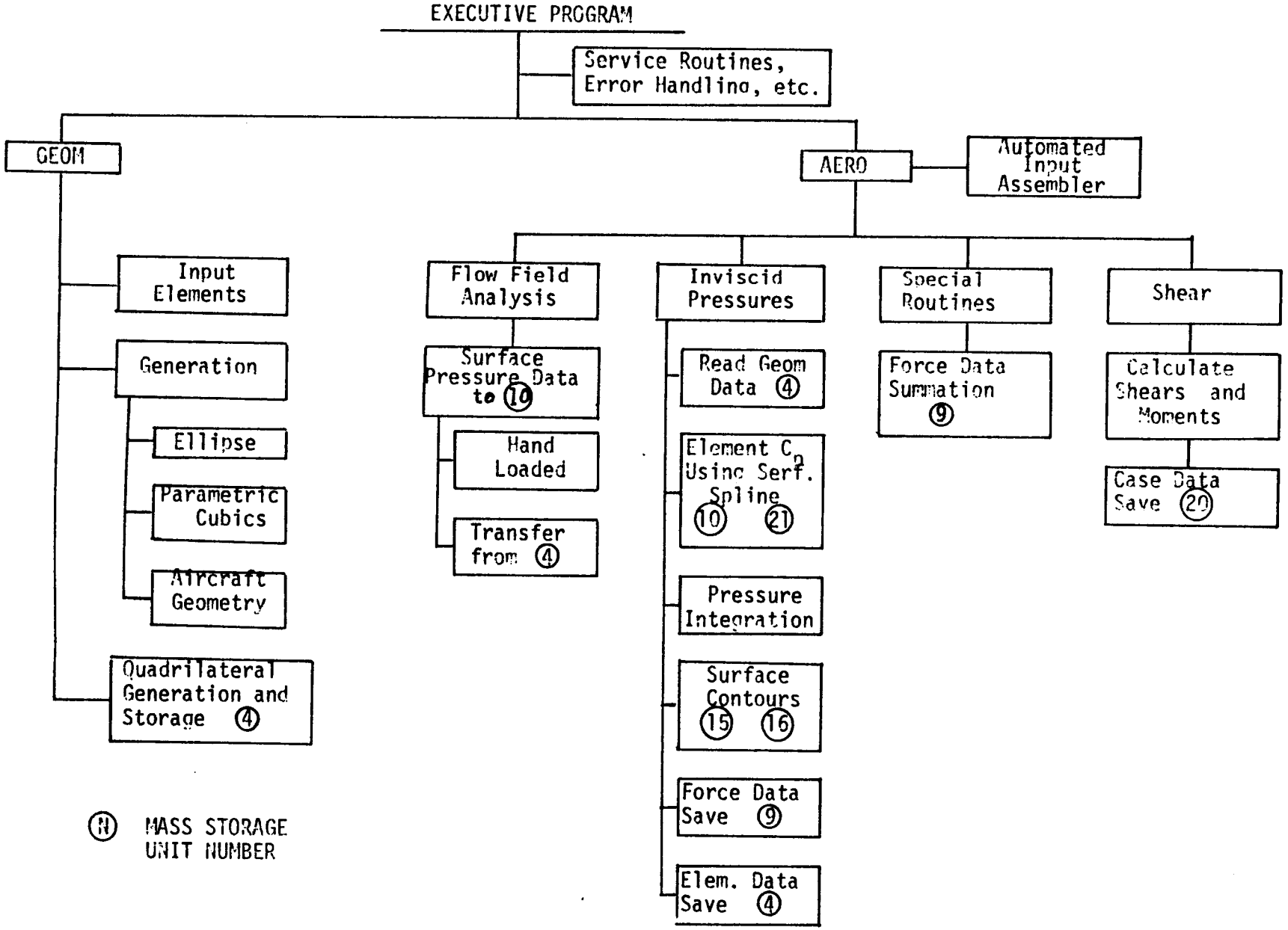


FIGURE 1. FUNCTIONAL ORGANIZATION OF THE MARK IVS PROGRAM

Input Assembler is accessed in the AERO component if requested. This is a convenience feature added to facilitate the handling of a large number of cases for a given configuration. The required MARK IVS input data are assembled from three auxiliary files (see Section 3).

The Flow Field Analysis Option provides the means for inputting the measured surface pressure coefficients. This data is collected in user defined flow regions and stored on Unit 10 using mass storage techniques. The surface pressure data may be either input directly using the Hand Input Option or they may be transferred from previously saved element data on Unit 4. The flow regions are defined over portions of the configuration for which the pressure variation is not extreme and normally with overlapping boundaries to provide continuity. Each region is identified by a storage sequence number and access in the Inviscid Pressure Option is by direct reference to the region sequence number.

The Inviscid Pressure part of the program calculates pressure and resultant forces on each quadrilateral element. The geometry data used are obtained from Unit 4 where they must have been previously stored by the Geometry Option. The geometry panels may be grouped to form a vehicle component and the flow regions to be considered are specified by their sequence number.

The flow region and surface pressure data must have been previously stored on Unit 10. A surface-spline fitting is performed over each region to obtain the pressure at the element centroids. A one-to-one correspondence between geometry panels and flow regions is <u>not</u> required. A given panel can encompass several regions and vice-versa. An option is available to save the coefficients of the spline fit for subsequent use. They are stored on Unit 21 using mass storage techniques.

Under the control of an input flag, the local flow properties on an element may also be stored back on Unit 4 for use in the Shear Option. The total force and moment coefficients for each vehicle component are determined and stored on Unit 9 (mass storage). A special option is also available for obtaining the pressure along specified contours on the vehicle surface. The contour coordinates must have been previously stored on Unit 15 and the calculated pressures may be stored on Unit 16 for subsequent access by a plotting program. Both units 15 and 16 use mass storage techniques.

The Force Data Summation Option under the Special Routines section may be used to selectively sum, print, and punch data that has been previously stored on Unit 9 by the inviscid pressure routines. The results of these summations may also be stored back on Unit 9 for use in subsequent summations. This is the only option provided in the program for combining the force contributions of the different vehicle panels or components. This feature permits the user to develop a component buildup of the aerodynamic characteristics (e.g., wing alone, wing plus body, wing plus body plus tail, etc.).

The Shear Option calculates the shear, bending moment, and torque at prescribed stations of the vehicle. The elevon hinge moments are also calculated. Use is made of the geometry data and element force data which must have been

previously stored on Unit 4. The shear and moment data, as well as the aerodynamic force data, may be stored on Unit 20 for subsequent use in a plotting routine. Unit 20 uses mass storage techniques and permits efficient access for various cross plots of the stored quantities.

SECTION 3. INPUT-DATA INSTRUCTIONS

This computer-program system uses a free-form approach to the preparation of the input data. That is, the order of the input cards depends upon the requirements of the problem being solved. This is true of both the system-control data and the input data to each of the major program components. For this reason, few standard-data load sheets are provided. In most cases the user will find that a simple 80-column load sheet can be used. This means that the cards can be written on the load sheet in the order in which they will be read into the program. All of the input data is read right at the beginning of the job and printed out on the output unit. In this way the user always has a record of the input data along with the output results of the program.

The Mark IVS Program requires an Executive-Flag Card and a System-Control Card. These cards are followed by the sets of data cards for each program option to be executed. Detailed card-by-card descriptions of all of the input items are provided in this section.

The general scheme used in describing the input data is shown below:

Column	Code	Routine Format	Explanation

The meaning of each of these headings is as follows:

Column - Indicates the position on the card for each data field.

Code - Gives the FORTRAN name used in the read statement by the program.

Routine - Indicates the subroutine where the data is read.

Format - Indicates the FORTRAN format of the data read statement. The parameter I4 would indicate that the parameter is an integer, right justified in a field that is 4 columns wide. The parameter F10.0 would indicate a fixed point number punched with a decimal point and sign, and placed anywhere in the card columns indicated.

Explanation - The description of the input-data parameters, flags, etc.

INPUT TO THE MAIN-EXECUTIVE ROUTINE

Only two input cards are required for the Executive Program.

Executive-flag card (3I1). -

This card must be the first card in the data deck.

Column	Code	Routine Format	Explanation
1	IEROR	Main I1	Automatic ABEND-Dump Flag. Used in IBM version of the program only. = 0 Any IHC type of error will cause an automatic OC4 type of System ABEND. This is useful in tracking down both program and input errors. = 1 Automatic ABEND capability is not turned on.
2	INMONT	Main I1	Input-Monitor-Control Flag. This flag causes all of the input data to be copied from the input unit (5) to Unit 1. As this is being done all the input cards (except for the Executive-Flag Card) are written out on the output unit (6). = 0 The monitor routine is called as is described above. = 1 The monitor routine is not called and the input data is not printed on Unit 6 or transferred to Unit 1.
3	INPUTD	Main I1	= 0 Will use automated input for AERO option The monitor routine will be called. = 1 Will use normal input for AERO option.

System-control card (20I1, 39X, A4). -

Column	Code	Routine Format	Explanation
1	IPG(1)	Main 20I1	System options in the order that they are to be executed.
2	IPG(2)		
3	IPG(3)		IPG Option = 1 Call GEOM. = 2 Call AERO.
	etc.		
20	IPG(20)		
60-63	CASE	Main A4	Case identification.

GEOMETRY-DATA PREPARATION

Geometry Nomenclature

Before proceeding with the detailed descriptions of the geometry-input data several (and often confused) geometry terms should be defined.

Surface Element:	This, the smallest geometry unit, consists of four related points on the surface of the vehicle and the area enclosed by lines connecting successive points. All geometry data must eventually be made available to the program in surface-element form.
Plane Quadrilateral Element:	Each surface element is converted by the program into a plane-quadrilateral element. The plane-quadrilateral element is the basic geometric unit used in the force calculations. This unit, in effect, is the integration-step size and is fixed once the surface-element representation of the shape is established.
Cross-Section Cut:	A cross-section cut is that view obtained by cutting the vehicle in the longitudinal plane (Z, Y plane), at a constant X-station.
Vehicle Section:	A vehicle section consists of an aggregation of surface elements that have similar size and proportions. On the Type 3 element-data cards a Section starts when a point with STATUS = 2 is found. A Section ends when the next STATUS = 2 (or 3) is found.
Vehicle Panel:	This parameter is used to indicate a part of the vehicle that is made up of several Sections. On the Type 3 element-data cards a Panel is ended when a STATUS = 3 is found.
Vehicle Component:	A Component is defined as a major part of the vehicle that is to be analyzed by the program as a unit (i.e., a wing, tail, etc.). A Component is usually made up of several Panels.

In this program the geometry data are stored on the Quadrilateral-Element-Storage Unit 4 by PANELS. That is, each PANEL is identified by a sequence number. In the AERO part of the program these PANELS may be grouped together to form vehicle COMPONENTS for the aerodynamic analysis. The PANELS are assigned consecutive-sequence numbers (ISTAT3) by the GEOMetry routine as they are stored on the Quadrilateral-Storage Unit 4. The user must keep track of the panel-sequence numbers for each Panel (each STATUS = 3) stored on Unit 4. These panel-sequence numbers are required as input to the other parts of the program to retrieve the proper geometry data for the force calculations.

These various definitions are illustrated in the following table and by Figure 2.

Basic Quantity	Combine	To Form	Examples
Points	→	Elements	
Elements	→	Sections	Inboard Section A, Flap Section B, outboard Section C.
Sections	→	Panels	Sections A, B, C form upper-wing panel.
Panels	→	Components	Upper and lower panels form wing component.
Components	→	Vehicle	Wing, fuselage, tail, etc. components form the complete vehicle.

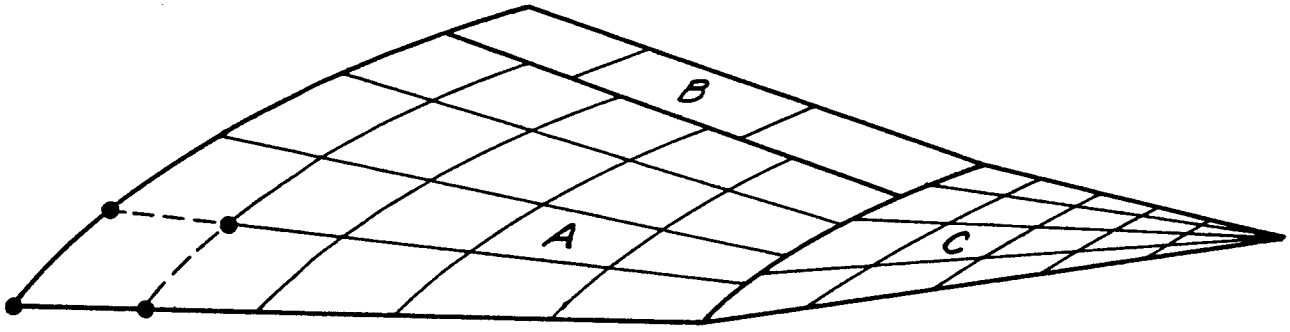


FIGURE 2. GEOMETRY DEFINITIONS DIAGRAM.

Geometry-Data-Preparation Methods

Several options are available for describing the geometric shape of a vehicle or component for use in this program. These methods are selected at the discretion of the user. A given vehicle may be defined by a combination of the methods. These methods permit the user to describe a completely arbitrary shape, or to synthesize a vehicle from simple component parts, depending upon the requirements of the problem. The general techniques used in achieving this are outlined below.

In the Mark IVS Program, the geometry data are input using one of the following methods:

1. Surface-Element Method (Distributed Elements). This method uses a large number of surface-coordinate points that are related to groups of 4 to form a surface element. The program then converts the area defined by each set of 4 points into a plane-quadrilateral element. All types of input-geometry data eventually end up in this form before actual force computations are made.

2. Elliptical-Cross-Section Data. In this method input data consist of the necessary radii, circle or ellipse center, and the sector of the circle or ellipse to be used. The program then converts these data into exactly the same surface-element form as described in the above paragraph. These data created by the program are in the correct format for conversion to plane quadrilaterals for subsequent calculations. The element-geometry data generated in this method are stored on the Geometry-Data-Storage Unit (8).

3. Mathematical-Surface Fit (Parametric Cubics). The surface-input data for this method consist of coordinates of points along the four boundaries of a patch. The coefficients for a mathematical surface-fit equation are then calculated to provide a description of the interior surface patch. This surface is then converted to exactly the same form as in Method 1 by a systematic variation of the two parametric parameters. Again, as in Method 2 above, the resulting element data are saved on the Geometry-Storage Unit for subsequent conversion to plane quadrilaterals.

4. Aircraft-Geometry-Generation Program. This program is designed to simplify the loading of aircraft types of configurations (i.e., wings, tails, nacelles, etc.).

The selection of the method to be used in describing a shape depends upon the detailed requirements of the problem and the vehicle shape. For completely arbitrary shapes either the Surface-Element or the Parametric-Cubic Method would be used. For simple shapes, such as a vehicle nose, leading edge, or circular or elliptical cross section, the Elliptical-Surface Generation Method (Method 2) would be used. For a vehicle synthesized from a number of simple components the geometry data can be generated by a separate program.

The Geometry Option will normally be the first option called (unless geometry data has been previously assembled and stored in quadrilateral form in the computer on a previous run).

The GEOM Program has two basic tasks: It provides for the input or generation of geometry data in ELEMENT form (TYPE3 cards), and for the conversion of these data to QUADRILATERAL form (surface area, outward normals, centroids, etc.). The ELEMENT cards when generated are usually stored on Unit IOUT (=8 for most cases). The QUADRILATERAL data are stored on Unit 4 using Mass-Storage techniques for the CDC version and Direct Access for the IBM version.

A simplified flow chart of how the geometry data are read in and used is presented in Figure 3.

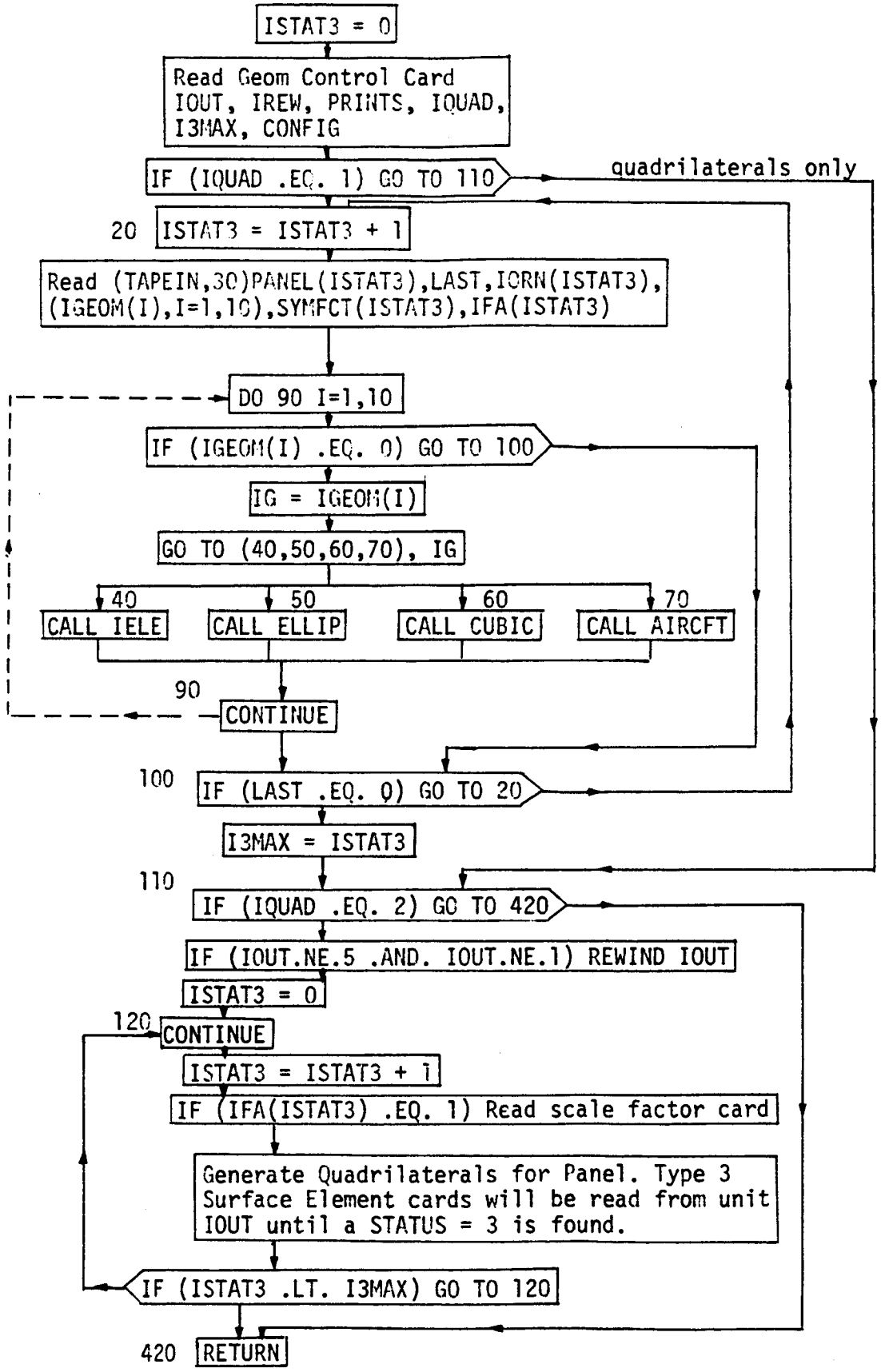


Figure 3 . GEOMETRY INPUT AND USAGE FLOW CHART

Input To Geometry Executive Control And Quadrilateral-Calculation Routine

Geometry-control card (1I2, 5I1, 1I2, 15A4). -

This is the first card for the GEOM option.

Column	Code	Routine Format	Explanation
1-2	IOUT	GEOM I2	Output-Storage Unit for Type 3 Element-Data Cards when they are generated by one of the Geometry-Generation routines or loaded by the Element-Load routine. Usually input = 8, but it may also be convenient to use it = 1 (or = 5 if INMONT = 1 on the Executive-Flag Card). If quadrilaterals are not to be calculated IOUT may be input = 7 to obtain a punched deck of the Type 3 element cards.
3	IREW	GEOM I1	Rewind Flag for Unit IOUT. = 0 Rewind Unit IOUT before storing any Type 3 cards on it. = 1 Do not rewind Unit IOUT.
4	PRINTS	GEOM I1	Print Flag for detailed quadrilateral characteristics. = 0 Do not print. = 1 Print.
5	IQUAD	GEOM I1	Quadrilateral-Calculation Flag. = 0 Program will expect to use the Geometry-Generation or Storage routines to put elements on to Unit IOUT. A Panel-Identification Card will be next in the deck. = 1 Program will by-pass the Element-Generation and Storage Options. A Panel-Identification Card will not be read. Program will read Type 3 geometry cards from Unit IOUT and convert them to quadrilaterals. = 2 Same as = 0 above except that the program will not calculate the quadrilaterals. A return from the GEOM routine will be called after the geometry generation or storage on Unit IOUT is completed.

Geometry-control card (Continued)

Column	Code	Routine Format	Explanation
6	I3MAX	GEOM I1	Number of Status 3's in the element Unit IOUT. Used in determining when the end of the quadrilateral computations are reached (used only when IQUAD = 1). When IQUAD ≠ 1 I3MAX is determined by the program.
7	NEW	GEOM I1	New Data-Set Flag. = 0 This is a new Data-Set Unit for geometry data. Set up all flags and pointers. = 1 This is not a new data set. Use old flags and pointers to store additional panels. = 2 Replace an existing panel.
8-9	NPMAX	GEOM I2	Maximum number of panels to be provided for on the new Geometry-Data Unit. Used only when NEW = 0. If input as = 0 then the program will set NPMAX = 50. When NEW = 2, NPMAX is the number of the panel to be replaced.
10-69	CONFIG	GEOM 15A4	Configuration identification to be written on the first record of the Geometry-Storage Unit (4).

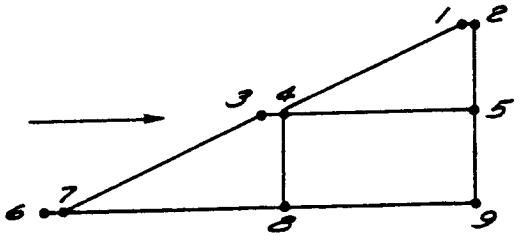
Panel-identification card (A4, 2I1, 10I1, 2I1, 4I2). -

This card is required if IQUAD ≠ 1. This card is used to control the flow to the various Geometry-Generation routines. One of these cards is required before each entry into the Geometry-Generation DO loop. The Geometry-Generation DO loop is terminated by the LAST parameter on this card. Each Panel-Identification Card can cause the Geometry-Generation routines to be entered up to 10 times. The number of times that the Geometry-Generation routines are to be entered is controlled by the IGEOM(I) parameter. When IGEOM(I) = 0 the DO loop is stopped. If IGEOM(1) = 0 the DO loop will not even be started but the Panel-Identification Cards will be read in until LAST = 1.

Column	Code	Routine Format	Explanation
1-4	PANEL(I)	GEOM A4	Panel number or code identification. This identification will be stored on the Quadrilateral-Data Storage Unit 4.

Panel-identification card (Continued)

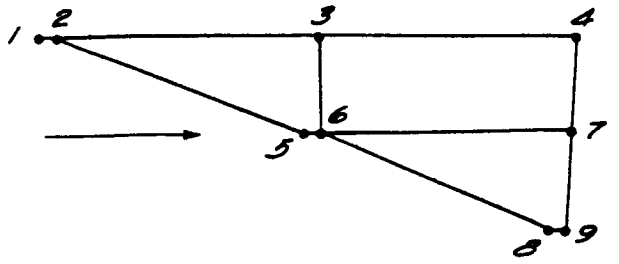
Column	Code	Routine Format	Explanation
5	LAST	GEOM I1	Last-Panel Flag. = 0 This is not the last Panel-Identification Card. = 1 This is the last Panel-Identification Card.
6	IORN(I)	GEOM I1	Geometry-Orientation Flag. = 0 Normal mode using cross-sections. = 1 Geometry-data input in streamwise strips. = 2 Geometry-data input in streamwise strips, but for each streamwise strip of elements the first coordinate point in the right-hand strip of points is not used in the formation of the leading-edge element, but is ignored by the program. This is illustrated in the diagram below for the lower wing of a vehicle. The streamwise direction is indicated by the arrow.

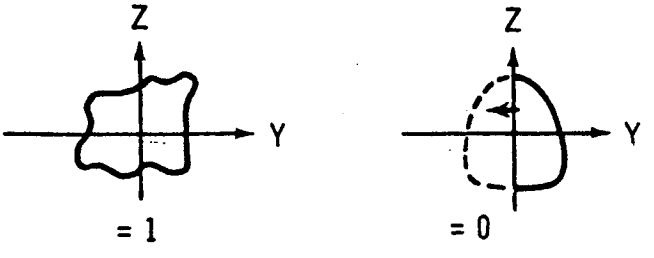


In the diagram above the numbers indicate the order of the input points. Note that points 1 and 2 are duplicate points (same values for X, Y, and Z). The first element is formed by points 1-2-4-5 and point 3 is ignored. In a similar manner, an element is formed by points 3-4-7-8 and point 6 is not used.

= 3 Same as =2 above except that the left point is ignored in the formation of the leading-edge elements. This would be useful for upper surfaces of a delta wing. The input schematic for this case is shown below.

Panel-identification card (Continued)

Column	Code	Routine Format	Explanation
			
7 8 9 10 etc.	IGEOM(I)	GEOM 10I1	Geometry-Generation-Method Flag. = 0 The Geometry-Generation routines (IELE, ELLIP, CUBIC, AIRCFT) will not be called. If LAST = 0 another Panel-Identification Card will be expected next. = 1 Routine IELE will be called to read Type 3 cards from the input unit and to copy these cards onto the Geometry-Storage Unit, IOUT. = 2 Routine ELLIP will be called to generate elements using the ellipse-generation techniques. = 3 Routine CUBIC will be called to generate elements using the Parametric-Cubic Method. = 4 Routine AIRCFT will be called to generate elements using the Aircraft-Geometry Option.
17	SYMFCT(I)	GEOM I1	Symmetry Flag. This flag indicates the type of vehicle symmetry to be used for this component of the vehicle (see diagrams below).



Panel-identification card (Continued)

Column	Code	Routine Format	Explanation
			Note: Although it is possible to use different Symmetry Flags for different components of a vehicle, the safe thing to do is to input or generate the geometry using the same Symmetry Flag for all parts of a vehicle. This is particularly important if pictures are to be drawn with a computer-graphics picture-drawing program.
18	IFA(I)	GEOM I1	Scale-Factor Flag. This flag permits the alteration of geometry data either by a shift of the reference coordinate system or by a multiplying factor. = 0 Use input-geometry coordinates (no scale factors will be used). The Scale-Factor Card will not be input. = 1 Use scale factors to scale and shift the geometry data in the basic coordinate system. Scale factors and coordinate increments are applied as the geometry data are being converted into quadrilateral data. The original element data on Unit IOUT are not altered. i.e. $X_{new} = X_{input} \cdot (XSC) + DELX$ etc. for Y and Z Note: The Scale-Factor Card is input after all of the Panel-Identification Cards and Geometry-Generation Cards (for IELE, ELLIP, CUBIC, AIRCFT) are input, and are the last cards read before the quadrilaterals are calculated.
19-20	NADJ1	GEOM I2	The number of some other panel that is adjacent to side-number 1 of this panel. The side-numbering convention is the same as was used to identify the sides of a single element.
21-22	NADJ2	GEOM I2	The number of some other panel that is adjacent to side-number 2 of this panel.

Panel-identification card (Continued)

Column	Code	Routine Format	Explanation
23-24	NADJ3	GEOM I2	The number of some other panel that is adjacent to side-number 3 of this panel.
25-26	NADJ4	GEOM I2	The number of some other panel that is adjacent to side-number 4 of this panel.

Scale-factor card (6F10.0). -

This card is input only if IFA = 1. One Scale-Factor Card is required for each Panel-Identification Card that has IFA = 1. All of the required Scale-Factor Cards are grouped together and input behind all of the Geometry Input or Generation Cards on each entry into the Geometry Option. The Scale-Factor Cards are read in one at a time as they are required (as specified by the IFA parameters on each Panel-Identification Card). This read occurs in the cycle of the quadrilateral calculations and the scale factors are applied to the input-geometry data as they are being converted into quadrilaterals. The original element data (Type 3 Cards) stored on Unit 8 is not affected by these scale factors as the scale factors are only used as the quadrilaterals are being generated (See Figure 3).

Column	Code	Routine Format	Explanation
1-10	XSC	GEOM F10.0	X Scale Factor to be multiplied times X_{input}.
11-20	YSC	GEOM F10.0	Y Scale Factor to be multiplied times Y_{input}.
21-30	ZSC	GEOM F10.0	Z Scale Factor to be multiplied times Z_{input}.
31-40	DELX	GEOM F10.0	ΔX Scale - Increment to be added to X.
41-50	DELY	GEOM F10.0	ΔY Scale - Increment to be added to Y.
51-60	DELZ	GEOM F10.0	ΔZ Scale - Increment to be added to Z.

Input To The Element-Reading Routine

This geometry option is used to transfer Element-Data Cards (Type 3 Cards) from the input unit (usually 5 or the Input-Monitor Storage Unit 1) to the Geometry-Element-Data Storage Unit (8).

Element-control card (2I2, 2I1). -

Column	Code	Routine Format	Explanation
1-2	I3MAX	IELE I2	The number of STATUS 3's that will be read before the IELE routine will stop and return to the Main-Geometry Program.
3-4	IN	IELE I2	Input unit for the Type 3 Cards. If input = 0 the program will set IN equal to TAPEIN as defined in the Main-Executive Program (= 1 or = 5).
5	IREW	IELE I1	Rewind Flag for Unit IN. The rewind control statement in IELE before the geometry read starts is as follows. IF (IN.NE.5 .AND. IREW.EQ.1) REWIND IN. = 0 Do not rewind Unit IN. = 1 If IN ≠ 5 rewind IN.
6	I3	IELE I1	Status 3 Control Flag. = 0 As the element cards are copied over to Unit IOUT all of the STATUS 3's will be removed except the last one. = 1 All of the STATUS 3's will be removed as the cards are copied to Unit IOUT.

Element-Data-Cards (Type 3 Cards) are input following the above Element-Control Card.

Surface-Element Input Data (Distributed-Element Method)

The important result of this general approach to the geometry problem is that the force-calculation part of the program is not affected by the method used to input the geometric shape. The form of the geometry data can be varied to meet the situation.

The coordinate system used for all the geometry data is shown in Figure 4. For symmetrical vehicles it is standard practice to input the left side of the vehicle only.

Since all of the geometry options finally produce geometry data in surface-element form, it is important that the methods and nomenclature used with this method be clearly understood. It is, therefore, recommended that the input instructions for the Surface-Element Method be studied before an attempt is made to use either the Ellipse or the Parametric-Cubic Options.

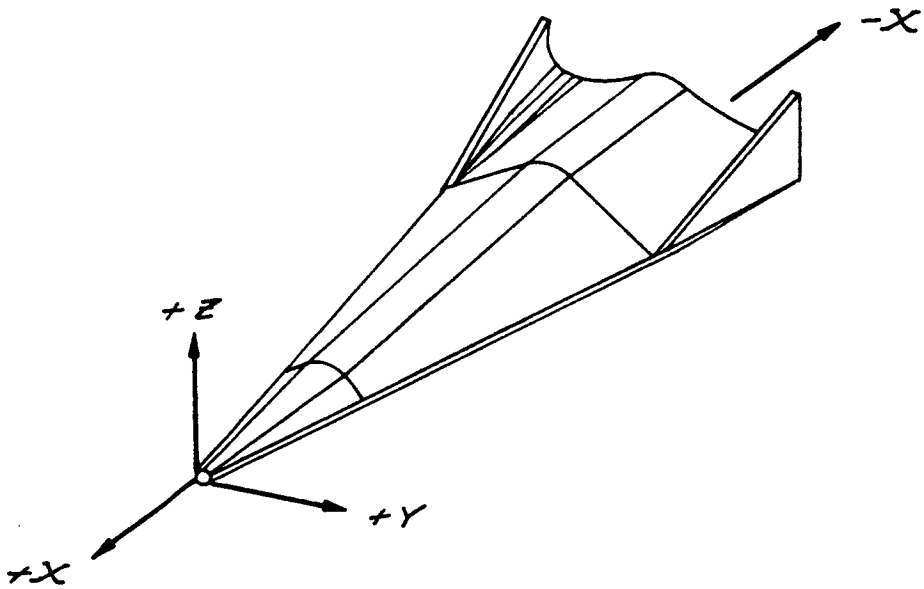


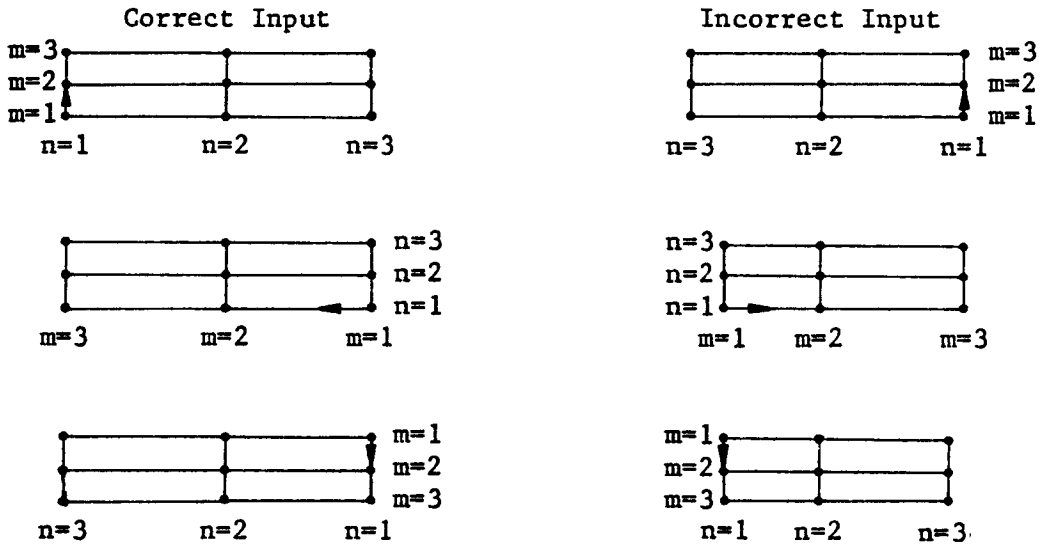
FIGURE 4. INPUT GEOMETRY COORDINATE SYSTEM.

The geometric-input data in this method include the coordinates of a large number of points on the vehicle surface. The input data are organized in a manner that permits the description of a vehicle on a component-buildup basis. This gives increased flexibility in shape description and makes it possible to use different force-calculation methods for different components. Because of possible changes in the surface contours of a component, it may also be necessary to divide the component into several sections. Each section of a vehicle component is further divided into a number of small units called elements, each defined by four points in space. In practice, the surface coordinates are usually recorded from cross-section drawings of the vehicle in such a way that each point need be read only once (even though it may be a member of as many as four adjacent elements). Each point is defined by its three coordinates and a Status flag that indicates whether it is the first point of a new section, a continuation of a column of points, the beginning of a new column, or the last point of the vehicle. The program uses the Status flags to determine how the input points are to be related to form elements, and how the elements are combined to form a section.

The first question that the user asks when starting to load the element geometry is, "In what order do I enter the surface points?" The basic rules to be followed are given below. These will be followed by a discussion of a visual technique that many users will find helpful in determining the proper loading order.

For the purpose of organizing the input data for computation, each point is assigned a pair of integers, m and n. The integers are not actually input to the program (they are calculated internally) but their use in

the following discussion will provide a better understanding of the input-data organization. For each point, n identifies the "column" of points to which it belongs, and m identifies its position in the "column", i.e., the "row". The first point of a "column" always has m = 1. To ensure that the program will compute outward-normal vectors, the following condition for the order of input points must be satisfied. If an observer is located in the flow and is oriented so that locally he sees points on the surface with m values increasing upward, he must also see n values increasing toward the right. Strict adherence to this simple rule will always lead to a correct set of input-geometry data. Examples of correct and incorrect input are shown in the sketches below. In these pictures the flow field lies above the paper, and the interior of the body lies below the paper. The arrows indicate the order of reading the points.



Associated with each input point is an input quantity called its Status. The first point of each new section has Status = 2. Except for the first n-line of a section, the first point of each n-line has Status 1. The last point of the component of the vehicle has Status 3. All other points have Status = 0; i.e., they may be left blank on the input sheet. The program will not exit properly from the surface-data subprogram and into the force-calculation phase until it reads a Status = 3.

The simple visual technique described below is helpful in determining the proper order of the input points.

1. First, assume that you are holding in your hand a small model of the vehicle shape. Many program users find it helpful to construct a small paper model to help in visualizing the geometry-loading procedure. On this model we will draw lines to represent the elements to be loaded for a given vehicle section. This process is illustrated in the photographs in Figure 5.

2. Next, decide which strips of elements are to constitute "columns" and which "rows". In most problems one of two procedures is selected - either a "column" of elements starts at the bottom of the

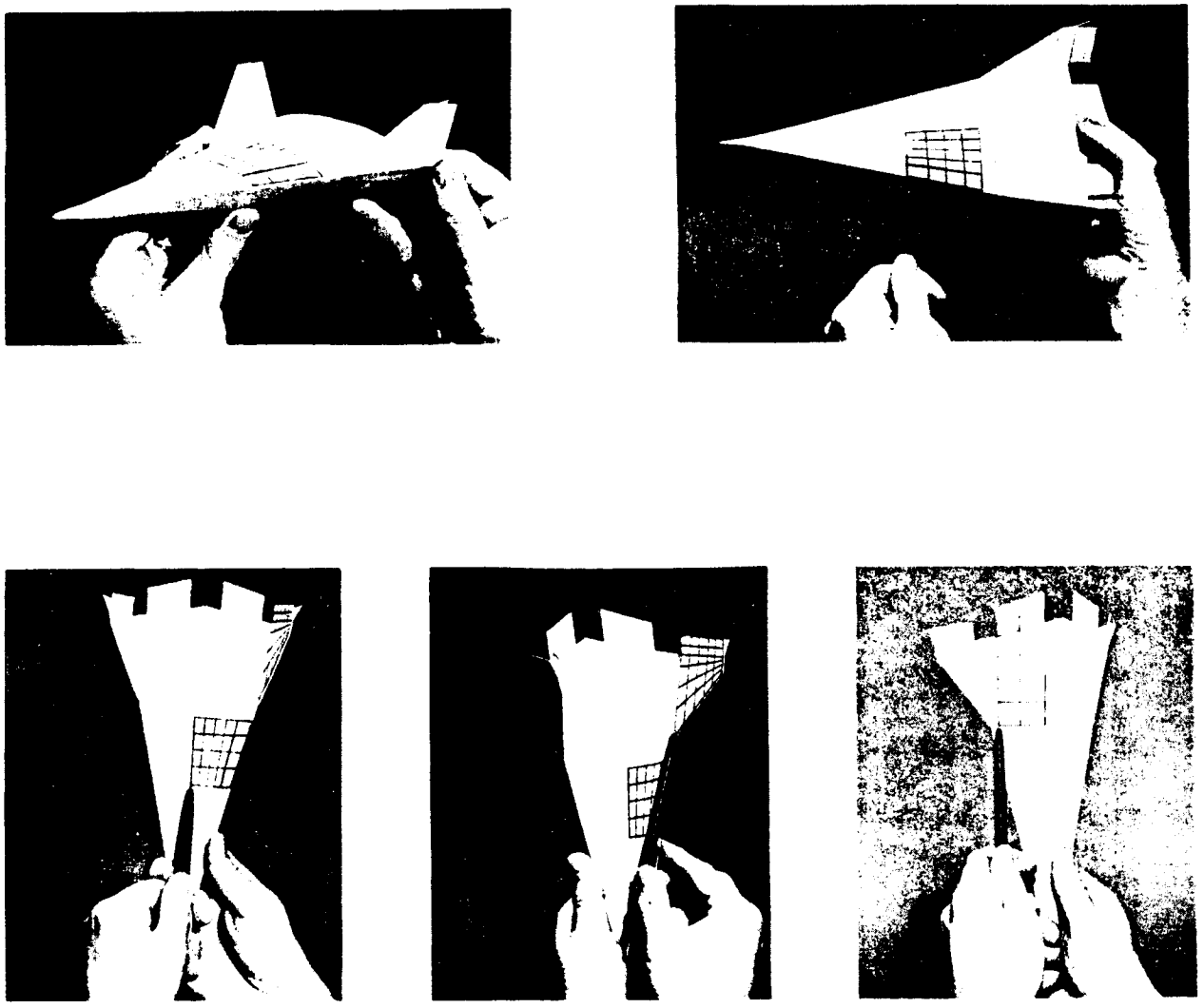


FIGURE 5. USE OF SMALL PAPER MODEL IN VISUALIZING GEOMETRY LOADING PROCEDURE. (In each example the pen is pointing at the first point to be loaded and in the direction of the first column of input points.)

shape and continues around to the top, roughly following vehicle cross-section lines, or a "column" is oriented so that it starts at the front part of the vehicle and runs aft toward the rear.

3. Hold the model out in front of you and rotate it until the columns are vertical with the first row of elements at the bottom. This procedure should be used regardless of what part of the vehicle is being loaded - the body, fin, inside of fin, etc. Always orientate the model so that you are looking at the section to be loaded (from the outside, looking at the surface) with the columns running vertical and the rows running horizontal.

4. Now that you have the section being loaded oriented in front of you, with the columns vertical, apply the following cardinal geometry rule:

   If a column of data points are loaded from the bottom to the top, then the next column of points (starting with a Status = 1) must be to the right.

All of the geometric-input data for this geometry option are input on Type 3 Element-Data Cards. Each card contains the X, Y, Z coordinates and Status flag for two points on the body surface. Every card in the element-geometry deck must contain two surface points except the last card, which may have only the first-surface-point coordinates and Status filled in. If a particular line of vehicle points is odd in number then it is usually advisable to repeat the last point (a dummy point) so that the last card will have two sets of point data. This permits the shifting of vehicle sections of the deck without disrupting other sections.

Element-data-input cards (3F10.0, I1, 3F10.0, I1, 2X, I2, 1A4, I2, 4X, I4). -

The detailed description of the input data for the Surface-Element Method is presented below.

Column	Code	Routine Format	Explanation
1-10	X	IELE F10.0	X-coordinate of surface point (the value of X is written anywhere in this space with a decimal point and sign; usually input only if it is negative).
11-20	Y	IELE F10.0	Y-coordinate of surface point.
21-30	Z	IELE F10.0	Z-coordinate of surface point.
31	STAT	IELE I1	Status flag for the above set of coordinates (= 2, 1, 0 or 3).

Element-data-input cards (Continued)

Column	Code	Routine Format	Explanation
32-41	XX	IELE F10.0	X-coordinate of surface point.
42-51	YY	IELE F10.0	Y-coordinate of surface point.
52-61	ZZ	IELE F10.0	Z-coordinate of surface point.
62	STATT	IELE I1	Status flag for the above set of coordinates (= 2, 1, 0, or 3).
65-66	CASE	IELE I2	Case number (right-justified integer).
67-70	SECT	IELE 1A4	Numbers or letters to identify the vehicle section. These must be legal machine characters.
71-72	TYPE	IELE I2	Card type number =03.
77-80	SEQ	IELE I4	Card-sequence number. This number is used to identify each card of a particular section and to aid in keeping the cards in order (right-justified integer).

A data load sheet for the above data is shown on the next page.

Elliptical-Cross-Section Data

This geometry option provides the capability of generating geometric data for vehicle components having whole or partial circular or elliptical cross sections with a minimum amount of input information required. This option is usually used to generate hemispherical noses and wing and tail leading edges.

The data generated by this option is saved on the Geometry-Storage Unit (Unit 8) in normal surface-element input-data form. In this manner it is possible to describe a vehicle with a combination of both hand-input data (in surface-element or parametric-cubic input form) and analytically derived circular or elliptical cross-section data.

The input data for this geometry option is described below. The input procedure is to define the basic properties of a circular or elliptic cross section (a cut in the Z-Y plane with X being the constant for the cross section). Each cross section where a set of element data are desired must

ELEMENT DATA

MARK IVS

ENGINEER: _____

DATE: _____

VEHICLE COMPONENT _____

DIRECTIONS FOR KEYPUNCH
PUNCH IN ALL CARDS-
DO NOT PUNCH
BLANK COLUMNS

CASE	SECT.	TYPE	
		3	AERO
66	70	72	76

±	X	±	Y	±	Z	STATUS	±	X	±	Y	±	Z	STATUS		SEQ. NO.
1	2 3 4 5 6 7 8 9 10	11	12 13 14 15 16 17 18 19 20	21	22 23 24 25 26 27 28 29 30	31	32	33 34 35 36 37 38 39 40 41	42	43 44 45 46 47 48 49 50 51	52	53 54 55 56 57 58 59 60 61	62		77 78 79 80

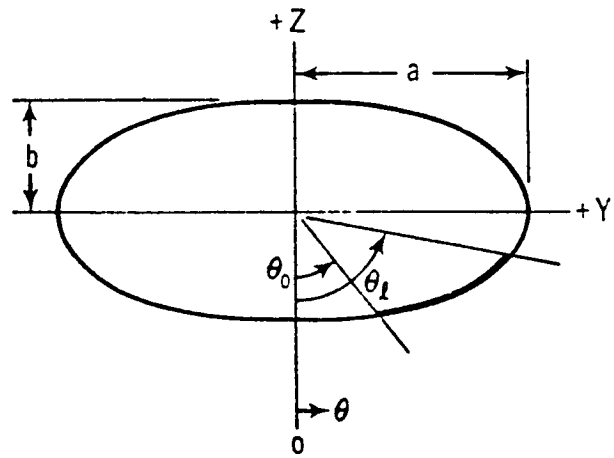
DIRECTIONS FOR KEYPUNCH: DO NOT PUNCH BLANK COLUMNS

NO UNDERPUNCHES IN SIGN FIELDS.

be input in this manner. The first cross section must be toward the front of the vehicle, and each succeeding section must be toward the rear.

Ellipse-generation-control card (12A4, 11X, 2I1, 3XI2, 1A4, I2). -

Column	Code	Routine Format	Explanation
1-48	TITLE	ELLIP 12A4	Vehicle section or component title. Any acceptable machine characters.
60	DISCON	ELLIP I1	Angular-Data Option Flag. This flag controls the angular division of the cross section and the dummy points generated to give complete card output for the Geometry-Storage Unit. See sketch below.



The angular-data options are given below.

= 1 All initial angles, θ_o, and all final angles, θ_ℓ, are the same for each cross section for this section of the vehicle.

= 2 All θ_ℓ in the vehicle section are the same but the θ_o varies.

= 3 All θ_o in the vehicle section are the same but the θ_ℓ varies.

61	IPRINT	ELLIP I1	Print Flag. This flag controls the printing of the element data generated in this option. This data printout will contain the exact information written on the Geometry-Storage Unit.

Ellipse-generation-control card (Continued)

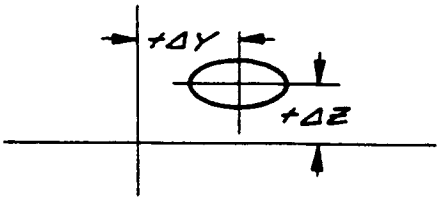
Column	Code	Routine Format	Explanation
			= 0 Do not print data.
			= 1 Print.
65-66	CASE	ELLIP I2	Case number. A right-justified integer used to identify the vehicle data.
67-70	SECT	ELLIP 1A4	Section identification. A number or letter used to identify this section or component of the vehicle. Any acceptable machine characters.
71-72	TYPE	ELLIP I2	Card Type number =04 (integer).

Cross-section-data cards (F10.0, 2F6.0, I3, 2F10.0, 2F7.0, I1, 4XI2, 1A4, I2).

One card for each cross-section cut desired.

Column	Code	Routine Format	Explanation
1-10	X	ELLIP F10.0	X-station (usually negative if the vehicle nose is at the coordinate-system origin).
11-16	THETO	ELLIP F6.0	Initial angle, θ_o, Degrees.
17-22	THETL	F6.0	Final angle, θ_ℓ, Degrees.
23-25	NN	ELLIP I3	Number of divisions of cross section desired. This number controls the number and spacing of the elements generated between θ_o and θ_ℓ (Right-justified integer).
26-35	A	ELLIP F10.0	Ellipse radius along the Y - axis, a.
36-45	B	F10.0	Ellipse radius along the Z - axis, b.
46-52	DELZ	F7.0	Offset of center of ellipse in the Z- direction, ΔZ.

Cross-section-data cards (Continued)

Column	Code	Routine Format	Explanation
53-59	DELY	ELLIP F7.0	Offset of center of ellipse in the Y-direction, ΔY. 
60	LAST	ELLIP I1	Last Flag. This flag controls the Status Flag (STATT) of the last element point generated and the position of the Geometry-Data Storage Unit (8). after the element data has been written on it. = 0 This is not the last cross section; set STATT = 0 and read in new cross-section card. = 1 Not active. Do not use. = 2 This is the last cross section for this vehicle section or component. Set the Status Flag STATT = 0, and read in a new ellipse-data title card. = 3 Not active. Do not use. = 4 This is the last cross section; no more sections are given, set last STATT = 3, write end of file on geometry unit.
65-66	CASE	ELLIP I2	Case number (right-justified integer).
67-70	SECT	ELLIP 1A4	Numbers or letters to identify the vehicle panel. These must be legal machine characters.
71-72	ITYPE	ELLIP I2	Card type number = 05 (integer).
77-80			Card sequence number. Not read by program.

A data load sheet for the above data is shown on the next page.

ELLIPSE GENERATION DATA

MARK IVS

ENGINEER: ______________________ DATE: ________

DIRECTIONS FOR KEYPUNCH	CASE (66)	SECT. (70)	(73–76)
PUNCH IN ALL CARDS DO NOT PUNCH BLANK COLUMNS			AERΦ

ELLIPSE GENERATION CONTROL CARD

COMPONENT TITLE (1–48)	(49–59)	ANGLE FLAG (60)	PRINT (61)	TYPE (72)	SEQ. NO. (77–80)
	//////////			4	

X-STATION (1)	θ_o (11)	θ_f (17)	NO. θ (25)	a (26)	b (36)	ΔZ (46)	ΔY (53)	LAST (60)	TYPE (72)	SEQ. NO. (77–80)
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	
									5	

Parametric-Cubic-Input Data

This geometry-input option is provided as an alternate input method in the description of arbitrary shapes. In this respect, it serves the same purpose as the Surface-Element-Input Method.

In the Surface-Element-Input Method a vehicle section is described by a large number of surface points organized in an element fashion. In the Parametric-Cubic Method only points along the boundaries of a patch are input to the program and the distributed surface points (surface elements) required for the subsequent quadrilateral calculations are determined by the program.

The basic features of this method are that: (1) fewer input points are required to describe a shape; (2) the input of this data is a little more complicated; and (3) the generated element size is controlled by two input parameters and may be changed to meet the requirements of the problem.

The input data consist of points along the four boundaries of a patch. The program calculates the coefficients for a mathematical surface-fit equation to provide a description of the interior surface of the patch. This surface is then converted into exactly the same form as the normal surface-element input data for further calculations. The element data generated is saved on the Geometry-Storage Unit (Unit 8) for use in other phases of the program.

Figure 6 illustrates how a section is described by this method.

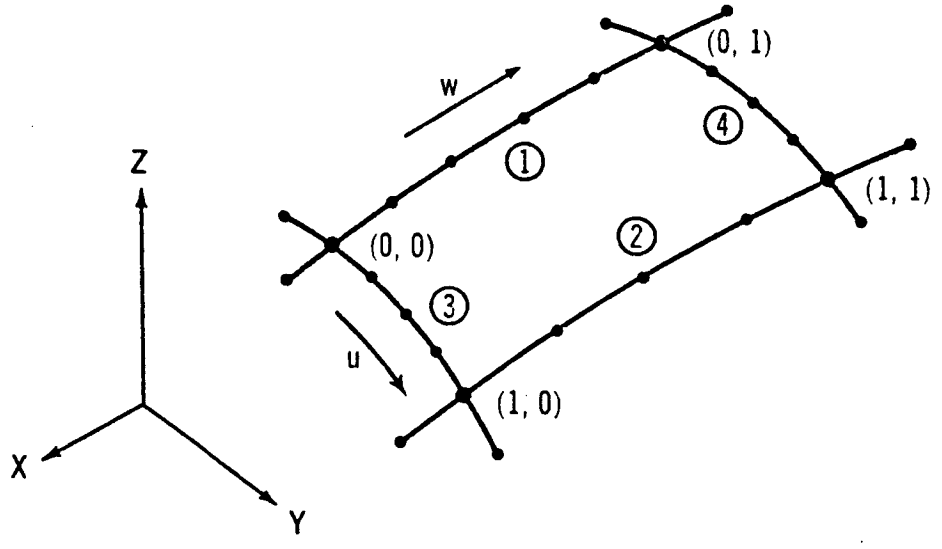


FIGURE 6 . PARAMETRIC CUBIC PATCH INPUT.

Each of the four boundaries is identified in Figure 6 by a number inside a circle. The input data for each of these boundaries must be input in the order indicated by these numbers, i.e., boundaries 1, 2, 3, and 4. The order of input points on a boundary and the order of the boundaries is important. The approach to ensure a correct input of the data is similar to that used

for the quadrilateral-element-input data. First, the user should imagine that he is holding a small model of the vehicle in hand. The vehicle is divided into a number of sections or patches. Figure 6 represents one such patch. The objective here is to describe how the data for one patch is loaded into the program.

The user orientates the model of the vehicle so that the number 1 boundary is to the left and the number 2 boundary to the right. Coordinates of points along the number 1 boundary are loaded first. The order of these points (from the user's view of the model) is from the bottom to the top of the patch. Note that a point must be included outside the patch at either end of the boundary to give proper slopes at the corner points. The next input points are for boundary number 2 and again from bottom to top. Boundary number 3 is loaded from left to right as is boundary number 4. A different number of points may be used to describe each boundary up to a maximum of 20 for each one.

Each of the input points has a status flag associated with it similar to that used in the Surface-Element-Input Method. The first point (the bottom point outside the patch on boundary number 1) has a status of 2. The first point on each of the other boundaries has a status of 1. All the other points have a status of 0 except the last point (the point on the right side outside the patch on boundary 4) which has a status of 3.

The input sheet contains two points per card. Every card must contain two points except the last which may have one point (loaded on the left side of the card).

The detailed input information required for this Geometry-Method Option is presented below.

Parametric-cubic-title card (12A4, 1X, I3, 1X, I3, 3X, 3I1, 2XI2, 1A4, I2). -

This card contains patch-control data and divisions to be used in converting the patch to element data.

Column	Code	Routine Format	Explanation
1-48	TITLE	CUBIC 12A4	Section or patch title. Any acceptable characters.
50-52	NOU	CUBIC I3	Number of division of the parametric-variable u. This controls the number of elements in the element mesh in the u-direction (right-justified integer).
54-56	NOW	CUBIC I3	Number of divisions of the parametric-variable w. This controls the number of elements in the element mesh in the w-direction (right-justified integer). If this number is an even number then the

Parametric-cubic-title card (Continued)

Column	Code	Routine Format	Explanation
			program will change it to the next higher odd number so that there will always be an odd number of elements in a column. This will give an even number of points in a column to fill out both the left and right sides of the Element-Data Card.
60	LAST	CUBIC I1	Last Flag. This flag controls the Status Flag (STATT) of the last-element point generated and the position of the Geometry-Data-Storage Unit (8). after all data has been written on it. = 1 Not active. Do not use. = 2 This is not the last patch. Set the last-point Status Flag STATT = 0, and read in a new set of patch data (including a new title card). = 3 Not active. Do not use. = 4 This is the last patch. Set the last-point status to STATT = 3, write an end of file on Unit IOUT.
61	ISOVR	CUBIC I1	First-Point Status-Override Flag. = 0 The Status Flag for the first coordinate point of the patch will be = 2 (normal mode). = 1 The Status Flag for the first-coordinate point of the patch will be set = 1. This will permit "joining together" several parametric-cubic patches to form a single section of surface-element data.
62	IPRINT	CUBIC I1	Print Flag. This flag controls the printing of the element data generated in this option. This data printout will contain the exact information written on the Geometry-Storage Unit (in BCD card image form). = 0 Do not print data.

Parametric-cubic-title card (Continued)

Column	Code	Routine Format	Explanation
			= 1 Print.
65-66	CASE	CUBIC I2	Case number (right-justified integer).
67-70	SECT	CUBIC 1A4	Numbers or letters to identify the vehicle panel.
71-72	ITYPE	CUBIC I2	Card type number = 06.

Parametric-cubic boundary-data card (3F10.4, I1, 3F10.4, I1, 8X, I2). -

This card contains the coordinates of the boundary curves for a parametric-cubic patch.

Column	Code	Routine Format	Explanation
1-10	X	CUBIC F10.4	X-coordinate of boundary-curve point.
11-20	Y	CUBIC F10.4	Y-coordinate of boundary-curve point.
21-30	Z	CUBIC F10.4	Z-coordinate of boundary-curve point.
31	STAT	CUBIC I1	Status Flag for the above set of coordinates (= 2, 1, 0, or 3). This flag controls the reading in of the boundary-curve data and is not the same as the STATUS flag that will be generated and written on the Geometry-Storage Unit along with the generated surface-element data.
32-41	XX	CUBIC F10.4	X-coordinate of boundary-curve point.
42-51	YY	CUBIC F10.4	Y-coordinate of boundary-curve point.
52-61	ZZ	CUBIC F10.4	Z-coordinate of boundary-curve point.
62	STATT	I1	Status Flag for the above set of coordinates (= 2, 1, 0, or 3). This flag

Parametric-cubic boundary-data card (Continued)

Column	Code	Routine Format	Explanation
			controls the reading in of the boundary-curve data and is not the same as the STATUS flag that will be generated and written on the Geometry-Storage Unit along with the generated surface-element data.
65-66	CASE		Case number. Not read by program.
67-70	SECT		Numbers or letters to identify the vehicle panel. Not read by program.
71-72	ITYPE	CUBIC I2	Card type number = 07.
77-80	SEQ		Card sequence number. Not read by program.

A data load sheet for the above data is shown on the next page.

Aircraft-Geometry Option

The Aircraft-Geometry Option has the capability of generating element data for six classes of aircraft surfaces. These are identified as follows.

1. Wings

   Airfoil ordinates are input at percent-chord locations along with the X, Y, Z coordinates of each airfoil leading edge. Wing-camber data and chord lengths are also input.

2. Fuselage

   A fuselage may be defined in segments and may be circular or arbitrary in cross section. If the fuselage is circular it may be input by a cross-sectional-area distribution. If the fuselage is arbitrary it may be input as X-Y-Z cross-sectional-coordinate data. Up to four fuselage segments may be specified on each entry into the Aircraft-Geometry Option.

3. Pods or Nacelles

   The X-Y-Z coordinates of the pod origin are input along with a pod-radii distribution. Up to nine pods may be specified on each entry to the Aircraft-Geometry Option.

PARAMETRIC CUBIC PATCH DATA

MARK IVS

ENGINEER: _____ DATE: _____

DIRECTIONS FOR KEYPUNCH
PUNCH IN ALL CARDS
DO NOT PUNCH
BLANK COLUMNS

CASE (66–70) SECT. (73–76) A,E,R,O

PARAMETRIC CUBIC TITLE CARD

NO. U (52–56) NO. W (56–60) LAST (60) ISOVR (61) PRINT (62) TYPE (72) SEQ. NO. (77–80)

1 10 20 30 40 52 56 60 62 72 77 80

PARAMETRIC CUBIC BOUNDARY DATA

±	X	±	Y	±	Z	STATUS	±	X	±	Y	±	Z	STATUS	TYPE	SEQ. NO.
1	2–10	11	12–30	21	22–30	31	32	33–41	42	43–51	52	53–61	62	72	77–80
														7	
														7	
														7	
														7	
														7	
														7	
														7	
														7	
														7	
														7	
														7	
														7	
														7	
														7	
														7	
														7	
														7	
														7	
														7	

DIRECTIONS FOR KEYPUNCH: DO NOT PUNCH BLANK COLUMNS

NO UNDERPUNCHES IN SIGN FIELDS

4. Fins

The X-Y-Z coordinates of the lower- and upper-airfoil leading edge of vertical fins are input along with airfoil ordinates at up to ten percent-chord locations. The chord lengths are also input. As many as six fins may be used on each entry to the Aircraft-Geometry Option. Only symmetrical airfoils are generated.

5. Canards or Horizontal Tails

The X-Y-Z coordinates of the inboard- and outboard-airfoil leading edge are input along with the airfoil ordinates and chord lengths. Up to two canards may be input on each entry to the Aircraft-Geometry Option. Unsymmetrical airfoils are permitted.

6. General-Airfoil Surface

The orientation of each airfoil is specified by X-Y-Z coordinates of the leading and trailing edges and a rotation angle. The airfoil ordinates and camber data are also input. This permits the description of wing or tail type of surfaces where the airfoils are not orientated in a fixed streamwise plane.

The first five classes of surfaces indicated above are the same as those available in the NASA Wave-Drag Program. The sixth surface is a new feature provided within the Arbitrary-Body Program that allows the geometric description of a surface composed of airfoil sections that may be arbitrarily orientated in space. This removes some of the restrictions imposed within the wing, fin, and canard options used in the NASA Wave-Drag Program. Also, additional parameters may be specified on the pod-input data to allow arbitrary orientation of the pods or nacelles.

Various combinations of the above shapes may be used in describing most aircraft configurations. However, in some situations a part of a vehicle may not be accurately described by one of the above components. In this case, the particular part of the vehicle may be input or generated using the completely arbitrary-shape capabilities in the other parts of the program (i.e., input elements, parametric-cubic patches, etc.). For other problems it may be easiest to generate the vehicle using the Aircarft-Geometry Option and then alter those cards that need changes by hand in order to give an accurate representation of the shape. This may be necessary to accurately describe such regions as wing roots, fillets, etc.

Note that in the description of each of the surfaces above, certain restrictions exist as to the maximum number of fuselage segments, pods, fins, etc., that may be generated on a single entry into the Aircraft-Geometry Option. It should be noted, however, that all such limitations may be overcome by entering the Aircraft-Geometry Option as many times as may be required.

The output data for the Aircraft-Geometry Option consist of element-data cards with two coordinate points and accompanying Status Flags recorded on each card

Aircraft-Geometry Option (Continued)

(Type 3 Arbitrary-Body Program Cards). These cards are written on the Geometry-Storage Unit (Unit 8) for use by the rest of the Arbitrary-Body Program. The card decks generated in this manner may be used directly as input data for subsequent runs on the Hypersonic Arbitrary-Body Program, or as input to the Douglas Arbitrary-Body Supersonic Wave-Drag Program and the Douglas Potential-Flow Program (the Neumann Program). These same element cards may be used to generate pictures using on-line interactive graphics programs, and in programs that use large electro-mechanical drafting devices such as the Orthomat and Gerber Plotters.

Some users may make use of the Aircraft-Geometry Option and the Arbitrary-Body Program picture-drawing capabilities as a tool in validating geometry data for the NASA Wave-Drag Program (the Harris Program). For such applications, however, care should be taken to verify that the input data for the users' version of the NASA program is the same as required by the Aircraft-Geometry Option of the Hypersonic Arbitrary-Body Program. Note that the sixth surface type provided in the Aircraft-Geometry Option (arbitrary-airfoil orientation) is not available for use on the NASA Wave-Drag Program. Since all programs tend to change with time it may be necessary for the user to make modifications to the Aircraft-Geometry Option to maintain consistency in input data with the NASA Wave-Drag Program.

Users of the Aircraft-Geometry Option should exercise care in selecting input parameters to assure that the resulting surface-element data will meet the needs of their problem. This rather obvious statement is necessary because of the multifunction uses that this option serves.

The various methods within the Aircraft-Geometry Program are selected by input flags on a control card. The various parameters, tables, etc., for each aircraft component are given on a separate set of cards for each type of surface. The order and identification of each of the input cards is given in the list below. Each card, if it is to be used, must be in the order indicated.

Card or Card Set Number	Card Identification
1	Title and Identification Card
2	Control-Flag Card
3	Wing-Area Card (if required)
4	Wing Percent-Chord-Location Card(s)
5	Airfoil Leading-Edge-Coordinate Card(s)
6	Wing Camber-Line Card(s) (for each airfoil, if required)
7	Wing Airfoil-Ordinate Card(s) (for each airfoil)

Card or Card Set Number	Card Identification
8	Fuselage X-Station Card(s) (for first segment)
9	Fuselage-Camber Card(s) (if required)
10	Fuselage Cross-Section-Area Card(s) (if required)
11	Fuselage Y-Ordinates (for arbitrary shape) (if required)
12	Fuselage Z-Ordinates (for arbitrary shape) (if required)
13	Repeat 11 and 12 for all cross sections of segment.
14	Repeat 8 through 13 for all fuselage segments.
15	Pod-Origin Card
16	Pod X-Station Card(s)
17	Pod-Radii Card(s)
18	Repeat 15 through 17 for all pods.
19	Fin Leading-Edge-Coordinate Card
20	Fin Percent-Chord-Location Card
21	Fin Airfoil-Ordinate Card
22	Repeat 19 through 21 for all fins.
23	Canard Leading-Edge-Coordinate Card
24	Canard Percent-Chord-Location Card
25	Canard Upper-Ordinate Card
26	Canard Lower-Ordinate Card (if required)
27	Repeat 23 through 26 for all canards
28	General Airfoil-Surface-Control-Flag Card
29	Airfoil Percent-Chord-Location Card(s)
30	Airfoil Orientation Card(s)
31	Airfoil Camber-Line Card(s)
32	Airfoil Ordinate-Thickness Card(s)

Card or Card Set Number	Card Identification
33	Repeat 28 through 32 for multiple airfoil surfaces as required.
34	Repeat 1 through 33 for multiple configurations as required.
35	Type 99 card (Normal Return to Executive Program).

The detailed descriptions given on the following pages include all input cards and parameters. Where it might be useful the mnemonics used in the program are also given. On some of the cards identification information is punched in card columns 73-80. Although this information is not used by the program its use may help to eliminate errors in card order. The card field to be used for input numbers is indicated for each card. All integers should be punched in the right-most column of the field. Real numbers may be punched anywhere in the field specified. Except for the integers on the Control-Flag Card, all other input is in the form of real numbers with the decimal point and sign punched on the card (i.e., -59.56).

The type 99 card contains a 99 in card columns 71 and 72. The remainder of the card has the same format as the Title and Identification Card (Card 1); however,these remaining fields are usually left blank.

A chart showing the data flow logic for the Aircraft-Geometry Option is presented in Figure 7.

Aircraft-geometry identification and control. -

Title and identification card (9A3, 8A4, I1, 2X, I1, 1XI2, 4X, A2, 2A4):

Column	Code	Routine Format	Explanation
1-59	CARD	AIRCFT 9A3, 8A4	The title that is to appear at the top of the output pages. Any acceptable machine characters.
60	ISTAT3	AIRCFT I1	Flag to control the generation of a dummy element with a very small surface area in order to introduce a Status = 3 at the appropriate place in a geometry deck. = 0 Dummy element will be included at the very end of the Type 3 cards produced in the Aircraft-Geometry Option. This means that the last Type 3 card will have a Status 3 flag.

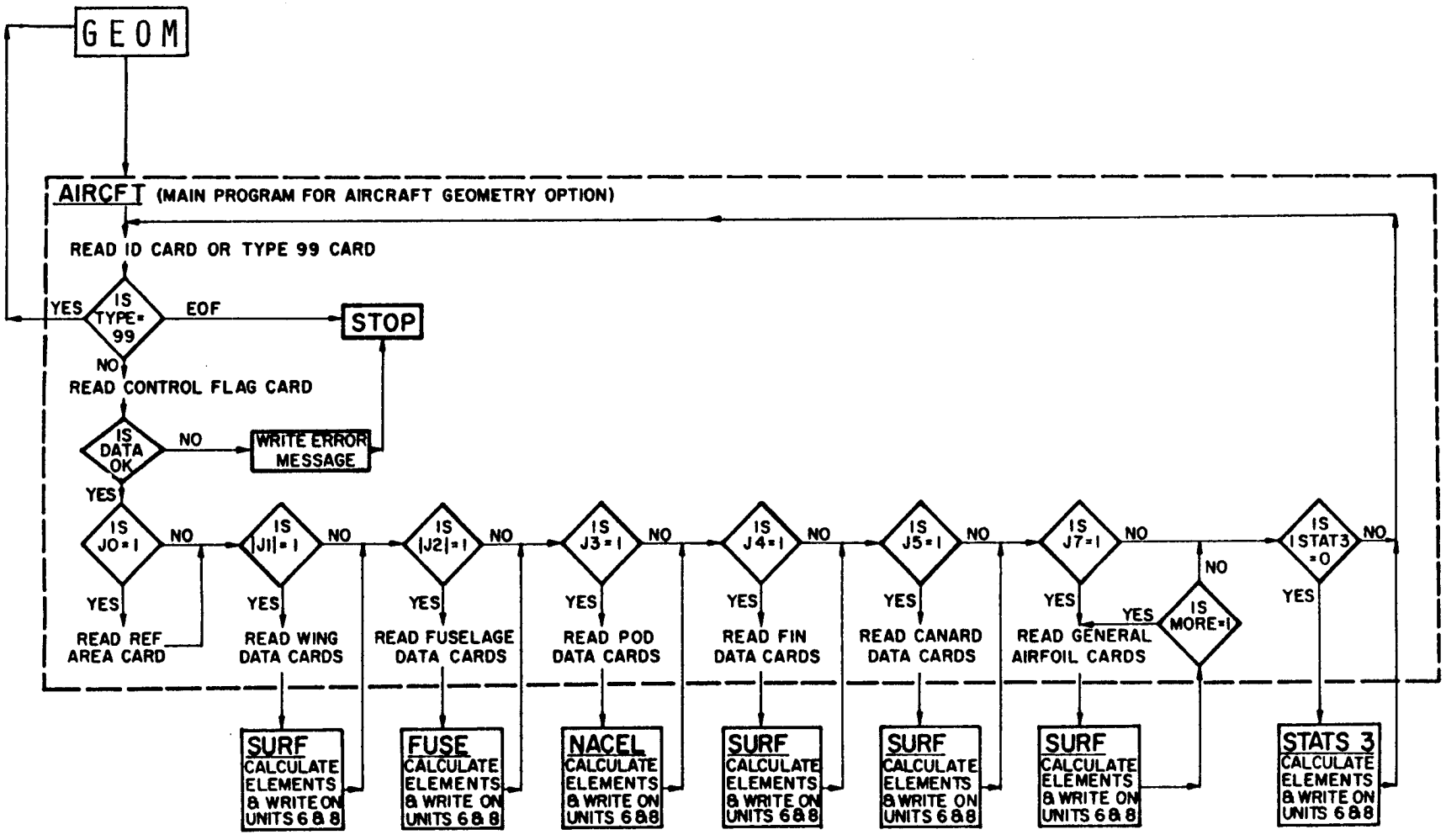


FIGURE 7. AIRCRAFT GEOMETRY OPTION DATA FLOW LOGIC.

Title and identification card (Continued)

Column	Code	Routine Format	Explanation
			= 1 No dummy Status 3 element will be generated. The last data point in the Aircraft-Geometry Cards produced will have a Status = 0.
63	IHARIS	AIRCFT I1	NASA-Harris Input-Coordinate Flag. = 0 Arbitrary-Body Program coordinates will be used. The usual practice is to have the nose of the vehicle at the origin of the coordinate system with the tail having a negative X-station. = 1 The NASA-Harris coordinate system will be used. The vehicle nose is at the origin and all the X-coordinates are positive for the input data on the Aircraft-Geometry-Data Cards. The program will change them to negative values before the final Type 3 Cards are written on the storage unit to be consistant with the Arbitrary-Body System.
65-66	CASE	AIRCFT I2	Case number to be printed at the top of data-output pages and in card columns 65-66 on the Type 3 Cards produced by the Aircraft-Geometry Option.
71-72	TYPE	AIRCFT A2	Case-Termination Flag. If input = 99 the Aircraft-Geometry Option will stop with this card and return to the Geometry routine. If left blank the program will continue reading in Aircraft-Geometry-Data Cards.

Control-flag card (7I3, I1, I2, 16I3, 8X):

Column	Code	Routine Format	Explanation
1-3	JO	AIRCFT I3	Wing-Area Flag. = 0 Wing-Area Card is not included. = 1 Wing-Area Card is to be read.

Control-flag card (Continued)

Column	Code	Routine Format	Explanation
4-6	J1	AIRCFT I3	Wing-Data and Camber Flag. = 0 No wing data are used. = 1 Cambered-wing data are to be read. = -1 Uncambered-wing data are to be read.
7-9	J2	AIRCFT I3	Fuselage-Control Flag. = 0 No fuselage data are used. = 1 Data for arbitrarily-shaped fuselage will be read. = -1 Data for circular fuselage will be read.
10-12	J3	AIRCFT I3	Pod-Control Flag. = 0 No pod data are used. = 1 Pod data are to be read.
13-15	J4	AIRCFT I3	Fin-Control Flag. = 0 No fin data are used. = 1 Fin data are to be read.
16-18	J5	AIRCFT I3	Canard-Control Flag. = 0 No canard data are used. = 1 Canard data are to be read.
19-21	J6	AIRCFT I3	Fuselage-Camber and Symmetry Flag. = 0 Fuselage-camber data will be read. = 1 Configuration is symmetrical with respect to the X-Y plane (uncambered-circular fuselage is used). = -1 Fuselage is uncambered. = 2 Uncambered, arbitrary fuselage.
22	J7	AIRCFT I1	General-Airfoil-Surface Flag. = 0 No airfoil data are used. = 1 General-Airfoil-Surface-Control Card to be read.
23-24	NWAF	AIRCFT I2	Number of airfoils used to describe the wing = 2 to 20.
25-27	NWAFOR	AIRCFT I3	Number of percent-chord points used to define each wing airfoil section. = 3 to 30.
28-30	NFUS	AIRCFT I3	The number of fuselage segments to be read. = 1 to 4.

Control-flag card (Continued)

Column	Code	Routine Format	Explanation
31-33	NRADX(1)	AIRCFT I3	Number of Y-Z coordinate points used to describe each cross section for the first fuselage segment. This parameter is used for both arbitrary and circular-fuselage segments. = 3 to 30.
34-36	NFORX(1)	AIRCFT I3	Number of X cross sections to be used for each fuselage segment. = 2 to 30.
37-39	NRADX(2)	AIRCFT I3	Same as Field 31-33 for second fuselage segment.
40-42	NFORX(2)	AIRCFT I3	Same as Field 34-36 for second fuselage segment.
43-45	NRADX(3)	AIRCFT I3	Same as Field 31-33 for third fuselage segment.
46-48	NFORX(3)	AIRCFT I3	Same as Field 34-36 for third fuselage segment.
49-51	NRADX(4)	AIRCFT I3	Same as Field 31-33 for fourth fuselage segment.
52-54	NFORX(4)	AIRCFT I3	Same as Field 34-36 for fourth fuselage segment.
55-57	NP	AIRCFT I3	Number of pods to be input (up to 9).
58-60	NPODOR	AIRCFT I3	Number of stations to be used in the pod-radii-distribution input. This is the same for all pods. = 2 to 30.
61-63	NF	AIRCFT I3	Number of vertical fins to be input (up to 6).
64-66	NFINOR	AIRCFT I3	Number of ordinates used to define each fin airfoil. This is the same for all fins. = 3 to 10.
67-69	NCAN	AIRCFT I3	Number of canards or horizontal tails to be input (up to 2).

Control-flag card (Continued)

Column	Code	Routine Format	Explanation
70-72	NCANOR	AIRCFT I3	Number of ordinates used to define the airfoils. This is the same for all canards. = 3 to 10 Airfoil is symmetrical; upper ordinates only will be read. = -3 to -10 Airfoil is unsymmetrical; lower ordinates will be read right after the upper values are read in.

Wing-area card (F7.2):

Column	Code	Routine Format	Explanation
1-7	REFA	AIRCFT F7.2	This card is required if JO = 1 on the Control-Flag Card. This parameter is not used by the program but may be present in some decks set up for the NASA Wave-Drag Program. If JO = 0 on the Control-Flag Card then the Wing-Area Card is not included in the deck.

Wings. -

The input information required by the Aircraft-Geometry Option to define a wing with streamwise airfoils is as follows:

1. Number of airfoils.

2. Number of airfoil percent-chord points used to define the airfoils.

3. A table of percent-chord locations that are to be used for the airfoil-thickness and camber distributions.

4. The X-Y-Z coordinates of the leading edge of each airfoil.

5. The chord length of each airfoil.

6. The airfoil-ordinate data in percent of chord length at each percent-chord position for each airfoil.

7. A flag to indicate when camber data are to be read in or set equal to zero.

Wings (Continued)

8. Camber values of the mean-camber line (ΔZ) at each percent-chord location for each airfoil.

The input information required to describe a wing is illustrated in Figure 8.

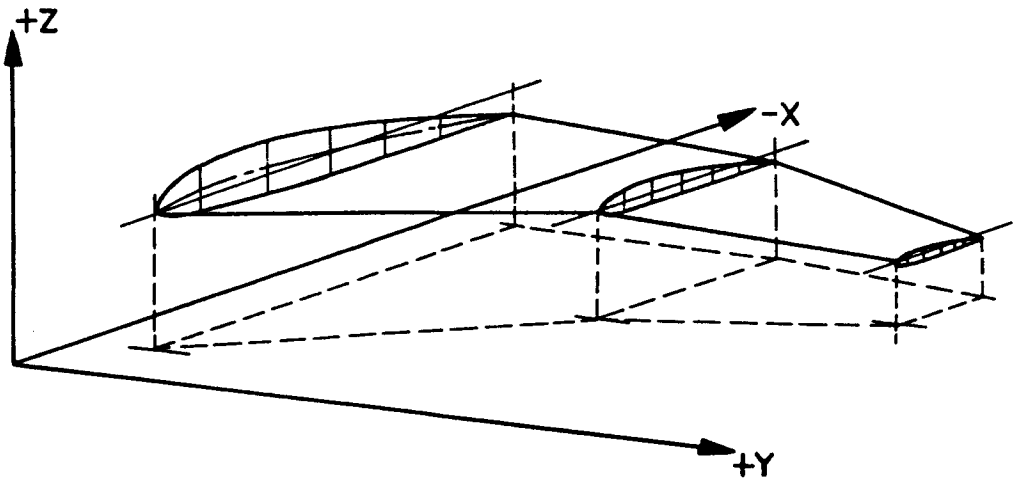


FIGURE 8. WING WITH STREAMWISE AIRFOILS.

Note that each airfoil lies in an X-Z plane at a fixed spanwise station, Y. This will pose problems in some applications at the wing-fuselage juncture, particularly for area-ruled fuselages.

The Z-coordinates of each surface point are calculated from the following relationship.

$$Z = ZO + DZ * C * WAFORD(I,J) + TZORD(I,J)$$

where

Z = final Z-coordinate of a point on the airfoil

DZ = (upper surface) - (lower-surface factor)
= +1.0 for upper surface.
= -1.0 for lower surface.

C = chord length/100.0

WAFORD = upper-airfoil thickness in percent of chord length
subscript I = the airfoil number (=1 for the inboard airfoil)
subscript J = the number of chordwise location

TZORD = camber, ΔZ

ZO = Z-coordinate of the airfoil-leading-edge point

The order of the generated X-Y-Z surface points is shown in Figure 9a for the upper surface of the wing. The wing lower surface is shown in Figure 9b.

In the wing shown in Figure 9 there were seven percent-chord locations from the leading edge to the trailing edge. Since each Element-Data Card generated (Type 3 Card) contains two data points, three and one-half cards will be required for the root upper surface. Rather than beginning the next wing chord on the last half of the third card, this field is filled by a dummy point (point 8) which is a repeat of the trailing-edge point (point 7). This dummy point is furnished automatically by the program when it is required and permits each airfoil to be started on a new card. This facilitates the manual manipulation of resulting data decks to meet various needs.

The wing lower surface is considered as a new vehicle section. To obtain the correct "outward" side of the surface the generation of points starts at the tip rather than the root as was done for the upper surface. Dummy trailing-edge points are generated just as was done for the upper surface. In the example shown, point 1 will have a Status Flag of 2, points 9 and 17 will have Status = 1, and all the rest will have Status = 0.

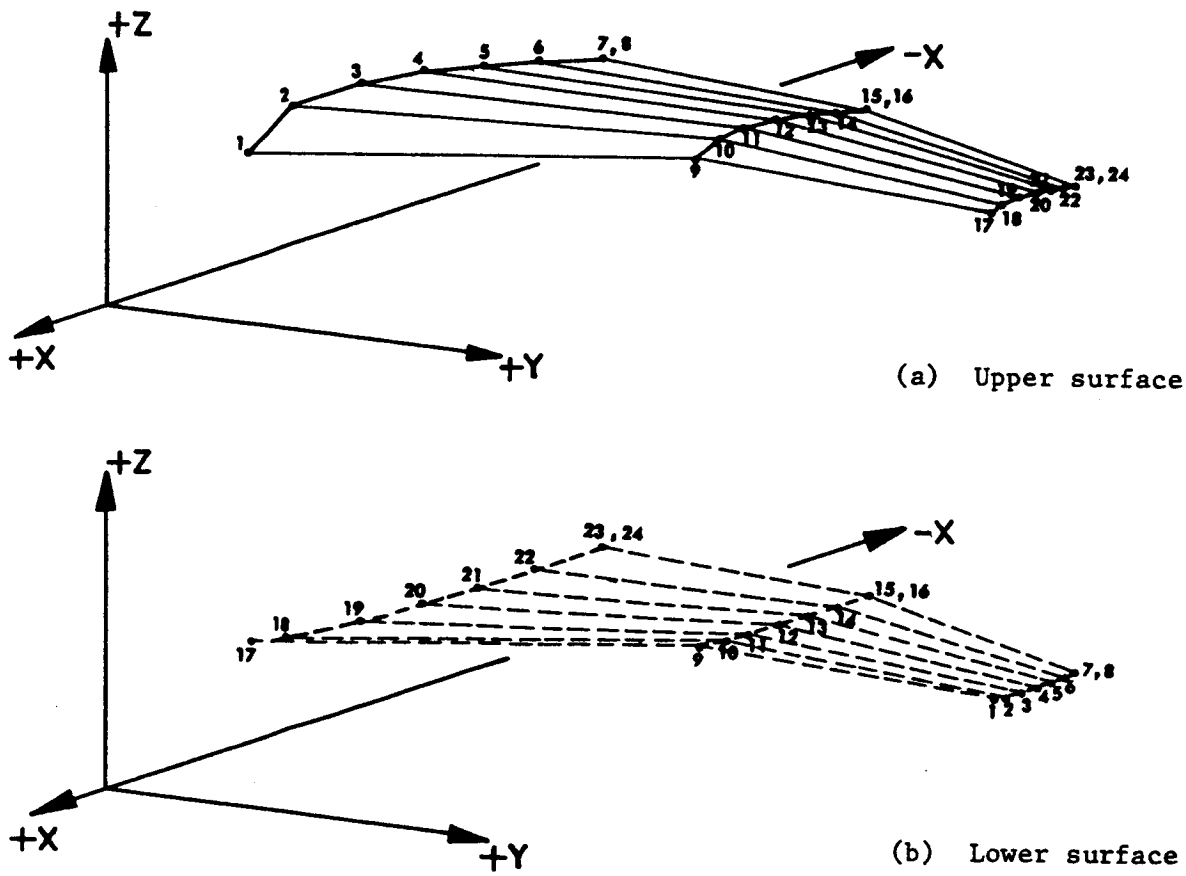


FIGURE 9. POINTS AND ELEMENTS GENERATED FOR WING SURFACE.

Note that a surface point is calculated at each input percent-chord location for each airfoil. The number of percent-chord locations and the number of airfoils will determine the number of surface points generated and the number of resulting surface elements. In the example shown in Figure 9 a total of 28 elements were generated (including 4 that result from the dummy trailing-edge points).

Wing percent-chord-location card(s) (10F7.0, 10X):

Column	Code	Routine Format	Explanation
1-7	XAF(1)	AIRCFT	Table of percent-chord locations that
8-14	XAF(2)	10F7.0	are to be used for the airfoil-thickness
15-21	XAF(3)		and camber coordinates. Use as many
22-28	XAF(4)		cards as required with 10 numbers on
29-35	XAF(5)		each card. Use as many fields and cards
36-42	XAF(6)		as is specified by NWAFOR (Field 25-27)
43-49	XAF(7)		on the Control-Flag Card.
etc.	etc.		

Airfoil leading-edge-coordinate cards (4F7.0, 52X):

Column	Code	Routine Format	Explanation
1-7	WAFORD(I,1)	AIRCFT	X-coordinate of the airfoil leading edge.
8-14	WAFORG(I,2)	4F7.0	Y-coordinate of the airfoil leading edge.
15-21	WAFORG(I,3)		Z-coordinate of the airfoil leading edge.
22-28	WAFORG(I,4)		The airfoil streamwise chord length.
73-80			May be punched WAFORGi, where i denotes the airfoil number.

Note: Repeat this card for all airfoils, starting with the inboard airfoil and working to the outboard-tip airfoil. The number of these cards is given by the parameter NWAF (Field 23-24) on the Control-Flag Card and must not be greater than 20.

Wing camber-line cards (10F7.0, 10X):

Not required if J1 = -1 on the Control-Flag Card.

Column	Code	Routine Format	Explanation
1-7	TZORD(J,1)	AIRCFT	Camber values of the mean-camber line
8-14	TZORD(J,2)	10F7.0	(ΔZ) at each percent-chord location for
15-21	TZORD(J,3)		each airfoil. Use as many cards as
22-28	TZORD(J,4)		required with 10 numbers on each card.
29-35	TZORD(J,5)		Each airfoil must have as many numbers
36-42	TZORD(J,6)		as was specified by the parameter NWAFOR
43-49	TZORD(J,7)		in field 25-27 on the Control-Flag Card.
50-56	TZORD(J,8)		There will be as many sets of these

Wing camber-line cards (Continued)

Column	Code	Routine Format	Explanation
57-63	TZORD(J,9)		cards as was indicated by the parameter NWAF in field 23-24 on the Control-Flag Card. The first number for each airfoil should start on a new card. The identification TZORDj may be punched in card columns 73-80, where j denotes the airfoil number.
64-70	TZORD(J,10)		
1-7	TZORD(J,11)		
etc.	etc.		

Wing airfoil-ordinate cards (10F7.0, 10X):

Column	Code	Routine Format	Explanation
1-7	WAFORD(J,1)	AIRCFT	Wing airfoil-thickness ordinates as a percent of chord length at each percent-chord-ordinate position for each airfoil. Use as many cards as required with 10 numbers on each card. Each airfoil must have as many numbers as was specified by the parameter NWAFOR in field 25-27 on the Control-Flag Card. There will be as many sets of these cards as was indicated by the parameter NWAF in field 23-24 of the Control-Flag Card. The first number for each airfoil should start on a new card. The identification WAFORDj may be punched in card columns 73-80, where j denotes the airfoil number.
8-14	WAFORD(J,2)	10F7.0	
15-21	WAFORD(J,3)		
22-28	WAFORD(J,4)		
29-35	WAFORD(J,5)		
36-42	WAFORD(J,6)		
43-49	WAFORD(J,7)		
50-56	WAFORD(J,8)		
57-63	WAFORD(J,9)		
64-70	WAFORD(J,10)		
1-7	WAFORD(J,11)		
etc.	etc.		

Fuselage. -

The input information required by the Aircraft-Geometry Option to define a fuselage is as follows:

1. Fuselage-shape flags (circular, arbitrary, cambered).

2. Number of fuselage segments (1 to 4).

3. Number of Y-Z coordinate points used to describe an X-cross section for each fuselage segment (3 to 30).

4. Number of X cross sections to be used for each fuselage segment (2 to 30).

5. A table of X-values of the fuselage cross sections for each fuselage segment.

6. Tables of Y-Z values to describe each cross section for arbitrary-shaped fuselage.

7. Fuselage centerline-camber distribution.

8. Cross-sectional-area distribution of the fuselage if it is circular.

From the above input items we see that the fuselage may be circular or arbitrary in cross section, may have camber, and may be made up of as many as four segments. However, a single fuselage cannot be made up to a combination of circular and arbitrary cross sections. (This comment only applied for a single pass into the Aircraft-Geometry Option. Multiple entries into the Aircraft-Geometry Option from the Hypersonic Arbitrary-Body Executive Main Program permits an unlimited combination of program capabilities.)

The order of the generated fuselage-coordinate points is from the bottom around to the top. The first point for each fuselage segment has a Status Flag of 2, each new cross section starts with a Status of 1, and all the other points have Status = 0. If the last point for a cross-section fills only the left half of the Type 3 Element-Data Card, a dummy point is generated to fill the right field of the card. Figure 10 shows the order of the generated surface points for an arbitrarily-shaped fuselage. Only two fuselage segments are shown.

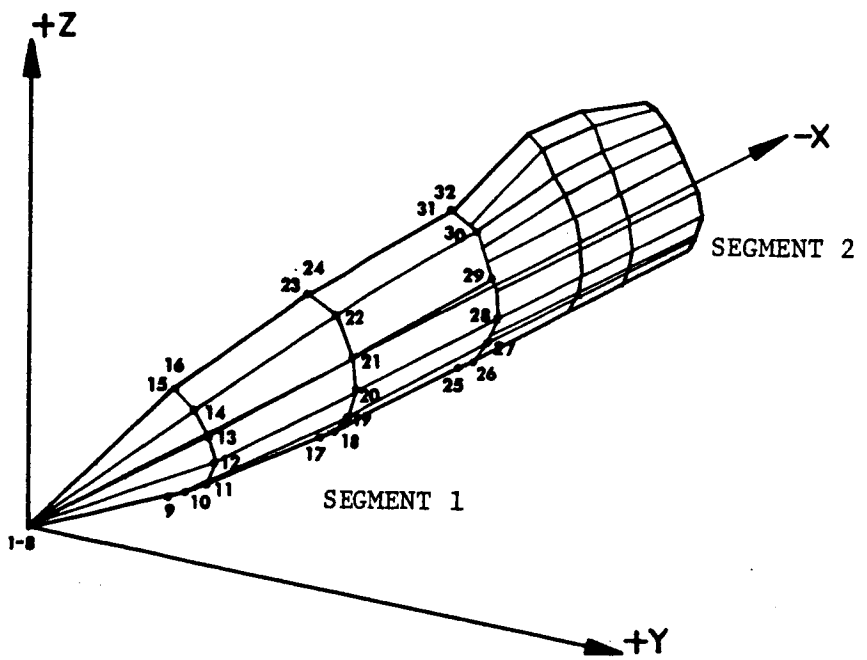


FIGURE 10. FUSELAGE GEOMETRY GENERATION

Fuselage X-station card(s) (10F7.0, 10X):

Column	Code	Routine Format	Explanation
1-7	XFUS(I,1)	AIRCFT	Table of X-station values to be used for first fuselage segment. Use as many cards as required with 10 numbers on each card. The number of cross sections used may be the same as indicated by the parameter NFORX on the Control-Flag Card. The X-stations must be in an order proceeding from the front of the vehicle to the rear.
8-14	XFUS(I,2)	10F7.0	
15-21	XFUS(I,3)		
22-28	XFUS(I,4)		
29-35	XFUS(I,5)		
36-42	XFUS(I,6)		
43-49	XFUS(I,7)		
50-56	XFUS(I,8)		
57-63	XFUS(I,9)		
64-70	XFUS(I,10)		
1-7	XFUS(I,11)		The identification XFUSj may be punched in card columns 73-80, where j denotes the number of the last fuselage station given on that card.
etc.	etc.		

Fuselage-camber card(s) (10F7.0, 10X):

Required only if J6 = 0 on the Control-Flag Card.

Column	Code	Routine Format	Explanation
1-7	ZFUS(I,1)	AIRCFT	Fuselage-camber distribution for first fuselage segment. Use as many cards as required with 10 numbers on each card. The number of camber points used must be the same as indicated by the parameter NFORX on the Control-Flag Card. For an arbitrarily shaped fuselage this parameter will not actually be used in generating the surface-coordinate points. However, if the parameter J6 = 0 on the Control-Flag Card, then the appropriate number of Fuselage-Camber Cards must be present in the deck (all the values may be = 0.0).
8-14	ZFUS(I,2)	10F7.0	
15-21	ZFUS(I,3)		
22-28	ZFUS(I,4)		
29-35	ZFUS(I,5)		
36-42	ZFUS(I,6)		
43-49	ZFUS(I,7)		
50-56	ZFUS(I,8)		
57-63	ZFUS(I,9)		
64-70	ZFUS(I,10)		
1-7	ZFUS(I,11)		The identification ZFUSj may be punched in card columns 73-80, where j denotes the number of the last fuselage station given on that card.
etc.	etc.		

Fuselage cross-section-area card(s) (10F7.0, 10X):

Not required if the fuselage is arbitrary in shape (if J2 = 1).

Column	Code	Routine Format	Explanation
1-7	FUSARD(I,1)	AIRCFT	A table of fuselage cross-sectional-area distribution at each station for the first fuselage segment. Use as many cards as required with 10 numbers on each card. The number of points used must be as indicated by the parameter NFORX on the Control-Flag Card. The identification FUSARDj may be punched in card columns 73-80, where j denotes the station number.
8-14	FUSARD(I,2)	10F7.0	
15-21	FUSARD(I,3)		
22-28	FUSARD(I,4)		
29-35	FUSARD(I,5)		
etc.	etc.		

Fuselage Y-ordinate card(s) (10F7.0, 10X):

Used for arbitrarily-shaped fuselage only. Do not use these cards unless J2 = 1 on the Control-Flag Card.

Column	Code	Routine Format	Explanation
1-7	YFUSY(I,J,1)	AIRCFT	Y-ordinates for one fuselage cross section, starting at the bottom and going around to the top of the section. Use as many cards as required with 10 numbers on each card. The parameter NRADX on the Control-Flag Card specifies the number of Y-ordinates required for each cross section. Each set of Y-Ordinate Cards are followed immediately by a Z-ordinate set of cards for that same cross section. The number of ordinates may range from 3 to 30.
8-14	YFUSY(I,J,2)	10F7.0	
15-21	YFUSY(I,J,3)		
22-28	YFUSY(I,J,4)		
29-35	YFUSY(I,J,5)		
36-42	YFUSY(I,J,6)		
43-49	YFUSY(I,J,7)		
etc.	etc.		

Fuselage Z-ordinate card(s) (10F7.0, 10X):

Used for arbitrarily-shaped fuselage only. Do not use these cards unless J2 = 1 on the Control-Flag Card.

Column	Code	Routine Format	Explanation
1-7	ZFUSZ(I,J,1)	AIRCFT	Z-ordinates for one fuselage cross section, starting at the bottom and going around to the top of the section. These Z-ordinates correspond to the Y-ordinates input on the Y-Ordinate Card described above. Use as many cards as required
8-14	ZFUSZ(I,J,2)	10F7.0	
15-21	ZFUSZ(I,J,3)		
22-28	ZFUSZ(I,J,4)		
29-35	ZFUSZ(I,J,5)		
36-42	ZFUSZ(I,J,6)		

Fuselage Z-ordinate card(s) (Continued)

43-49	ZFUSZ(I,J,7)	with 10 numbers on each card. The parameter NRADX on the Control-Flag Card specifies the number of Z-ordinates required for each cross section. Each set of Z-Ordinate Cards must be right behind the corresponding set of Y-Ordinate Cards.
50-56	ZFUSZ(I,J,8)	
57-63	ZFUSZ(I,J,9)	
etc.	etc.	

Note Paired sets of Y-Ordinate and Z-Ordinate Cards are repeated for each fuselage cross section until all cross sections for a single fuselage segment are read in. The number of paired sets is given by the parameter NFORX input on the Control-Flag Card.

The order of cards for the second fuselage segment is the same as prepared for the first segment. This includes fuselage station, camber, cross-section area, and Y-Z ordinates for each fuselage segment.

Pods or nacelles. -

A pod or nacelle is a body of revolution with its axis arbitrarily located with reference to the vehicle-axis system. This increased capability has been added without effecting the NASA Wave-Drag Program input format (the NASA program is limited to having the pod axis parallel to the vehicle X-axis). The pod is defined with respect to its own coordinate system (X'-Y'-Z'), the orientation of which is considered to have been achieved through a yaw-pitch sequence of rotations. The parameters used in defining the pod and the formation of surface elements are illustrated in Figure 11.

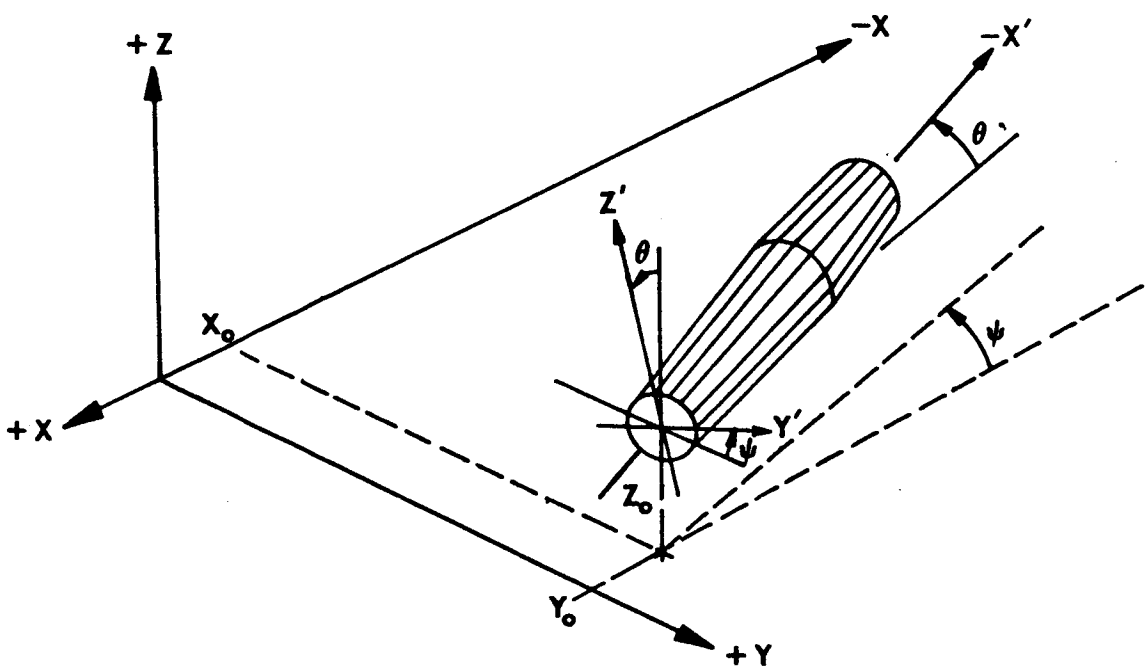


FIGURE 11. POD OR NACELLE GEOMETRY.

The input information required to define a pod or nacelle is as follows:

1. Number of pods (up to 9).

2. Number of stations to be used in the pod-radii-distribution input (2 to 30). This is the same for all pods.

3. The X-Y-Z coordinates of the origin and end of each pod in the vehicle-coordinate system.

4. A table of X-ordinates (relative to pod origin) for the pod-radii distribution.

5. Pod-radii distribution for each pod.

The order of the generated surface points is from the bottom around to the top. The first point of each pod has a Status of 2, each new station starts with a Status of 1, and all other points have Status = 0. If the last point for a station fills only the left half of the Type 3 Element-Data Card, a dummy point is generated to fill the right half of the card. When the pod axis lies in the X-Z origin plane, only half the pod is generated ($-90° \leq \omega \leq +90°$). Otherwise elements for the complete pod are determined.

In addition to specifying the axis orientation, the number of elements in 180° may also be specified. If this expanded capability is not used and the input fields are left blank, the program assumes the pod axis is parallel to the vehicle axis, and elements are generated every 15° in ω.

<u>Pod-origin card</u> (3F7.0, 3X, I1, 3X, 3F7.0, 6X, I1, 5X, I2):

Column	Code	Routine Format	Explanation
1-7	PODORG(I,1)	F7.0	X-coordinate of the origin of the first pod with respect to the vehicle-coordinate-system origin.
8-14	PODORG(I,2)	F7.0	Y-coordinate of the origin of the first pod with respect to the vehicle-coordinate-system origin.
15-21	PODORG(I,3)	F7.0	Z-coordinate of the origin of the first pod with respect to the vehicle-coordinate-system origin.
25	IOR	I1	Arbitrary Orientation Flag. If this value does not equal 1, fields 29-49 are ignored and the pod axis is assumed parallel to the vehicle X-axis.
29-35	PODORG(I,4)	F7.0	X-coordinate of the end point of the first pod with respect to the vehicle-coordinate system.

Pod-origin card (Continued)

Column	Code	Routine Format	Explanation
36-42	PODORG(I,5)	F7.0	Y-coordinate of the end point of the first pod with respect to the vehicle-coordinate system.
43-49	PODORG(I,6)	F7.0	Z-coordinate of the end point of the first pod with respect to the vehicle-coordinate system.
56	IEL	I1	Element-Number Flag. If this value does not equal 1, field 62-63 is ignored and 12 elements are assumed in 180°.
62-63	NEL	I2	Number of elements in 180° ($\leq$ 36).
73-80			The card identification, PODORGi may be punched in these columns where i denotes the pod number.

Pod X-station card(s) (10F7.0, 10X):

1-7	XPOD(I,1)	AIRCFT	Table of X-ordinates (relative to pod
8-14	XPOD(I,2)	10F7.0	origin) to be used for the pod-radii
15-21	XPOD(I,3)		distribution. Use as many cards as re-
22-28	XPOD(I,4)		quired with 10 numbers on each card.
29-35	XPOD(I,5)		The number of cross sections used must
36-42	XPOD(I,6)		be the same as indicated by the parameter
43-49	XPOD(I,7)		NPODOR given in field 58-60 on the
50-56	XPOD(I,8)		Control-Flag Card. The first X-ordinate
57-63	XPOD(I,9)		must be zero, and the last X-ordinate is
64-70	XPOD(I,10)		the length of the pod. The identifica-
1-7	XPOD(I,11)		tion XPODi may be punched in card columns
etc.	etc.		73-80, where i denotes the pod number.

Pod-radii card(s) (10F7.0, 10X):

1-7	PODR(I,1)	AIRCFT	A table of pod-radii distribution at each
8-14	PODR(I,2)	10F7.0	X-station for the first pod. Use as many
15-21	PODR(I,3)		cards as required with 10 numbers on each
22-28	PODR(I,4)		card. The number of points used must be
29-35	PODR(I,5)		the same as indicated by the parameter
36-42	PODR(I,6)		NPODOR given in field 58-60 on the
43-49	PODR(I,7)		Control-Flag Card. The identification
etc.	etc.		PODRi may be punched in card columns 73-80,
			where i denotes the pod number.

Note: A new set of all three pod-input card sets is required for each pod (a maximum of 9 are provided for). Note that every pod uses the same value for the parameter NPODOR. If the Y-ordinate of the

Pod-radii card(s) (Continued)

Note: (Continued)
pod origin is 0.0 then only half of the symmetrical-centerline pod is generated. If the Y-ordinate is not equal to 0.0 then the entire pod will be generated.

Fins. -

Fins are defined by upper- and lower-uncambered airfoils. Each airfoil lies in an X-Y plane at a fixed vertical distance, Z. This restriction may pose problems in some applications at the fin-fuselage juncture. The leading edge of each airfoil is located relative to the coordinate system origin by input X-Y-Z displacements. For fins located on the plane of symmetry only half of the surface is generated. If the fin is located off the plane of symmetry both sides will be generated. The generation of surface points and elements follows the same general procedure as outlined previously for the wing.

The input information required to define a fin is as follows:

1. Number of vertical fins (up to 6).

2. Number of ordinates used to define each fin airfoil section (3 to 10).

3. The X-Y-Z coordinates of the fin-lower and -upper leading edges.

4. Chord lengths for the lower and upper airfoils.

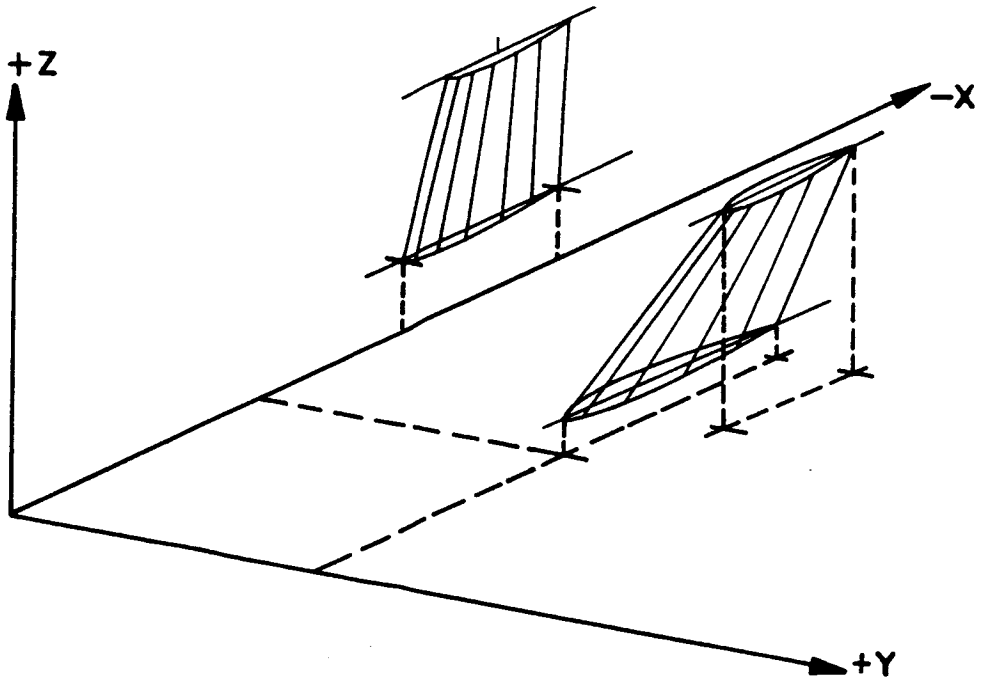


FIGURE 12. FIN GEOMETRY GENERATION

Fins (Continued)

5. A table of percent-chord locations that are used to define airfoils.

6. A table of airfoil ordinates as a percent of the chord length.

The input and formation of fin-surface elements are illustrated in Figure 12.

Fin leading-edge-coordinate card (8F7.0, 24X):

Column	Code	Routine Format	Explanation
1-7	FINORG(N,1,1)	AIRCFT	X-coordinate of root-airfoil leading edge.
8-14	FINORG(N,1,2)	8F7.0	Y-coordinate of root-airfoil leading edge.
15-21	FINORG(N,1,3)		Z-coordinate of root-airfoil leading edge.
22-28	FINORG(N,1,4)		Chord length of root airfoil.
29-35	FINORG(N,2,1)		X-coordinate of tip-airfoil leading edge.
36-42	FINORG(N,2,2)		Y-coordinate of tip-airfoil leading edge.
43-49	FINORG(N,2,3)		Z-coordinate of tip-airfoil leading edge.
50-56	FINORG(N,2,4)		Chord length of tip airfoil.
73-80			The card identification FINORGn, where n denotes the fin number.

Fin percent-chord-location card (10F7.0, 10X):

Column	Code	Routine Format	Explanation
1-7	XFIN(N,1)	AIRCFT	Table of percent-chord locations that
8-14	XFIN(N,2)	10F7.0	are to be used for the airfoil-thickness
15-21	XFIN(N,3)		ordinates. Only one card is used and
22-28	XFIN(N,4)		may contain up to 10 numbers. The number
29-35	XFIN(N,5)		of percent-chord locations used must be
36-42	XFIN(N,6)		the same as indicated by the parameter
43-49	XFIN(N,7)		NFINOR in field 64-66 on the Control-Flag
50-56	XFIN(N,8)		Card. The card identification XFINn may
57-63	XFIN(N,9)		be punched in card columns 73-80; where
64-70	XFIN(N,10)		n denotes the fin number.

Fin airfoil-ordinate card (10F7.0, 10X):

Column	Code	Routine Format	Explanation
1-7	FINORD(N,1)	AIRCFT	Table of fin-airfoil-thickness ordinates
8-14	FINORD(N,2)	10F7.0	as a percent of chord length at each
15-21	FINORD(N,3)		percent-chord-ordinate position for the
22-28	FINORD(N,4)		fin. Only one card is used and may con-
29-35	FINORD(N,5)		tain up to 10 numbers. The number of
36-42	FINORD(N,6)		percent-chord locations used must be the

Fin airfoil-ordinate card (Continued)

Column	Code	Routine Format	Explanation
43-49	FINORD(N,7)		same as indicated by the parameter NFINOR in field 64-66 on the Control-Flag Card. The card identification FINORDn may be punched in card columns 73-80, where n denotes the fin number.
50-56	FINORD(N,8)		
57-63	FINORD(N,9)		
64-70	FINORD(N,10)		

Note: A new set of all three fin-input cards is required for each fin (a maximum of 6 are provided for). Note that every fin uses the same value for the parameter NFINOR. If the Y-ordinates of the fin leading edge are input as 0.0 only half of the symmetrical-centerline fin is generated. If the Y-ordinates of the leading edge are not equal to 0.0 then the entire fin will be generated.

Canards or horizontal tails. -

Canards or horizontal tails are defined in a manner similar to that used for fins using an inboard airfoil and an outboard airfoil. Each airfoil lies in an X-Z plane at a fixed Y distance. The airfoils may be symmetrical or unsymmetrical. Both the top and bottom of the surface will be generated using the same procedures as outlined for the wing.

The input information required to define a canard or horizontal tail is as follows:

1. Number of canards (up to 2).

2. Number of ordinates used to define the canard airfoils (3 to 10).

3. The X-Y-Z coordinates of the inboard- and outboard-airfoil leading edges.

4. Chord lengths of the inboard and outboard airfoils.

5. A table of percent-chord locations that are to be used to define the airfoils.

6. A table of airfoil upper-surface ordinates as a percent of chord length. If the airfoil is not symmetrical another table contains the lower-surface ordinates.

The generation of canard and horizontal-tail surfaces is illustrated in Figure 13.

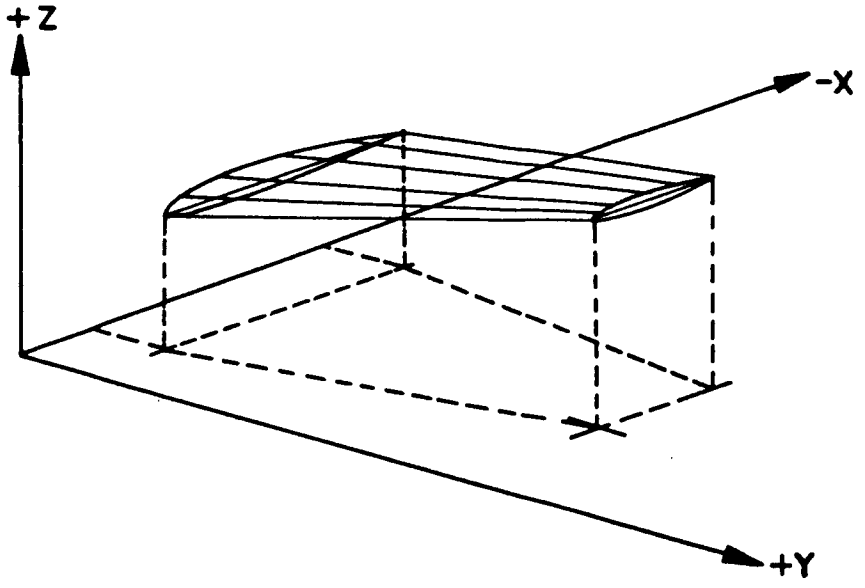


FIGURE 13. GENERATION OF CANARD AND HORIZONTAL-TAIL GEOMETRY

Canard leading-edge-coordinate card (8F7.0, 24X):

Column	Code	Routine Format	Explanation
1-7	CANORG(N,1,1)	AIRCFT	X-coordinate of inboard-airfoil leading edge.
8-14	CANORG(N,1,2)	8F7.0	Y-coordinate of inboard-airfoil leading edge.
15-21	CANORG(N,1,3)		Z-coordinate of inboard-airfoil leading edge.
22-28	CANORG(N,1,4)		Chord length of the inboard airfoil.
29-35	CANORG(N,2,1)		X-coordinate of outboard-airfoil leading edge.
36-42	CANORG(N,2,2)		Y-coordinate of outboard-airfoil leading edge.
43-49	CANORG(N,2,3)		Z-coordinate of outboard-airfoil leading edge.
50-56	CANORG(N,2,4)		Chord length of the outboard airfoil.
73-80			The card identification CANORGn may be punched in card columns 73-80, where n denotes the fin number.

Canard percent-chord-location card (10F7.0, 10X):

1-7	XCAN(N,1)	AIRCFT	Table of percent-chord locations that are
8-14	XCAN(N,2)	10F7.0	to be used for the airfoil-thickness
15-21	XCAN(N,3)		ordinates. Only one card is used and may
22-28	XCAN(N,4)		contain up to 10 numbers. The number of
29-35	XCAN(N,5)		percent-chord locations used must be the
36-42	XCAN(N,6)		same as indicated by the parameter NCANOR
43-49	XCAN(N,7)		in field 70-72 on the Control-Flag Card.
50-56	XCAN(N,8)		The card identification XCANn may be

Canard percent-chord-location card (Continued)

Column	Code	Routine Format	Explanation
57-63	XCAN(N,9)		punched in card columns 73-80, where n denotes the canard number.
64-70	XCAN(N,10)		

Canard upper-ordinate card (10F7.0, 10X):

This card also used for lower surface if canard is symmetrical.

1-7	CANORD(N,1)	AIRCFT 10F7.0	Table of canard-airfoil-thickness ordinates as a percent of chord length at each percent-chord-ordinate position. Only one card is used and may contain up to 10 numbers. The number of percent-chord locations used must be the same as indicated by the parameter NCANOR in field 70-72 on the Control-Flag Card. If the parameter NCANOR is positive (+) these airfoil ordinates will be used for both the top and bottom of the canard. The card identification CANORDn may be punched in card columns 73-80, where n denotes the canard number.
8-14	CANORD(N,2)		
15-21	CANORD(N,3)		
22-28	CANORD(N,4)		
29-35	CANORD(N,5)		
36-42	CANORD(N,6)		
43-49	CANORD(N,7)		
50-56	CANORD(N,8)		
57-63	CANORD(N,9)		
64-70	CANORD(N,10)		

If the canard airfoil is not symmetrical the bottom airfoil-thickness ordinates are input on the following card.

Canard lower-ordinate card (10F7.0, 10X):

This card is only used if the airfoil is not symmetrical (NCANOR = -).

Column	Code	Routine Format	Explanation
1-7	CANOR1(N,1)	AIRCFT 10F7.0	Table of canard-airfoil-thickness ordinates for the lower surface as a percent of chord length at each percent-chord ordinate position. Only one card is used and may contain up to 10 numbers. The number of percent-chord locations used must be the same as indicated by the parameter NCANOR in field 70-72 on the Control-Flag Card. The parameter NCANOR must be negative (-). Both the upper- and lower-thickness ordinates are input as positive percent-of-chord values. The card identification CANOR1n may be punched in card columns 73-80, where n denotes the canard number.
8-14	CANOR1(N,2)		
15-21	CANOR1(N,3)		
22-28	CANOR1(N,4)		
29-35	CANOR1(N,5)		
36-42	CANOR1(N,6)		
43-49	CANOR1(N,7)		
50-56	CANOR1(N,8)		
57-63	CANOR1(N,9)		
64-70	CANOR1(N,10)		

Canard lower-ordinate card (Continued)

Note: A new set of all canard cards is required for each canard (a maximum of 2 canards are provided for). Note that every canard uses the same value for the parameter NCANOR.

General-airfoil surfaces. -

This geometry-surface type may be used to generate surfaces that are defined by airfoil sections having arbitrary orientations in space. The airfoils are not confined to fixed planes as was the case for the wings, fins, and canards previously described. This more general approach permits the use of non-streamwise airfoil sections and is useful in describing intersecting components such as the wing- and tail-fuselage junctures. Input cards for this surface type cannot be used as input to the NASA Wave-Drag Program.

The general-airfoil surface is defined by connecting two or more airfoil sections with straight lines. The orientation of each airfoil is given by coordinates of the leading and trailing edges and an airfoil rotation angle. The techniques used in defining these airfoils and in performing the necessary transformation to obtain the required X-Y-Z coordinates in the vehicle-coordinate system are discussed below.

Each airfoil section is defined relative to a coordinate system fixed within the airfoil. The airfoil-thickness displacements may be measured either from the mean-camber line along a line perpendicular to the airfoil axis or on a line that is normal to the mean-camber line. This latter method is used in

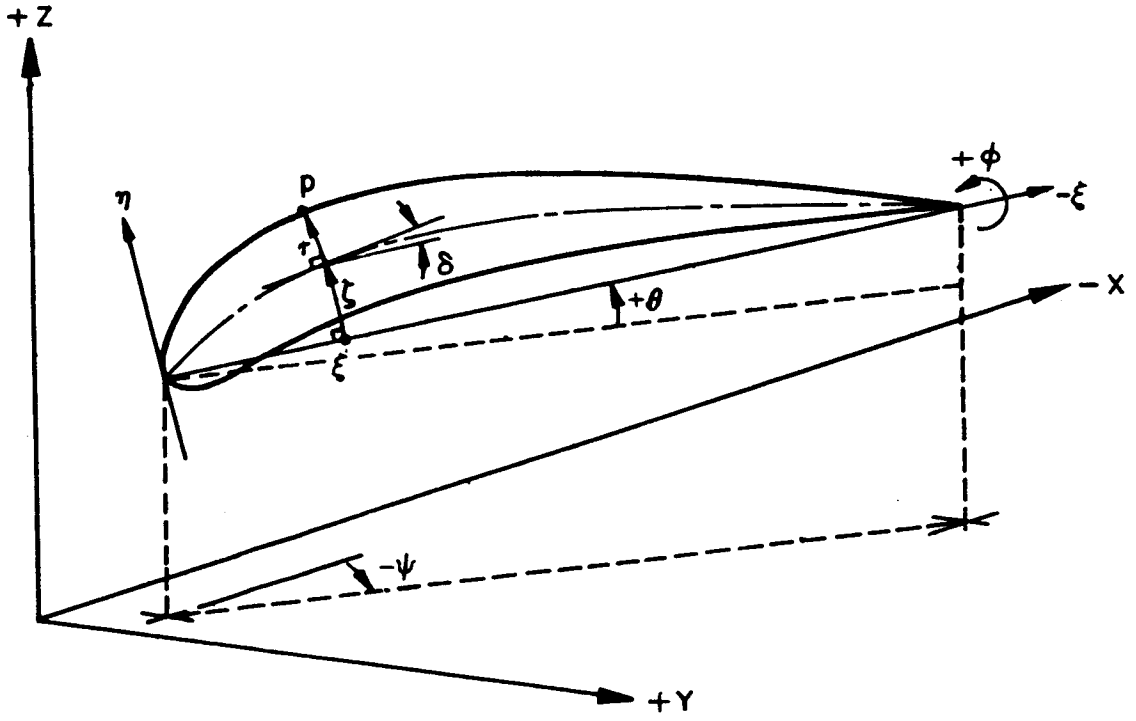


FIGURE 14. GENERAL AIRFOIL COORDINATE SYSTEM

General-airfoil surfaces (Continued)

some of the early NASA-airfoil documents. All airfoil-section parameters are expressed as a percent of the airfoil chord. The parameters used in defining an airfoil are illustrated in Figure 14. In this illustration the airfoil lies in the η-ξ plane.

The input information required by the Aircraft-Geometry Option to define a general airfoil surface is as follows:

1. Number of airfoils.

2. Number of airfoil percent-chord points used to define the airfoils.

3. Flags to control the thickness-distribution type, generation of tip- and root-closure elements, and repetitive use of mean-camber line and thickness distributions.

4. A table of percent-chord locations that are to be used for the airfoil-thickness and -camber distributions.

5. The X-Y-Z coordinates of the leading and trailing edge of each airfoil section.

6. The roll angle ϕ of each airfoil section.

7. The mean-camber-line ordinates in percent chord at each percent-chord location for each airfoil.

8. Thickness distribution in percent chord at each percent-chord position for each airfoil.

The roll angle ϕ is input explicitly and together with ψ and θ are positive in the right-handed sense of the reference system.

Zero values for the rotation angles indicate the airfoil is orientated parallel to the X-Z plane. Zero yaw and pitch angles and a +90 degree roll angle gives an airfoil in the X-Y plane (such as a vertical-tail-root airfoil).

This surface type differs from those previously described in that repetitive use may be made of the arbitrary-airfoil option on a single pass into the Aircraft-Geometry Option. This stacking option allows wings, fins, etc. to be generated on a single pass into the Aircraft-Geometry Option. A control flag also permits repetitive use of airfoil data for subsequent airfoils to save input time when all the surface airfoils are identical. Tip- and root-closure elements may also be generated to give a completely enclosed surface.

The arrangement of the generated X-Y-Z surface points and elements is similar to the procedure outlined for the wing surface with the exception of elements that may be generated to close the tip and root sections.

General-airfoil control-flag card (22X,I3, 2(3X,I2), 7(4X,I1),2X,A4,4X):

Column	Code	Routine Format	Explanation
1-22			Surface-description statement. Any acceptable machine characters. Not used by the program.
23-25	INSURF	AIRCFT I3	Control-Surface-Deflection Flag (= 0).
29-30	NAF	AIRCFT I2	Number of airfoils used to describe the surface. = 2 to 20.
34-35	NAFORD	AIRCFT I2	Number of percent-chord points used to define each airfoil section. = 3 to 30.
40	NCAM	AIRCFT I1	Flag for mean-camber-line distribution. = 0 No camber data will be input. = 1 Camber data will be input.
45	NACA	AIRCFT I1	Airfoil Thickness-Type Flag. = 0 Thickness will be calculated normal to the chord line. = 1 Thickness will be calculated normal to the mean-camber line. = 2 Same as = 0, but camber input as ΔZ.
50	ITIP	AIRCFT I1	Flag for closure surface at the tip. = 0 Do not generate tip-closure surface. = 1 Tip-closure surface will be generated.
55	IROOT	AIRCFT I1	Flag for closure surface at the root. = 0 Do not generate root-closure surface. = 1 Root-closure surface will be generated.
60	ISIMC	AIRCFT I1	Flag for similar camber-line distribution. = 0 Each airfoil camber line will be input. = 1 Camber-line distribution will be the same for all airfoils and need be input only for the first.
65	ISIMT	AIRCFT I1	Flag for similar thickness distribution. = 0 Each airfoil-thickness distribution will be input. = 1 Thickness distribution will be the same for all airfoils and need be input only for the first.

General-airfoil control-flag card (Continued)

Column	Code	Routine Format	Explanation
70	MORE	AIRCFT I1	Flag to indicate stacking of general-airfoil surfaces. = 0 This is the last surface. = 1 Another surface follows and a complete set of arbitrary-airfoil input cards is expected immediately following this set.
73-76	SURFID	AIRCFT A4	Surface identification to be punched in fields 73-76 of the Output-Element-Data Type 3 Cards. Any acceptable machine characters.

Airfoil percent-chord-location card(s) (10F7.0, 10X):

Column	Code	Routine Format	Explanation
1-7	XOC(1)	AIRCFT 10F7.0	Table of percent-chord locations that are to be used for the airfoil-thickness and camber distributions. Each card may contain up to 10 fields, if more are required continue with additional cards of the same format. Use as many fields as is specified by NAFORD (field 34-35) on the General-Airfoil Control-Flag Card. Maximum number of fields is 30 (3 cards).
8-14	XOC(2)		
15-21	XOC(3)		
22-28	XOC(4)		
29-35	XOC(5)		
36-42	XOC(6)		
43-49	XOC(7)		
50-56	XOC(8)		
57-63	XOC(9)		
64-70	XOC(10)		
1-7	XOC(11)		Fields 73-80 of each card may be used for identification.
etc.	etc.		

Airfoil-orientation cards (10F7.0, 10X):

Column	Code	Routine Format	Explanation
1-7	AFORG(I,1)	AIRCFT 10F7.0	X-coordinate of the airfoil leading edge.
8-14	AFORG(I,2)		Y-coordinate of the airfoil leading edge.
15-21	AFORG(I,3)		Z-coordinate of the airfoil leading edge.
22-28	AFORG(I,4)		X-coordinate of the airfoil trailing edge.
29-35	AFORG(I,5)		Y-coordinate of the airfoil trailing edge.
36-42	AFORG(I,6)		Z-coordinate of the airfoil trailing edge.
43-49	AFORG(I,7)		Airfoil roll angle ϕ in degrees.
73-80			May be used for identification.

Airfoil-orientation cards (Continued)

Note: Repeat this card for all airfoils, starting with the inboard-root airfoil and working to the outboard-tip airfoil. The number of these cards is given by the value of NAF (field 29-30) of the General-Airfoil Control-Flag Card and must not be greater than 20.

Airfoil camber-distribution cards (10F7.0, 10X):

Required only if NCAM = 1 on the General-Airfoil Control-Flag Card. If NACA = 2, use wing camber cards (see p. 47)

Column	Code	Routine Format	Explanation
1-7	AFCAM(I,1)	AIRCFT	Mean-camber line distribution, in percent-chord, at each percent-chord location XOC. Use as many cards as required with 10 numbers on each card. Each airfoil must have as many numbers as was specified by the parameter NAFORD (field 34-35) on the General-Airfoil Control-Flag Card (30 maximum). There will be as many sets of cards as given by the parameter NAF (field 29-30) on the General-Airfoil Control-Flag Card. The first value for each airfoil should start on a new card. If parameter ISIMC = 1 (field 60 of General-Airfoil Control-Flag Card), only one set of cards is required. Fields 73-80 may be used for identification.
8-14	AFCAM(I,2)	10F7.0	
15-21	AFCAM(I,3)		
22-28	AFCAM(I,4)		
29-35	AFCAM(I,5)		
36-42	AFCAM(I,6)		
43-49	AFCAM(I,7)		
50-56	AFCAM(I,8)		
57-63	AFCAM(I,9)		
64-70	AFCAM(I,10)		
1-7	AFCAM(I,11)		
etc.	etc.		

Airfoil thickness-distribution cards (10F7.0, 10X):

Column	Code	Routine Format	Explanation
1-7	AFORD(I,1)	AIRCFT	Airfoil thickness coordinates, in percent chord, at each percent-chord location XOC. Use as many cards as required with 10 numbers on each card. Each airfoil must have as many numbers as was specified by the parameter NAFORD (field 34-35) on the General-Airfoil Control-Flag Card (30 maximum). There will be as many sets of cards as given by the parameter NAF (field 29-30) on the General-Airfoil Control-Flag Card. The first value for each airfoil should start on a new card. If parameter ISIMT = 1 (field 65 of General-Airfoil Control-Flag Card), only one set of cards is required. Fields 73-80 may be used for identification.
8-14	AFORD(I,2)	10F7.0	
15-21	AFORD(I,3)		
22-28	AFORD(I,4)		
29-35	AFORD(I,5)		
36-42	AFORD(I,6)		
43-49	AFORD(I,7)		
50-56	AFORD(I,8)		
57-63	AFORD(I,9)		
64-70	AFORD(I,10)		
1-7	AFORD(I,11)		
etc.	etc.		

Airfoil thickness-distribution cards (Continued)

Column	Code	Routine Format	Explanation
			Note: Data are input as 1/2 of total thickness values (i.e., if $t/c_{total} = 0.04$, then input AFORD = t/c = 2.0%).

AERODYNAMIC PROGRAM INPUT DATA

The Aerodynamic portion of the program contains four major options:

1. Flow-Field Analysis
3. Inviscid Pressures
5. Special Routines
7. Shear

Access to these major options is obtained by use of the AERO Option on the System-Control-Card input to the Main-Executive Routine. All four of the above major options are controlled by an executive routine called AERO.

The input data to the AERO Executive Routine includes a set of flags (IPG) to determine the sequence of calls to the above four options. Up to 20 calls may be made to the various options on a single entry into the AERO-Executive Routine After the last non-zero option is executed the program will return to the Main-Executive Program. The basic flight conditions, vehicle reference dimensions, and the Angle-of-Attack and Yaw-Angle Cards are input to the AERO-Executive Routine and apply for all of the AERO Options called by the IPG commands.

Input to AERO-Executive Routine

Aero-system title and control card (10A4, 5A1, 2I1, 3XI3, 2XI4, 4I1, F9.4). -

Column	Code	Routine Format	Explanation
1-40	TITLE	AERO 10A4, 5A1	Title to be printed for AERO Option.
46	ISVFMS	AERO I1	Flag to save the forces, moments, and shears on Unit 20. = 0 Do not save data. = 1 Save new case on existing data set. = 2 Replace existing case on existing data set. = 3 Generate a new data set.
47	ISVSPL	AERO I1	Flag to save the Surface-Spline pressure-distribution data on Unit 21. = 0 Do not save data. = 1 Save new data on existing data set. = 2 Replace old data on existing data set. = 3 Generate a new data set.

Aero-system title and control card (Continued)

Column	Code	Routine Format	Explanation
51-53	IRUNID	AERO I3	Run ID number of the input data.
56-59	ICORID	AERO I4	Correlation ID number of the input data.
60	ICKPRT	AERO I1	Flag for special print of spline data and C_p data. = 0 No special print. = 1 Special print on. = 2 Detail print on element C_p.
61	IPLTCP	AERO I1	Flag to save C_p row data on Unit 16. = 0 Do not save data. = 1 Save new C_p data on existing data set. = 3 Generate a new data set.
62	IBETA	AERO I1	Beta-Case Flag. = 0 Not a beta case (β=0). = 1 Beta case at input value of beta (CC's 64-72).
63	IWRIT	AERO I1	Shear-Option Input and Special-Print Flag. = 0, = 1, = 2, = 3, = 4 } Special-Print Flag Levels; = 0, minimum print; = 4, most detailed print If IWRIT <5, predefined parameters will be used. If IWRIT >5, card input will be expected in the Shear Option and Print Option determined by the value of (IWRIT-5). For example, input IWRIT = 8 will indicate card input and the third level Special-Print Option.
64-72	BETA	AERO F9.4	Sideslip angle (β).

Aero-flag card (20I1, 5X, I2, I1). -

Column	Code	Routine Format	Explanation
1	IPG(1)	AERO 20I1	Aerodynamic sub-options to be used in the order in which they will be solved. Maximum of 20.
2	IPG(2)		
3	IPG(3)		= 1 Flow Field Analysis.
etc.	etc.		= 3 Inviscid Pressures.
20	IPG(2)		= 5 Special Routines.
			= 7 Shear.
26-27	NTAPE	AERO I2	(Not Active)
28	IREP	AERO I1	Next Empty Record Pointer Control Flag.
			= 0 Surface property data will be saved right after the geometry data on Unit 4 (if IFSAVE = 1 on the Pressure-Method Card). That is, the parameter NEXT is set equal to the next record after the geometry data. Previous Surface-Property Data saved will be lost. This is the normal program option.
			= 1 Surface-Property Data will be saved right after the end of the previously saved Surface-Property Data instead of right after the geometry data. Previous Surface-Property Data saved will not be lost. This option is used on the second entry into AERO when a different set of vehicle panels are to be analyzed, and if previously saved Surface-Property Data are not to be lost.

Flight-condition card (4F10.0, I1, I2, 7XF10.0). -

Column	Code	Routine Format	Explanation
1-10	MACH	AERO F10.0	Free-stream Mach number. (Not used if INPUTD = 0 on the Executive-Flag Card.)

Flight-condition card (Continued)

Column	Code	Routine Format	Explanation
11-20	ALT	AERO F10.0	Not used by MARK IVS, but is printed in the Inviscid-Pressures Output.
21-30	PSTAG	AERO F10.0	Not used by MARK IVS.
31-40	TSTAG	AERO F10.0	Not used by MARK IVS.
41	IGAS	AERO I1	Gas-Selection Flag. = 0 Air properties will be used. = 1 Helium properties will be used.
42-43	NAB	AERO I2	Number of α-β Cards to be read. A maximum of 20 cards are permitted.
51-60	AMBP	AERO F10.0	Engine by-pass mass flow, used for case identification only (not used if INPUTD = 0 on the Executive-Flag Card).

Reference-dimension card (6F10.0). -

Column	Code	Routine Format	Explanation
1-10	SREF	AERO F10.0	Reference area for the force coefficients (wing area). Must be in units consistent with input-scaled-geometry data.
11-20	MAC	AERO F10.0	Reference length to be used in pitching-moment calculations.
21-30	SPAN	AERO F10.0	Reference length to be used in rolling- and yawing-moment calculations.
31-40	XCG	AERO F10.0	Longitudinal position of center of gravity for moment calculations. Note that XCG will frequently be input as a negative number since the negative X-axis is usually taken as directed from the nose to the tail.
41-50	YCG	AERO F10.0	Lateral position of the center of gravity. Usually = 0.0.
51-60	ZCG	AERO F10.0	Vertical position of the center of gravity.

α-β cards (7F10.0). -

The number of these cards to be input is controlled by the parameter NAB. The complete set of α-β Cards is assumed to be used by all the AERO Options unless the various options specify otherwise.

Column	Code	Routine Format	Explanation
1-10	ALPHA(I)	AERO F10.0	Vehicle angle of attack (α), deg. (Not used if INPUTD = 0 on the Executive-Flag Card).
11-20	BETA(I)	AERO F10.0	Vehicle sideslip angle (β). Positive with the wind striking the right side of the vehicle, deg. (Not used if INPUTD = 0 on the Executive-Flag Card).
21-30	ROL(I)	AERO F10.0	Not used by MARK IVS.
31-40	CDELTA(I)	AERO F10.0	Not used by MARK IVS.
41-50	QI(I)	AERO F10.0	Not used by MARK IVS.
51-60	RI(I)	AERO F10.0	Not used by MARK IVS.
61-70	PI(I)	AERO F10.0	Not used by MARK IVS.

Note: The above parameters are stored in data arrays. A maximum of 20 conditions are permitted.

Flow-Field Program-Input Data

The Flow-Field Program is used to load the pressure-coefficient (Cp) data for use in calculating the forces. These data are stored on the Flow-Field Data Storage Unit (10) using mass-storage (random access) techniques. The unit contains several directory tables to provide the necessary pointers to each level of data. These include a Master Directory, an α-β Directory for each data set (Mach number) that provides the pointers to each α-β set, a Region Directory Table for each new α-β set that points to the Flow-Field Table, which in turn points to the surface C_p data. It is important to be thoroughly familiar with the way that these tables are generated and used. Subroutine FLOW should be studied to obtain this knowledge. The general manner in which the flow-field information is stored is summarized in Section 5.

Flow-Field Data

The input cards described below control the generation and use of the Flow-Field-Storage Unit Directories and Tables and the pointers that they contain. In general, options are provided for either reading data already stored on the unit and printing them on the output unit for inspection, or the actual loading of the flow-field data. The cards described below are concerned with the generation and use of the directory tables. Subsequent input-card descriptions will give the input requirements for each of the methods actually used in loading or generating the flow-field data itself using the directory tables that have been prepared.

Master-directory title card (1I1, 19X, 10A4). -

Column	Code	Routine Format	Explanation
1	MFLAG	FLOW I1	Master-Directory-Status Flag. = 0 Set up complete new Master-Directory Table. = 1 Directory Table already exists, so just use old pointers.
21-60	TITLEM	FLOW 10A4	Master-Directory Title. This title is only used when MFLAG = 0.

Master-directory data-set card (2I1, 1X, I1, F6.0, 10X, 10A4). -

This card is used to establish the different sets of data; i.e., different Mach numbers.

Column	Code	Routine Format	Explanation
1	LASTS	FLOW I1	Last-Set Flag. = 0 This is not the last set of data. = 1 This is the last set of data.
2	NEWS	FLOW I1	New-Set-Directory Flag. = 0 Set up a new Set-Directory Table. = 1 The desired Set Directory already exists so just read the pointers to get the desired set. = 2 A new Set Directory will be added and the Master Directory Pointer updated.
4	NSET	FLOW I1	Data-Set Number. Maximum of 5 permitted. (must be = 1 if NEWS = 0)

Master-directory data-set card (Continued)

Column	Code	Routine Format	Explanation
5-10	MACH	FLOW F6.0	Mach number. This does not have to be the same as was input to AERO.
21-60	TITLES	FLOW 10A4	Set Directory Title. This title is used only when NEWS = 0.

α-β set directory-table card (2I1, I2, 2F6.0, 4X, 10A4). -

Column	Code	Routine Format	Explanation
1	LASTAB	FLOW I1	Last α-β Set Flag. = 0 This is not the last α-β set. = 1 This is the last α-β set.
2	NEWAB	FLOW I1	New α-β Set Flag. = 0 Set up a new α-β Set Directory Table. = 1 The α-β Set Directory Table already exists so just read the pointers to get to the desired α-β set. = 2 A new α-β set will be added to the set table that already exists.
3-4	IAB	FLOW I2	α-β Set Number. Maximum of 20 permitted.
5-10	ALPHA	FLOW F6.0	Angle of attack, α. This does not have to be the same as was input to AERO.
11-16	BETA	FLOW F6.0	Yaw angle, β. This does not have to be the same as was input to AERO.
21-60	TITLEA	FLOW 10A4	Title for the α-β Set Directory Table. This title is only used when NEWAB = 0.

Region-directory-table card (2I1, I2, 4I1, 12X, 10A4). -

Column	Code	Routine Format	Explanation
1	LASTR	FLOW I1	Last-Flow-Region Flag. = 0 This is not the last-flow region. = 1 This is the last-flow region.
2	NEWR	FLOW I1	New-Region-Set Flag. = 0 Set up a new-region-set directory. = 1 The Region-Set-Directory Table already exists so just read the pointers to get to the desired flow region.

Region-directory-table card (Continued)

Column	Code	Routine Format	Explanation
			= 2 A new flow region will be added to the Region-Set-Directory Table that already exists.
3-4	IREG	FLOW I2	Flow-Region Number.
5	IDTYP(1)	FLOW I1	Data-Type Flag. = 2 Surface-property data will be read or loaded. Note: When accessing an existing data set, IDTYP(1) is overwritten by a value on the data set.
6	ISORCE	FLOW I1	Not used by MARK IVS.
7	IRW	FLOW I1	Data Read or Write Flag. = 0 Data will be input or generated by one of the sub-options and the flow-field data saved on the Storage Unit (10). = 1 Data will be read from the Flow-Field-Data Save Unit (10) and written out on the output file.
21-60	TITLER	FLOW 10A4	Title for the Region-Directory Table. This title is only used when NEWR = 0.

Surface-Data-Transfer Option

This option is used to hand load surface data on Unit 10. This option may also be used to transfer surface data (that has been previously stored by FORCE) from Unit 4 to Unit 10. The Surface-Data-Transfer Option is exercised only if IDTYP(1) = 2 on the Region-Directory-Table Card. If IRW=1, then this option may be used to read and print out surface data previously stored on Unit 10.

The surface data (i.e., pressure coefficients) are stored on Unit 10 and will be recalled for use in the surface-spline interpolation routine to determine the values at the element centroids in order to calculate the forces. The surface spline, or any other interpolation scheme, needs to be specified in appropriate coordinates to do the job correctly. For example, consider the flow over a swept wing. Interpolation relative to the space coordinates used to define the quadrilateral elements will produce erroneous results. The

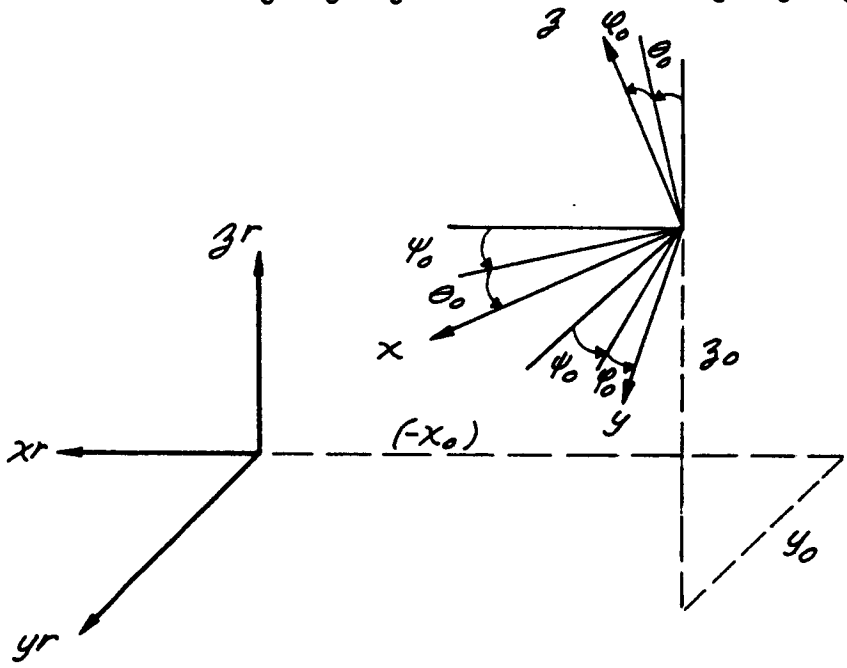
interpolation must be done in coordinates consistent with the physics of the problem and for flow on a swept wing, distance from the leading edge and distance along the span would be proper.

In addition, the surface spline, due to the nature of the basic solution (symmetric point load), works best in a one-to-one domain of the independent variables. For the swept wing, coordinates of relative chord (x/c) and relative span (y/b) would be used to define a range from 0.0 to 1.0 for both variables.

The process of selecting and scaling the appropriate coordinates is referred to as normalization. The local coordinate system is defined at this time on the Flow-Field Control and Plane-Orientation Card, and the choice of independent variables and scaling are indicated on the Input-Pressure-Option Cards, described later. The normalization procedure is described as follows.

Surface-data normalization. -

The surface data, input in body reference coordinates (xr, yr, zr) is first transformed to the required local system (x, y, z). The local system is defined by a translation (X_o, Y_o, Z_o) and a rotation (ψ_o, θ_o, ϕ_o).



The local coordinates (x, y, z) are given by

$$[x, y, z] = [T]\,[xx, yy, zz]$$

where

$$[T] = \begin{bmatrix} (\cos\theta_o\cos\psi_o) & (\cos\theta_o\sin\psi_o) & (-\sin\theta_o) \\ (-\cos\theta_o\sin\psi_o+\sin\phi_o\sin\theta_o\cos\psi_o) & (\cos\phi_o\cos\psi_o+\sin\phi_o\sin\theta_o\sin\psi_o) & (\sin\phi_o\cos\psi_o) \\ (\sin\phi_o\sin\psi_o+\cos\phi_o\sin\theta_o\cos\psi_o) & (-\sin\phi_o\cos\psi_o+\cos\phi_o\sin\theta_o\sin\psi_o) & (\cos\phi_o\cos\theta_o) \end{bmatrix}$$

and

$$xx = xr - x_0$$
$$yy = yr - y_0$$
$$zz = zr - z_0$$

Also calculated are the axial, radial, and meridian coordinates:

$$A = x$$

$$R = (y^2 + z^2)^{1/2}$$

$$\phi = \text{ARCTAN } (y/-z)$$

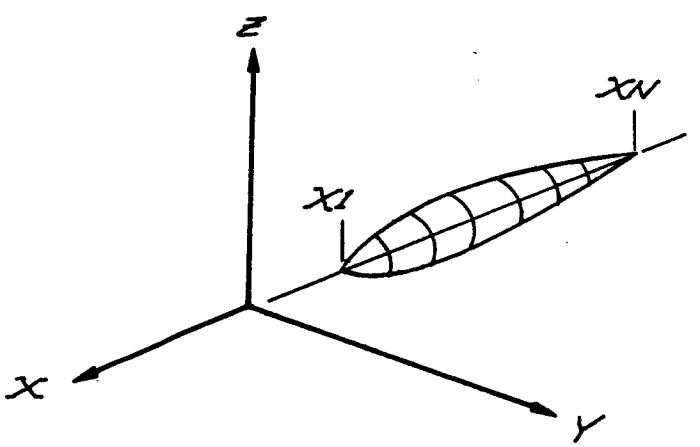
Six coordinates (x, y, z, A, R, ϕ) are now available in the local reference system and the pair of independent variables to be used for interpolation are selected by the input flag INORM. The five options available are;

INORM = 0,	$\phi = f(A, R)$
INORM = 1,	$z = f(x, y)$
INORM = 2,	$y = f(x, z)$
INORM = 3,	$x = f(y, z)$
INORM = 4,	$R = f(A, \phi)$

To scale the data, the surfaces are grouped into two types: bodies and lifting surfaces (indicated by the input flag ISURF = 0 and 1, respectively). More complex surfaces may be composed from combinations of these two types. In addition, four boundary points must be input to scale the data. These points are input in the body-reference system and transformed to the six local coordinates. The boundary data are used differently for each surface type. An example of the use of the boundary data for each of the two surface types is discussed below.

I. Bodies (ISURF = 0)

Consider a fuselage with approximately an axially-symmetric cross-section shape about the X-axis.



For this case the obvious choice of independent variables is the axial coordinate (A) and the meridian angle (ϕ) (i.e., INORM = 4).

The following boundary data are input:

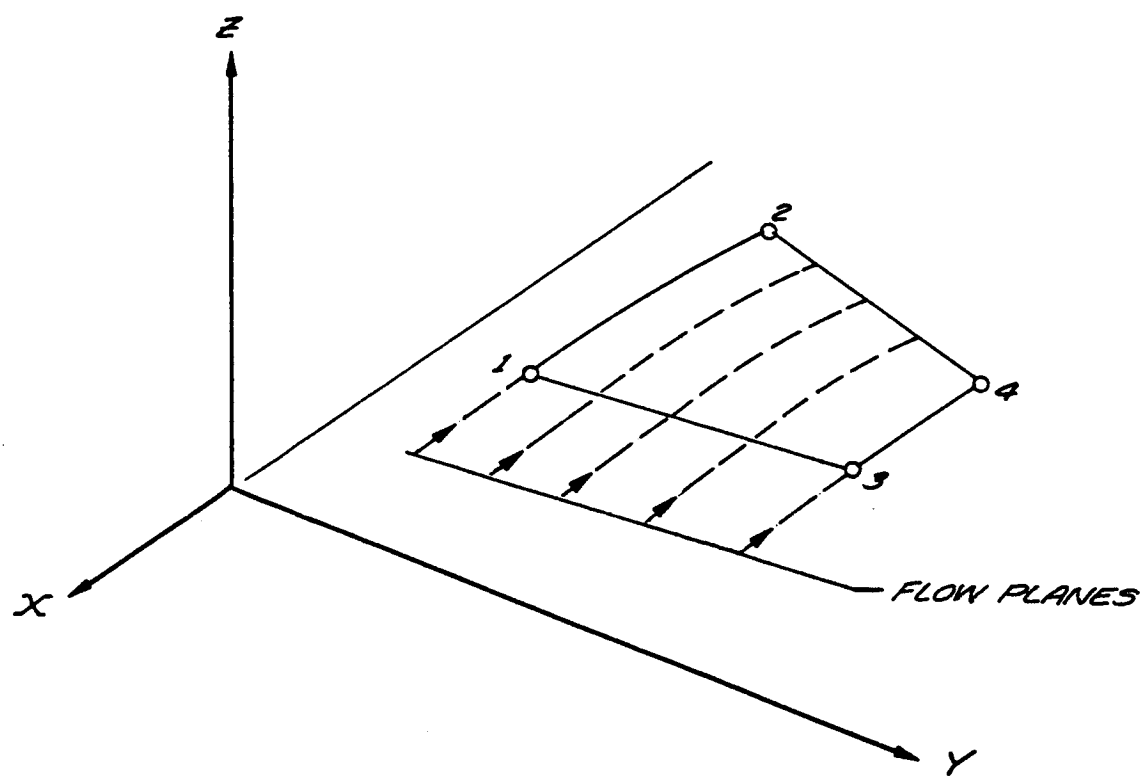
```
XB(1) = X1    XB(2) = XN
YB(1) = 0.0   YB(2) = 0.0
ZB(1) = 0.0   ZB(2) = 0.0

XB(3) = 0.0   XB(4) = 0.0
YB(3) = 0.0   YB(4) = 0.0
ZB(3) = -1.0  ZB(4) = 1.0
```

The interpolation will be performed over the length of the body ($X1 \leq X \leq XN$) and for ($0 \leq \phi \leq \pi$).

If there happen to be large variations in flow properties between $\phi = 0$ and $\phi = \pi$ (because of large α, or M_∞), then the interpolation could be segmented. For example, use three segments with ranges ($0.0 \leq \phi_1 \leq 60°$), ($60° \leq \phi_2 \leq 120°$) and ($120° \leq \phi_3 \leq 180°$). Since the surface spline is a global fit, segmenting will relax the constraints that must be met.

II. Lifting Surface (ISURF = 1)

Consider a wing whose leading edge and trailing edge are approximately in the x, y plane. The flow properties on a lifting surface vary essentially with relative chord (x/c) and relative span (y/b) and the surface should be normalized with respect to these parameters. Therefore, use INORM = 1.



The four boundaries are input as indicated on the sketch. They are first transformed to local coordinates and the following parameters calculated:

Root Chord, CR = XB(2) - XB(1)
Tip Chord, CT = XB(4) - XB(3)
Local Span, B = YB(3) - YB(1)

The normalized coordinates (x/c, y/b) for a given point on the surface (x, y, z) are

x/c = (x - XLE)/CY
y/b = (y - YB(1)/B

where XLE = XB(1) + [XB(3) - XB(1)] * y/b

and CY = CR + (CT - CR) * y/b

Flow-field control and plane-orientation card (I1, I4, I1, 4X, 6F10.0). -

This card is input only if IDTYP(1) = 2 and IRW = 0.

Column	Code	Routine Format	Explanation
1	NSREG	FFSURF I1	Total number of sub-regions to be hand loaded or transferred from Unit 4 to Unit 10. (Assumed at least = 1).
2-5	IFC1	FFSURF I4	Number of data points to be read from the input unit (hand loading of surface data). If IFC1 input = 0, then the data will be read from Unit 4 and loaded on to Unit 10.
6	ITFLAG	FFSURF I1	= 0 Surface-property data will be input. = 1 Surface-pressure-coefficient data will be input.
11-20	DAT(1)	FFSURF F10.0	$X_{\emptyset}$
21-30	DAT(2)	FFSURF F10.0	$Y_{\emptyset}$
31-40	DAT(3)	FFSURF F10.0	$Z_{\emptyset}$
41-50	DAT(4)	FFSURF F10.0	$PSI_{\emptyset}$

Flow-field control and plane-orientation card (Continued)

Column	Code	Routine Format	Explanation
51-60	DAT(5)	FFSURF F10.0	$THET_{\emptyset}$
61-70	DAT(6)	FFSURF F10.0	$PHI_{\emptyset}$

Hand-loaded surface-pressure-coefficient-data cards (4F10.0). -

These cards are used to hand load surface-pressure-coefficient (Cp) data directly on the Flow-Field Unit 10. These cards are input only if IFC1≠0, ITFLAG = 1, and INPUTD=0 (in Executive-Flag Card). Each card contains the x, y, z coordinates and Cp. This card is repeated until the required number of data points has been read-in as specified by the IFC1 counter.

Column	Code	Routine Format	Explanation
1-10	X	FFSURF F10.0	X-coordinate of the surface point.
11-20	Y	FFSURF F10.0	Y-coordinate of the surface point.
21-30	Z	FFSURF F10.0	Z-coordinate of the surface point.
31-40	C_p	FFSURF F10.0	Pressure coefficient of the surface point.

Hand-loaded surface-property-data cards. -

These cards are used to hand load surface-property data directly on the Flow Field Unit 10. These cards are input only if IFC1 ≠ 0 and ITFLAG = 0. Two cards are required for each data point. The first card contains the X, Y, Z coordinates and the second card has the surface-velocity-vector data. Each pair of cards is repeated until the required number of data points has been read in as specified by the IFC1 counter.

Surface-data-coordinate card (6F10.0):

Column	Code	Routine Format	Explanation
1-10	DATA(1)	FFSURF F10.0	X-coordinate of the surface-data point.
11-20	DATA(2)	FFSURF F10.0	Y-coordinate of the surface-data point.

Surface-data- coordinate card (Continued)

Column	Code	Routine Format	Explanation
21-30	DATA(3)	FFSURF F10.0	Z-coordinate of the surface-data point.
31-40	DATA(4)	FFSURF F10.0	(not used)
41-50	DATA(5)	FFSURF F10.0	(not used)
51-60	DATA(6)	FFSURF F10.0	(not used)

Surface-data-property card (6F10.0):

Column	Code	Routine Format	Explanation
11-10	DATA(7)	FFSURF F10.0	Surface Mach number.
11-20	DATA(8)	FFSURF F10.0	X-direction cosine component of the surface-velocity vector.
21-30	DATA(9)	FFSURF F10.0	Y-direction cosine component of the surface-velocity vector.
31-40	DATA(10)	FFSURF F10.0	Z-direction cosine component of the surface-velocity vector.
41-50	DATA(11)	FFSURF F10.0	P/P_∞ at the surface point.
51-60	DATA(12)	FFSURF F10.0	T/T_∞ at the surface point.

If this option is used, then the number of sets of these data furnished after the Flow-Field Control and Plane-Orientation Card must be = NSREG.

Surface-data panel-selection card (10I2). -

This card is used when surface data are to be transfered from Unit 4 to Unit 10 by the FFSURF Routine. This card is input if IFC1 = 0 on the Flow-Field-Control and Plane-Orientation Card. The parameters input on the Surface-Data Panel-Selection Card identify the geometry panel numbers on Unit 4 (that also have had surface data stored by the FORCE Routine) that are to be grouped together to form the surface data region. In subsequent calculations each surface sub-region set of data are fit at one time with the Surface-Spline Routine for interpolation purposes. It

is therefore important that the geometry panels grouped together form a regular surface (one that does not have rapid local changes in character that might fool the Surface-Spline Routine and give bad interpolation values).

Column	Code	Routine Format	Explanation
1-2	IPANL(1)	FFSURF I2	Panel number of geometry data on Unit 4 (including surface-velocity-vector data) to be grouped to form a sub-region on the Flow Field Storage Unit 10. A total of 10 panels may be used for one sub-region.
3-4	IPANL(2)	FFSURF I2	
5-6	IPANL(3)	FFSURF I2	
7-8	IPANL(4)	FFSURF I2	
9-10	IPANL(5)	FFSURF I2	
11-12	IPANL(6)	FFSURF I2	
13-14	IPANL(7)	FFSURF I2	
15-16	IPANL(8)	FFSURF I2	
17-18	IPANL(9)	FFSURF I2	
19-20	IPANL(10)	FFSURF I2	

If this option is used, then the number of these cards is equal to NSREG.

Pressure-Calculation Program-Input Data

The Pressure-Calculation Program is reached by way of sub-option calls from the Aero-Executive Routine. The sequence of calls to the FORCE Routine and the saving and summation of force data is accomplished in routine PRES. The geometry data for use by the PRES and FORCE Routines must be stored and available on the Quadrilateral-Element-Storage Unit (4). The Geometry Options must be used to accomplish this storage. An input to routine PRES specifies how the vehicle Panels are to be grouped to form vehicle Components for the force analysis. These selections are made by using the Panel Sequence Numbers assigned by the Geometry Program (the Panels are numbered in the order in which they are placed on the Quadrilateral-Element-Storage Unit (4)).

Title card and basic flags (I2, I1, 3X, 15A4). -

Column	Code	Routine Format	Explanation
1-2	NCOMP	PRES I2	Total number of vehicle Components to be analyzed. Each Component may consist of one or more vehicle Panels. The grouping of Panels to form Components is controlled by the Component-Organization Card below.
3	IFSAVE	PRES I1	Force-Data-Save Flag. = 0 Set up a new Unit 9 Force-Data Save File. Save component force data. = 1 Save force data. Use existing Unit 9 Force-Data File and just add a new set to it. = 2 Do not save force data.
7-66	TITLE	PRES 15A4	Title to be printed out at the top of the force-data output.

Component-organization card (10I2, 3I1). -

Column	Code	Routine Format	Explanation
1-2	IPANL(1)	PRES 10I2	The identification numbers for all of the Panels on the Quadrilateral-Element-Storage Unit (4) that are to be grouped together to form this vehicle Component. Up to a maximum of 10 Panels may be grouped together to form a Component. All the elements in a given Component will be analyzed using the same pressure-calculation method.
3-4	IPANL(2)		
5-6	IPANL(3)		
etc.			
19-20	IPANL(10)		

Note: The number of sets of Component-Organization, Pressure-Method Cards, and Input-Pressure Option Cards must be = NCOMP.

21	IPM	PRES I1	Pressure-Method-Card Flag. = 0 Pressure-Method Cards will be read for each α-β. = 1 Read only one Pressure-Method Card and assume that it will apply for all the α-β's for this component. = 2 Use the same Pressure-Method-Data Set as was used for the previous vehicle Component. No Pressure-Method Cards will be input.

Component-organization card (Continued)

Column	Code	Routine Format	Explanation
22	INT	PRES I1	Not used by MARK IVS.
23	ISHEF	PRES I1	Not used by MARK IVS.

Pressure-method cards (2I2, 3I1, 3X, 6F10.0). -

The Pressure-Method Cards are used to specify what pressure-calculation methods are to be used for each vehicle component. The only method active in the MARK IVS program is the Input Cp Option IPIN = 1, cc6). The number of Pressure-Method Cards to be input is controlled by the parameter IPM on the Component-Organization Card. If IPM = 0, then the number of Pressure-Method Cards must be equal to the parameter NAB as input on the Flight-Condition Card to the AERO-Executive Routine.

Column	Code	Routine Format	Explanation
1-2	IMPACT	PRES I2	Not used by MARK IVS.
3-4	ISHAD	PRES I2	Not used by MARK IVS.
5	IPRINT	PRES I1	Print Flag. This flag controls the printing of the detailed force characteristics of each vehicle element. = 0 Do not print detailed element-force data. = 1 Print detailed force contributions for each element (a large amount of output will be produced and machine time will increase). = 2 Print detailed local-property calculation and iteration results.
6	IPIN	PRES I1	Non-uniform Input Cp-Table Flag. = 0 Input Cp Table will not be used. = 1 Input Cp Table will be used. Input Pressure-Option Cards will be input. This option may be used to input wind-tunnel-pressure data in order to obtain resultant vehicle forces.

Pressure-method cards (Continued)

Column	Code	Routine Format	Explanation
7	ISAVE	PRES I1	Save Surface-Property-Data Flag. = 0 Do not save the surface-property data (local-surface Mach number, P_T/P_∞, T/T_∞, etc.). = 1 Save surface-property data on Unit 4. This option must be used to store data that is required by the Shear Option. These data may also be used in the Surface-Data-Transfer Option.
11-70	PDATA	PRES 6F10.0	Not used by the MARK IVS.

Input-pressure option. -

The Input-Pressure-Option Cards are used when the vehicle component forces are to be calculated using pressure data previously stored on the Flow-Field-Data Unit 10. For example, the forces on a particular component may be calculated using experimental results which have been previously stored on Unit 10 by use of the Flow-Field-Data Hand-Load Option.

Pressure data are available at a limited number of discrete locations on the component. The forces are calculated by summing the contributions of all the elements that make up the component. The function of the Input-Pressure Option is to obtain the value of pressure at the centroid of each element. This is accomplished by interpolation using the Surface-Spline Method and, as in the other applications of this method, proper normalization of the coordinates is required to obtain meaningful results (see the section on Surface-Data-Transfer Option for a discussion of surface-data normalization). The use of the Input-Pressure Option is controlled by the parameter IPIN on the Pressure-Method Cards. If IPIN = 0, no Input-Pressure Cards are read: If IPIN = 1 then a set of Input-Pressure Option Cards will be required. Since the parameter IPIN is used for each α-β, a set of Input-Pressure Cards will be needed for each α-β for which IPIN = 1.

A set of Input-Pressure-Option Cards consists of five cards; a Surface-Property Access-Flag Card and four Normalization-Data Cards.

Surface-property access-flag card (I1, 3I2, 3I1, 10I2, 5I1, 4X, I4):

Column	Code	Routine Format	Explanation
1	LASTR	CPINPT I1	(Not active)
2-3	NDSET	CPINPT I2	Data-Set Number where surface-data properties will be found on Unit 10.
4-5	IABSET	CPINPT I2	α-β Set Number where surface-data properties will be found on Unit 10.
6-7	IR	CPINPT I2	Flow-Region Number where surface-data properties will be found on Unit 10.
8	INORM	CPINPT I1	Normalization Flag for surface-property spline interpolation. The surface-property data are stored at varying X, Y, Z locations. However, it is usually best to convert the X-Y-Z coordinates to some combination of other coordinates for the spline interpolation (axial distance, A; radial distance, R; radial angle, ϕ). For example, on a body of revolution A and ϕ would be best. For a wing X and Y. For a vertical tail, X-Z. The available flags are: = 0 ϕ = fn(A,R) = 1 Z = fn(X,Y) (i.e., wings) = 2 Y = fn(X,Z) (i.e., vertical tails) = 3 X = fn(Y,Z) = 4 R = fn(A,ϕ) (i.e., body of revoluation)
9	ISURF	CPINTP I1	Surface-Boundary-Normalization Flag. This flag determines how the normalization data input on the next card will be used by the program in the normalization for the spline interpolation and for establishing the boundary limits for the interpolation.

Surface-property access-flag card (Continued)

			= 0 The input values for XB(1), YB(1), ZB(1) and XB(2), YB(2), ZB(2) will be used to establish the normalization limits for the first variable (i.e., if INORM = 4, they will determine the limits of the axial distance, A). The input values for XB(3), YB(3), ZB(3) and XB(4), YB(4), ZB(4) will be used to establish the normalization limits for the second variable (i.e., if INORM = 4, they will determine the limits of the angular coordinate, ϕ).
			= 1 The surface is assumed to be of a wing or tail type and the XB, YB, ZB normalization limits input on the next cards will be used to establish a chordwise-spanwise normalization. The point XB(1), YB(1), ZB(1) is the leading-edge point of the root chord, and the point XB(2), YB(2), ZB(2) is the trailing edge at the root. The point XB(3), YB(3), ZB(3) is the tip leading edge, and XB(4), YB(4), ZB(4) is the tip trailing edge.
10	IPF	CPINPT I1	Primary-Flow Flag. Not used in present program. IPF is set = 0 by the program.
11-12 13-14 15-16 17-18 19-20 21-22 23-24 25-26 27-28 29-30	ISR(1) ISR(1) ISR(1) ISR(1) ISR(1) ISR(1) ISR(1) ISR(1) ISR(1) ISR(10)	CPINPT 10I2	Subregion-Flow Number where surface-data properties will be found on Unit 10. If ISR(1) = 0, all subregions associated with flow-region IR (CC 6-7) will be used.
31	ISF(1)	CPINPT 5I1	Secondary Flow Number where surface data will be found on Unit 10.
32 33 34 35	ISF(2) ISF(3) ISF(4) ISF(5)		The Secondary Flow Numbers must be input to be considered and if indicated, will be used with both IPF = 0 and IPF = 1.

Surface-property access-flag card (Continued)

Column	Code	Routine Format	Explanation
40-43	NN	CPINPT I4	Flag and counter for special Cp data interpolation of fitted region. = 0 No special interpolation. = 1 Cp interpolation at coordinates of row data previously stored on Unit 15. ≥ 2 Cp values will be interpolated and printed for an (NNXNN) array points determined in the normalized coordinates. ≤-1 Cp values will be interpolated and printed for |NN| input coordinates.

Normalization-data cards (3F10.0):

Four cards are required that contain the coordinates for the data-normalization-limit process in the spline interpolation. The meaning of the four points is determined by the parameter ISURF on the preceeding card.

Column	Code	Routine Format	Explanation
1-10	XB(i)	CPINPT F10.0	XB coordinate
11-20	YB(i)	CPINPT F10.0	YB coordinate
21-30	ZB(i)	CPINPT F10.0	ZB coordinate
	i = 1 to 4		

Special Cp Interpolation At Previously-Stored Coordinates

This option is used only when NN = 1 on the Surface-Property Access-Flag Card and when IPLTCP ≠ 0 on the Aero-System Title and Control Card. The coordinate data is arranged as contours or rows on the vehicle surface and must have been loaded and stored on Unit 15 prior to the MARK IVS run. The interpolated Cp values are stored on Unit 16 for subsequent access by a plotting program. The Cp values will be printed if ICKPRT ≠ 0 on the Aero-System Title and Control

Card. The coordinates used are determined by a geometry-panel number and a row number which are input on two cards. More than one panel may be specified as applicable to the pressure region (i.e., spline fit) in question, in which case these two cards are repeated. This is controlled by the parameter MOREP. If the requested point is not within the region boundaries, the C_p is set = 999.999.

Panel-number card (3I5). -

Column	Code	Routine Format	Explanation
1-5	NPAN	CPINPT I5	Panel number.
6-10	NR	CPINPT I5	Number of rows in NPAN to be considered.
11-15	MOREP	CPINPT I5	Flag for additional panels. = 0 This is the last panel. = 1 This is not the last panel.

Row-sequence-number card (14I5). -

Column	Code	Routine Format	Explanation
1-5	IRW(1)	CPINPT I5	Row sequence number in panel NPAN from which coordinates for C_p interpolation will be determined. NR values will be read (maximum of 14).
6-10	IRW(2)	CPINPT I5	
.	.	.	
.	.	.	
.	.	.	
66-70	IRW(14)	CPINPT I5	

Note: If MOREP = 1, another set of these two cards will be expected next.

Special C_p-interpolation input-coordination cards (3F10.3, I3, 3I4). -

This option is used only if NN<0 on the Surface-Property Access-Flag Card. One coordinate per card is input and the number of cards expected = |NN| . If the point is not within the region boundaries, the parameter IFLG is set = -1.

Column	Code	Routine Format	Explanation
1-10	XX	CPINPT F10.3	X-coordinate.

Special C_p-interpolation input-coordination cards (Continued)

Column	Code	Routine Format	Explanation
11-20	YY	CPINPT F10.3	Y-coordinate.
21-30	ZZ	CPINPT F10.3	Z-coordinate.
55-57	IFLG	CPINPT I3	= 0
62-65	IRØW	CPINPT I4	Row number.*
69-72	JJ	CPINPT I4	Point number in row.*
77-80	IPAN	CPINPT I4	Panel number containing point.*

*The last three items are used to correlate with coordinates stored on Unit 15 and are not essential inputs. They may be left blank.

Special-Routines-Option-Input Data

The Special-Routines Section of the program is reached by way of suboption calls from the AERO-Executive Routine. The MARK IVS has only one option available, the Summation Routine. The basic purpose of the Summation Routine is to add together the aerodynamic-force coefficients of the separate vehicle components as calculated in the Force Program. The components to be added are selected by component number (sequence number in which they were placed on to the Force-Data Unit 9). This option may also be used to punch force data from the Force-Data-Save Unit (9) and/or to print out the force data.

Special-option-selection card (20I1). -

Column	Code	Routine Format	Explanation
1	IPG(1)	SPEC 20I1	Special-Routines sub-options to be used in the order given.
2	IPG(2)		
3	IPG(3)		= 1 Summation Program will be called. This program may be used to add, print and punch force data saved on the Force-Data-Save Unit (9).
.	.		
.	.		
.	.		

Special-option-selection card (Continued)

Column	Code	Routine Format	Explanation
	etc.		
20	IPG(20)		

Summation-routine option:

This routine may be used to selectively sum, print, and punch data that has been stored on the Force-Data-Save Unit (9) by the Force Routine. The results of the summations may also be stored on the Force-Data-Save Unit. This is the only option provided in the program for adding together the force contributions of different vehicle components.

A maximum of 40 vehicle components may be included in a particular summation. However, previously saved particular summations may also be recalled for summation (ISUM=2). In this latter case, the desired summation sequence numbers are input as the ICOMP parameters. Two cards are required for each summation even if the second card is blank.

Summation-control cards (5I1, I4, 20I2/20I2):

First card.

Column	Code	Routine Format	Explanation
1	LAST	SUM I1	Last Control-Card Flag. = 0 This is not the last control card. Another control card will be expected next. = 1 This is the last control card. Return to the AERO routine after this card set of sum/print/punch instructions is completed.
2	ISUM	SUM I1	Summation Flag. = 0 Do not sum up component forces. = 1 Sum up all the vehicle components as specified by the ICOMP parameter. = 2 Sum up previously saved partial sums.
3	ISAVE	SUM I1	Data-Save Flag. = 0 Save summation data on the Force-Data-Save Unit (9). = 1 Do not save summation data.

Summation-control cards (Continued)

Column	Code	Routine Format	Explanation
4	IPRINT	SUM I1	Print-Control Flag. = 0 Print out summmation data only. Component data will not be printed. = 1 Print component and summation data. Print component data even if summation was not performed.
5	IPUNCH	SUM I1	Punch-Control Flag. = 0 Do not punch any data. = 1 Punch summation data only. = 2 Punch both component and summation data. The data will be punched in the following order on a single card. ALPHA, C N, C A, C M, C Y, C LN, C LL, BETA, ICOR The card format is as follows: FORMAT (7F9.4, F7.2, 1X, I4) One identification card is punched for each set.
6-9	IRUN	SUM I4	Run number to be punched on the punched cards.
10-11	ICOMP(1)	SUM 20I2	Selected component-sequence numbers to be summed, printed, and/or punched. These numbers must correspond to the order of the data placed on the Force-Data-Save Unit (9) in the FORCE routine. For example, if you wish to sum up component numbers 5, 7, and 11, then input ICOMP(1)=5, ICOMP(2)=7, and ICOMP(3)=11.
12-13	ICOMP(2)		
14-15	ICOMP(3)		
etc.			
48-49	ICOMP(20)		

Summation-control cards (Continued)

Second card.

Column	Code	Routine Format	Explanation
1-2	ICOMP(21)	SUM 20I2	Continuation of ICOMP array of first card.
3-4	ICOMP(22)		
.	.		
.	.		
.	.		
39-40	INCOMP(40)		

If LAST = 0 then another set of Summation-Control Cards will be expected next.
If LAST = 1 then the return to AERO will be called next.

Shear Option

The Shear Option was added to the MARK IVS to calculate the shear, bending moment, and torque at various stations on the fuselage, wing, and vertical surfaces, as well as the determination of elevon hinge moments on the YF-12A configuration. Portions of the coding key on specific characteristics of the YF-12A geometry and thus, in its present form, the Shear Option is applicable only to the YF-12A.

The shear and moments may be calculated for five portions or components of the vehicle:

1. Fuselage
2. Wing
3. Nacelle vertical
4. Nacelle ventral
5. Fuselage ventral

All five components must be considered and in the order given. This was done to provide calculation efficiency since several of the components have contributions to the others (e.g., the nacelle vertical and nacelle ventral contribute to the wing shear and moments).

The shear (C_V), bending-moment (C_B), and torque (C_T) coefficients are summations of the element-force coefficients: ΔC_X, ΔC_Y, and ΔC_N - the incremental forces in the X-, Y-, and Z-directions, respectively, of the body-reference coordinate system.

Fuselage. -

The fuselage shears are calculated in the Z-direction and the bending-moments are about a line parallel to the Y-axis. A torque or moment about the X-axis is not calcualted.

$$C_V = \frac{S_{Ref}}{S_{SHR}} \sum_{NOSE}^{X_{SHR}} \Delta C_N$$

$$C_B = \frac{S_{Ref}}{S_{SHR} b_{SHR}} \sum_{NOSE}^{X_{SHR}} \Delta C_N (X_{CENT} - X_{SHR}) + \Delta C_X (Z_{CENT} - Z_{SHR})$$

Wing. -

The wing shears are defined normal to the wing mean plane which is at an incidence angle i_W with respect to the X-axis. The wing bending moments are about a line defined by the intersection of the mean plane and a plane parallel to the XZ-plane containing the point (X_{SHR}, Y_{SHR}, Z_{SHR}). The torque is the moment about a line parallel to the Y-axis containing the point (X_{SHR}, Y_{SHR}, Z_{SHR}).

$$C_V = \frac{S_{Ref}}{S_{SHR}} \sum_{WING\ TIP}^{Y_{SHR}} \Delta C_N \cos i_W + \Delta C_X \sin i_W$$

$$C_B = \frac{S_{Ref}}{S_{SHR} b_{SHR}} \sum_{WING\ TIP}^{Z_{SHR}} \Delta C_V (Y_{SHR} - Y_{CENT}) + \Delta C_Y [Z_{CENT} - (X_{CENT} - X_{SHR}) \sin i_W - Z_{SHR}] \cos i_W$$

$$C_T = \frac{S_{Ref}}{S_{SHR} \bar{c}_{SHR}} \sum_{WING\ TIP}^{Y_{SHR}} \Delta C_N (X_{CENT} - X_{SHR}) + \Delta C_X (Z_{CENT} - Z_{SHR})$$

Vertical and ventral surfaces. -

The shears are defined normal to the mean plane of the surface which is at a cant angle θ with respect to the Z-axis. The bending moments are about a line defined by the intersection of the mean plane and a plane parallel to the XY-plane containing the point $(X_{SHR}, Y_{SHR}, Z_{SHR})$. The torque is the moment about a line determined by the intersection of the mean plane and a plane parallel to the YZ-axis containing the point $(X_{SHR}, Y_{SHR}, Z_{SHR})$.

$$C_V = \frac{S_{Ref}}{S_{SHR}} \sum_{TIP}^{Z_{SHR}} \Delta C_Y \cos\theta + \Delta C_N \sin\theta$$

$$C_B = \frac{S_{Ref}}{S_{SHR} b_{SHR}} \sum_{TIP}^{Z_{SHR}} \Delta C_Y (Z_{CENT} - Z_{SHR}) + \Delta C_N (Y_{SHR} - Y_{CENT})$$

$$C_T = \frac{S_{Ref}}{S_{SHR} \bar{c}_{SHR}} \sum_{TIP}^{Z_{SHR}} \Delta C_V (X_{CENT} - X_{SHR}) + \Delta C_X [(Z_{CENT} - Z_{SHR}) \sin\theta + (Y_{CENT} - Y_{SHR}) \cos\theta]$$

The parameters appearing in the above equations are defined as follows:

S_{Ref}	is the reference area (wing area) (input on the Reference-Dimension Card)
S_{SHR}	is the area over which the shear force is integrated (shear area)
b_{SHR}	is the span of the shear area (length for fuselage)
$\bar{c}_{SHR}$	is the mean aerodynamic chord of the shear area.
$X_{SHR}, Y_{SHR}, Z_{SHR}$	are the coordinates of the shear and moment station.
$X_{CENT}, Y_{CENT}, Z_{CENT}$	are the coordinates of the centroids of the elements.

Elevon hinge moments. -

The elevon hinge-moment coefficients are calculated by using the element geometry, its position relative to the hinge line, and the pressure coefficient at the element centroid. The hinge-moment coefficient (C_H) may be expressed as the product of the pressure coefficient and a hinge-moment factor (HMFCT):

Elevon hinge moments. - (Continued)

$$C_H = \frac{1}{S_e \bar{c}_e} \sum (HMFCT)(C_p)$$

where

HMFCT	is the hinge-moment factor
C_p	is the pressure coefficient at the element centroid $(\frac{\Delta P}{q})$.
S_e	is the area of the elevon
$\bar{c}_e$	is the mean aerodynamic chord of the elevon.

The hinge-moment factor is the product of the distance from the centroid of the element to the elevon hinge line (Δx_{Elevon}) and the element area (ΔA); i.e.,

$$HMFCT = (\Delta x_{Elevon})(\Delta A)$$

The hinge-moment factors need only be calculated once and are done so on the first call to the Shear Option. The HMFCT values are stored on Unit 4 and a flag is set to so indicate.

Shear-option-input cards. -

The shears, bending moments, and torques were requested at specific locations on the YF-12A configuration and, as a convenience, the necessary input data has been predefined in the program. However, any desired location for these calculations may be obtained by overwriting this information through the use of input cards. This capability is selected by inputing the parameter IWRIT>4 on the Aero-System Title and Control Card.

A set of Shear Input Cards is required for each of the five components and must be in the following order:

Component Order, I	Component Name
1	Fuselage
2	Wing
3	Nacelle Vertical
4	Nacelle Ventral
5	Fuselage Ventral

The Elevon Deflection Cards, and the Hinge-Line Coordinate and Panel-Number Cards must also be input.

Shear component-description card (3I2):

Column	Code	Routine Format	Explanation
1-2	NSHST(I)	SHEAR I2	Number of shear stations (1 minimum, 3 maximum).
3-4	NSBTP(I,1)		Number of panels to be used in shear calculation for α-case.
5-6	NSBTP(I,2)		Number of additional panels to be used in shear calculation for β-case.

Shear station-definition cards (6F10.0):

One card is required for each new shear station requested (i.e., NSHST(I) cards).

Column	Code	Routine Format	Explanation
1-10	XSHR	SHEAR F10.0	Shear-station X-coordinate.
11-20	YSHR	SHEAR F10.0	Shear-station Y-coordinate.
21-30	ZSHR	SHEAR F10.0	Shear-station Z-coordinate.
31-40	SSHR	SHEAR F10.0	Reference area for shear coefficient.
41-50	CSHR	SHEAR F10.0	Reference length for moment coefficient.
51-60	BSHR	SHEAR F10.0	Reference length for torque coefficient.

Shear panel-number-definition cards (15I2):

Two cards are required for each shear station.

Column	Code	Routine Format	Explanation
The first card defines the panels to be used for an α-case.			
1-2	NOFP(1,I,1)	SHEAR I2	ID number of panels to be used in the shear, moment, and torque calculations for an α-case.

Shear panel-number-definition cards (Continued)

Column	Code	Routine Format	Explanation
3-4	NOFP(2,I,1)	SHEAR I2	NSBTP(I,1) values required (15 maximum).
.	.	.	
.	.	.	
.	.	.	
29-30	NOFP(15,I,1)	SHEAR I2	

The second card defines the additional panels to be used for a β-case.

1-2	NOFP(1,I,2)	SHEAR I2	ID number of additional panels to be used in the shear, moment, and torque calculations for a β-case.
3-4	NOFP(2,I,2)	SHEAR I2	NSBTP(I,2) values required (15 maximum).
.	.	.	
.	.	.	
.	.	.	
29-30	NOFP(15,I,2)	SHEAR I2	

A set of shear cards (Component-Description, Station-Definition,and Panel-Number-Definition Cards) are input for each of the five components.

Elevon-definition cards (4F10.0):

The reference area and length for each elevon are input for both sides of the vehicle. Two cards are required.

Column	Code	Routine Format	Explanation
1-10	SEH(1)	SHEAR F10.0	Reference area for hinge-moment coefficient, inboard elevon.
11-20	CEH(1)	SHEAR F10.0	Reference length for hinge-moment coefficient, inboard elevon.
21-30	SEH(2)	SHEAR F10.0	Reference area for hinge-moment coefficient, outboard elevon.
31-40	CEH(2)	SHEAR F10.0	Reference length for hinge-moment coefficient, outboard elevon.

Hinge-line-coordinate card (I2, 6F10.0, I3):

The hinge-line coordinates are the two end points of the elevon hinge line, input in the order of increasing Y (i.e., YHL2 > YHL1). An additional Hinge-Line-Coordinate and Panel-Number Card will be expected until LAST=1 is input in CC's 1-2.

Column	Code	Routine Format	Explanation
1-2	LAST	HNGFT I2	Last-Card Flag. = 0 This is not the last card. = 1 This is the last card.
3-12	XHL1	HNGFT F10.0	First X-hinge-line coordinate.
13-22	YHL1	HNGFT F10.0	First Y-hinge-line coordinate.
23-32	ZHL1	HNGFT F10.0	First Z-hinge-line coordinate.
33-43	XHL2	HNGFT F10.0	Second X-hinge-line coordinate.
44-52	YHL2	HNGFT F10.0	Second Y-hinge-line coordinate.
53-62	ZHL2	HNGFT F10.0	Second Z-hinge-line coordinate.
63-65	NIP	HNGFT I3	Number of panels for which the hinge-moment factor will be calculated.

Panel-number card (30I2):

Column	Code	Routine Format	Explanation
1-2	NØFP(1)	HNGFT I2	Panel numbers for which the hinge-moment factor will be calculated.
3-4	NØFP(2)	.	
5-6	NØFP(3)	.	
.	.	.	
.	.	.	
.	.	.	

Panel-number card (Continued)

Column	Code	Routine Format	Explanation
59-60	NØFP(30)	HNGFT I2	

Note that the Hinge-Line-Coordinate Card and the Panel-Number Card are read in pairs.

Automated Input Assembler for AERO Option

The analysis of the YF-12A wind-tunnel data presented a special situation in which a given configuration was to be considered for a very large number of test conditions (in excess of 1000). If the input data were handled in the normal manner, one would quickly be inundated with card decks.

The geometry, region definitions, and pressure tap locations remain the same from run-to-run. The only changes for each run are certain control parameters and the values of the measured pressure coefficients. To simplify the input procedure, a subroutine (INPUT) was added to the MARK IVS to assemble the input data from three auxiliary files:

Unit 17 - Aero Input Data, Alpha Case
Unit 18 - Aero Input Data, Beta Case
Unit 19 - Measured Pressure-Coefficient Data (corrected)

Unit 18 differs from Unit 17 in that both sides of the vehicle must be described for a Beta case and thus twice as many regions are defined. Otherwise the two units are very similar, consisting of card images of the required Aerodynamic Program Input Cards beginning with the Aero Flag Card. The pressure coefficient is obtained from Unit 19 by tap number and by position in the C_p-data array. A slightly modified format of the Hand-Loaded Surface-Pressure-Coefficient-Data Card is used to convey this information:

Column	Code	Routine Format	Explanation
1-10	X	Input F10.0	X-coordinate of the surface point
11-20	Y	Input F10.0	Y-coordinate of the surface point
21-30	Z	Input F10.0	Z-coordinate of the surface point
31-35	ITAP	Input I5	Pressure tap number (a negative value is used to indicate a pressure tap on the opposite side of the vehicle)
36-40	LOC	Input I5	Position in the C_p-data array

The format of Unit 19 is as follows:

Record Number	Word Number	Description	
1	1	ICOR,	Correlation ID number
	2	CONF,	Configuration
	3	A(1),	Angle of attack
	4	A(2),	Sideslip angle
	5	A(3),	Mach number
	6	A(4),	Reynolds number
	7	A(5),	Total pressure
	8	A(6),	Dynamic pressure
	9	A(7),	L.H. plug position
	10	A(8),	R.H. plug position
	11	A(9),	By-pass mass-flow ratio
	12	A(10),	Center-body-bleed mass-flow ratio
	13	A(11),	Calibrated mass-flow ratio at the rake
	14	B(1),	Tunnel used
	15	B(2),	Test number
	16	B(3),	Run number
	17-631	CP(I),	I=1,615. Pressure coefficient data array
2		Same format as record Number 1	
3			
.			
.			
.			
etc.			

SECTION 4. ILLUSTRATED EXAMPLE

An example case is presented to supplement the card-by-card input instructions in the hope that it will provide the user with a better understanding of the capabilities of the MARK IVS Program. To demonstrate the use of the MARK IVS program it is first necessary to have a configuration for which a pressure distribution is available. This would normally be obtained from experimental results, a process which in itself can involve considerable effort. To simplify this process, a case for which the pressure distribution has previously been calculated using the MARK IV Program was selected for the example. This alleviates the burden of sorting through experimental data and also serves as a basis to assess the accuracy of the MARK IVS spline-fitting technique.

One of the simple wing-body configurations (aft mounted wing, t/c = 0.10) of Reference 5 is used for the present example. This configuration was used for Sample Problem #3 of the MARK IV Program (see p. 212, Reference 2) and thus the pressure distributions and total vehicle forces are already available. As set up for the original MARK IV calculation, the geometry was assumed to have been input and stored on a previous run and viscous as well as inviscid calculations were performed. In the present example, only the inviscid pressures are used and the problem is set up in its entirity: geometry, input pressures, force calculations, and summations. The flight condition for the example case is Mach 4.5 at 6° angle of attack, which is the third condition of the original MARK IV problem.

The sample problem has been set up to run in the MARK IVS in two stages: (1) geometry definition and load to Unit 4, and (2) pressure input, force calculation, and component summations. This is the recommended procedure in that it allows verification of the geometry before any attempt is made to obtain the forces, the latter being the more expensive task of the two. It is also strongly recommended that the geometry check be done with some graphical or pictorial means. The illustrated example, while a relatively simple configuration, has been constructed using a number of different options to demonstrate the generality and flexibility of the MARK IVS Program. Of particular importance in this regard, is the complete independence between the geometry panels and the input pressure regions.

Geometry Definition

The configuration, designated 101A, consists of a body with circular cross-sections and an aft-located wing having a double-wedge airfoil section with swept leading and trailing edges. Six panels have been used to define the configuration as shown schematically in Figure 15. The attributes of these panels are summarized in Table 1 and the complete card input required to generate the configuration is presented in Appendix A. Dimensional data for configuration 101A are given in Reference 5. In both the MARK IV sample problem and the present example, the geometry has been scaled to give the body a length of 100. A three-view of the computer generated configuration is given in Figure 16. The body has approximately half the number of axial rows (and thus half as many elements) used in the original MARK IV Program to

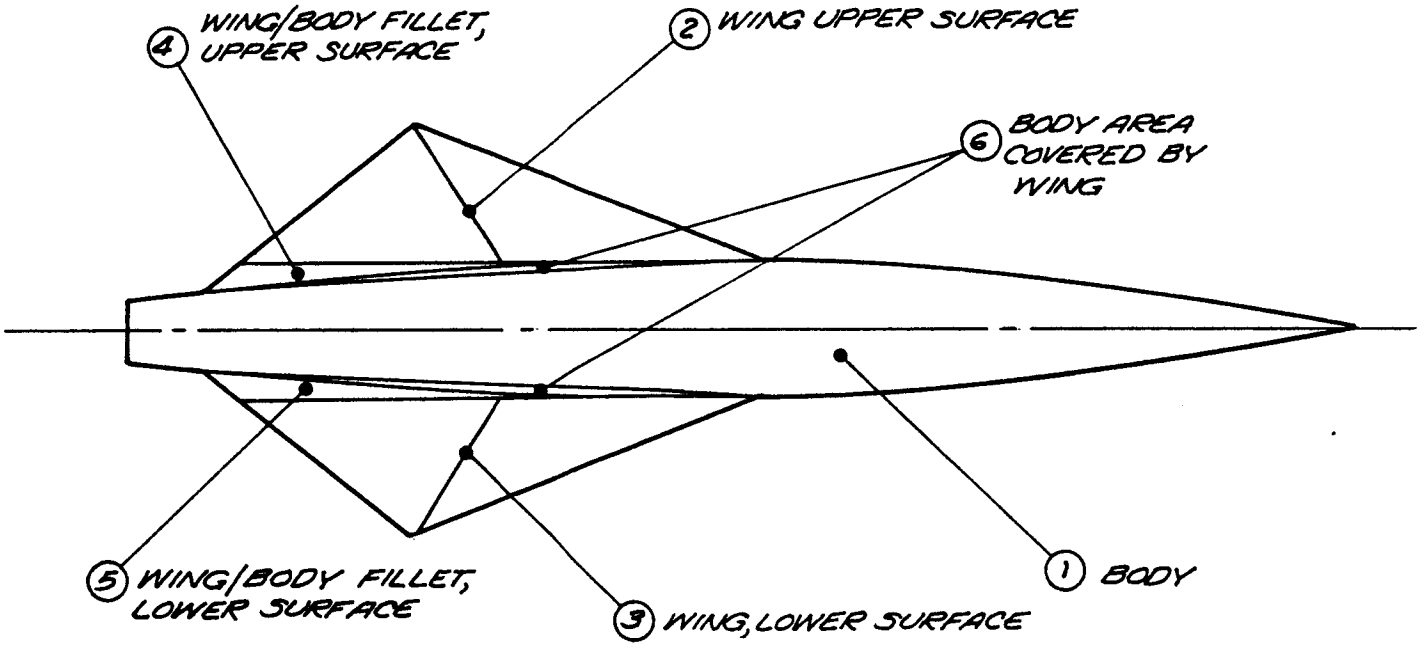


FIGURE 15. CONFIGURATION 101A GEOMETRY PANELS

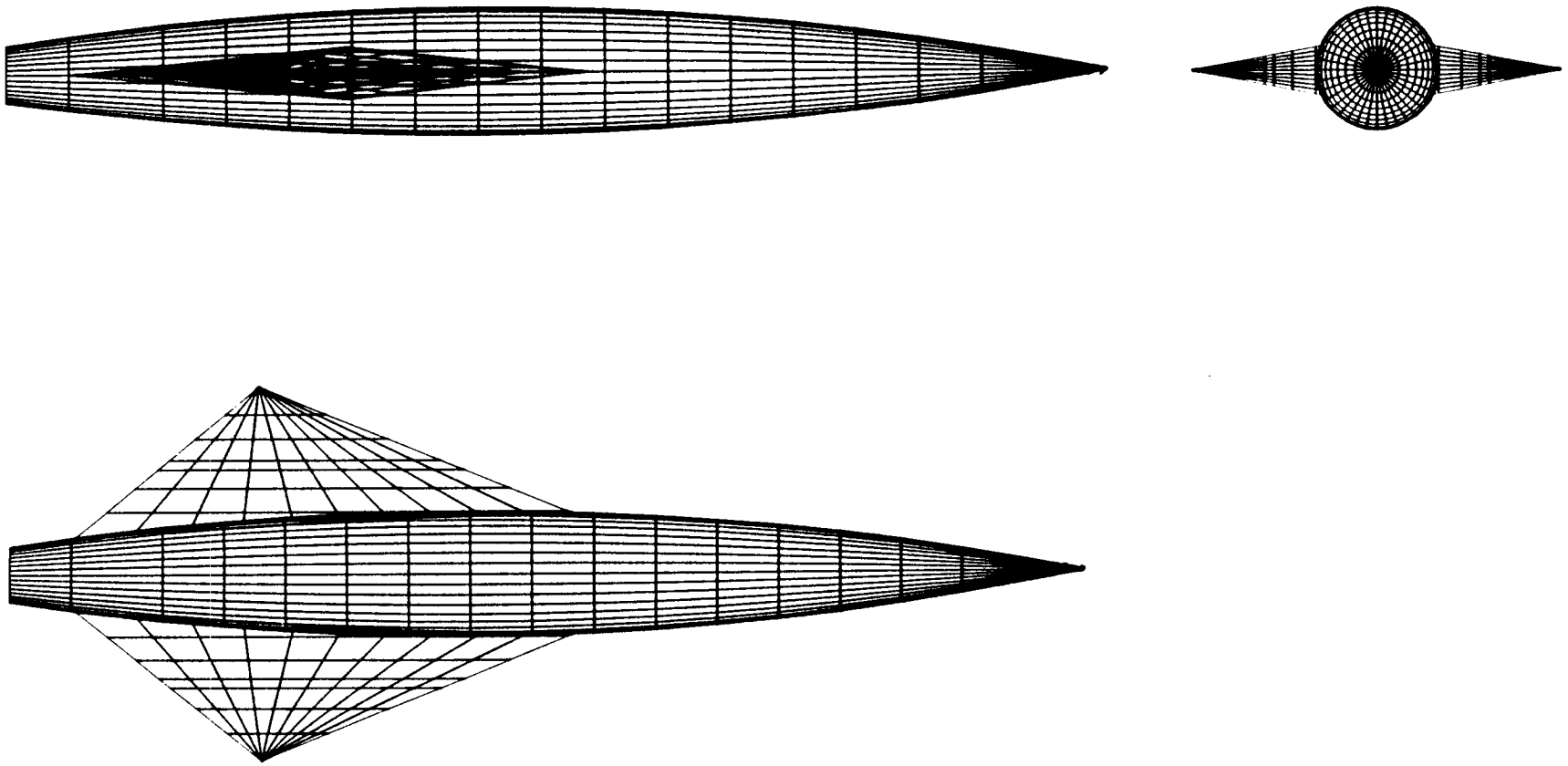


FIGURE 16. CONFIGURATION 101A THREE VIEW

calculate the pressures. The wing, however, is set up identically to the original MARK IV version. The printed output (shortened for convenience) for the geometry generation are also included in Appendix A.

Pressure Input, Force Calculation, and Summations

Once the configuration geometry has been loaded onto Unit 4 and verified, the calculation portion of the problem, or stage 2 can proceed. This could have also been done in multiple passes since the results of each task are saved on mass storage units for future access. That is, the pressure input and storage on Unit 10 could be accomplished separately. Once the input pressures were verified the force calculation run could be made and the results saved on Unit 9 for subsequent summations. In the present example, these tasks are all completed during a single machine pass. A complete listing of the required input cards along with selected output pages for the Stage 2 operation are presented in Appendix B. Various annotations are included on these pages as additional aids to the user. It is intended that Appendix B will be used in conjunction with the following discussion and therefore, no further direct reference to it will be made.

The problem is set up with three separate calls to the AERO Option. The first loads the input pressure data to Unit 10, images this data to allow verification, and calculates the forces on all of the panels except the sixth. The sixth panel, the area on the body covered by the wing, should have an opposite contribution in the summation process. This is accomplished in the second call to AERO by entering a negative value for the reference area. Finally, the forces and moments for various combinations are summed during the third call to the AERO Option.

Six flow regions have been defined for inputting pressures as shown in Figure 17. The choice of flow regions, while arbitrary, should be selected with regard to the variation in surface pressures anticipated. The present example is for supersonic flight conditions (Mach = 4.5) and the flow regions have been selected accordingly. Region 1 is the body forward of any wing effects. Region 2 is the aft body and will include any wing interference. The wing having a double-wedge or diamond airfoil shape naturally divides into fore and aft regions on both the upper and lower surfaces.

The results from the original MARK IV included pressures calculated at 842 locations on the configuration. Since both the configuration and flight conditions are symmetric (zero yaw), this refers to 842 locations on the half-vehicle. Similarly, in the present example, pressures are input only on one-half the configuration, and the complete vehicle forces are handled via a symmetry flag. The choice of input locations was completely arbitrary but has been done in a uniform manner. Of the available 39 axially located rows on the fuselage, 12 (six in Region 1 and six in Region 2) have been used with approximately every other point in a row for a total of 10 points-per-row. On the wing all the points in every other row starting with the second have been used. The detailed identification of the input pressures is provided in Table 2. The correspondence between the original MARK IV problem and the present example is summarized in Table 3.

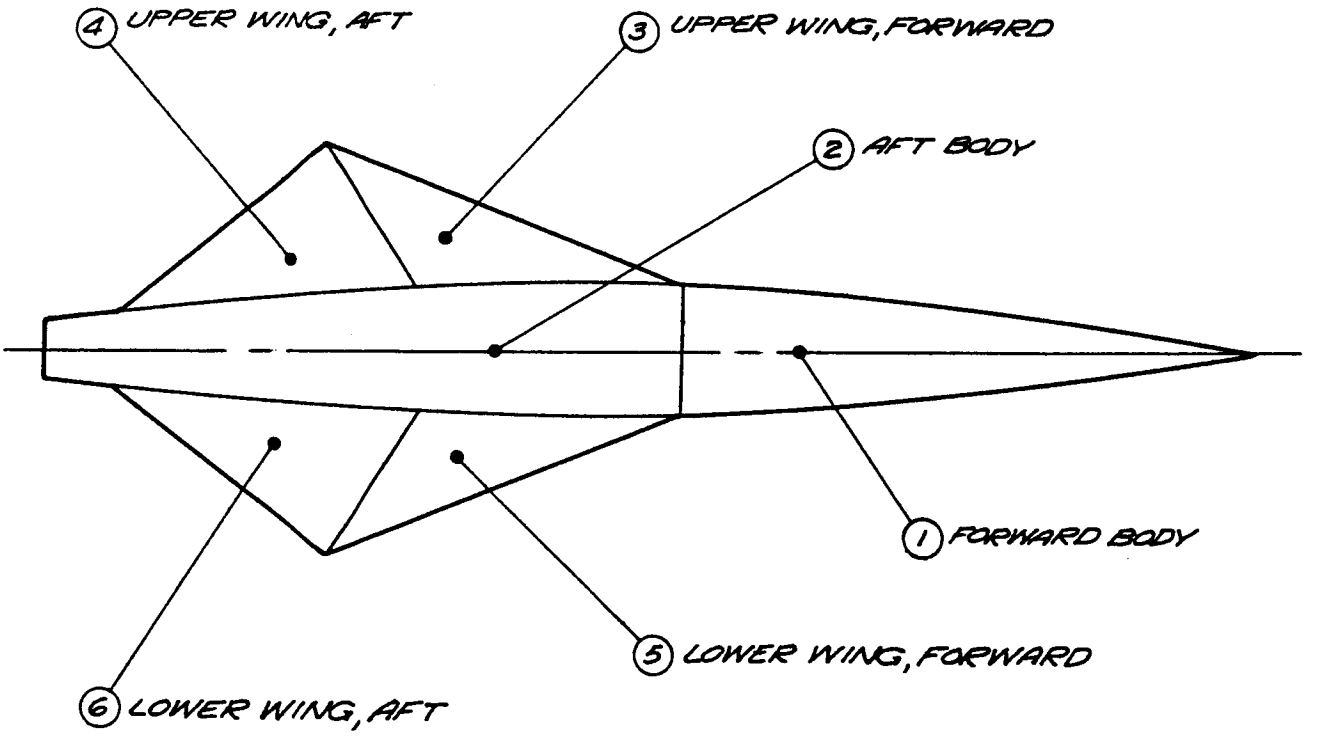


FIGURE 17. CONFIGURATION 101A FLOW REGIONS

After the pressure data has been loaded to the flow field data Unit 10, it was imaged to allow opportunity to verify that the data was loaded correctly. The forces and moments are calculated by component which may be composed of one or more panels. The geometry panels, in turn, may be subject to one or more flow regions. Seven components have been defined for the present example as shown in Table 4. Thus Panel 1, the fuselage, is influenced by two flow regions and has been accounted for as two components. The extent of a flow region is controlled by the normalization parameters. In a similar fashion flow Regions 3 and 4 influence two geometry panels (the wing upper surface and the wing/body fillet upper surface) which make up Component number 3. The remaining components are similarly composed.

The third call to AERO is to obtain summations of the force components in various combinations. This is commonly used to arrive at a component build-up of the forces and moments. In the present example, seven summation data sets are used to demonstrate combinations of both force components and previously defined summation data sets. These data sets are listed in Table 5 along with a comparison of the lift/drag ratios determined by the original MARK IV calculations and the present MARK IVS Program. The lift/drag ratio is very sensitive to small changes in either the lift or drag and thus serves as a good indicator of accuracy.

Summation Data Set 1 combines Components 1 and 2, the forward and aft fuselage portions, and thus represents the body alone. Summation Data Set 2 is just the body area covered by the wing, Component 7. Summation Data Set 3 is the combination of Summations 1 and 2, and is therefore the body alone less the area covered by the wing. The wing surfaces are collected in Summations 4 and 5 with the complete wing alone given by Summation 6. Finally, the complete configuration is presented in Summation 7 by combining Summations 3 and 6. The difference in lift/drag ratios between the two programs is of the order of 0.1% for both the body alone and the complete wing alone. The body area covered by the wing, Summation 2, shows an inordinately large difference in L/D of about 4%. This is particularly puzzling in view of the fact that this is a simple four-element panel. However, it is just this fact that causes the difference. The original MARK IV pressures were calculated using an impact or local-slope method on both the body and the four-element area directly. The present method obtains the pressure at the centroids of the four elements by interpolation of the body pressures. Since the body is modeled with a much more dense mesh of elements, it is not surprising that these pressures will differ. It is concluded that the Surface-Spline Pressure-Interpolation Technique used in the MARK IVS Program is more than adequate.

TABLE 1. CONFIGURATION 101A GEOMETRY PANEL CHARACTERISTICS

PANEL NO.	PANEL ID	NUMBER OF ELEMENTS	DESCRIPTION
1	B	342	BODY ELLIPSE GENERATION 19 AXIAL ROWS, 18 ELEMENTS/ROW
2	WNGU	67	WING UPPER SURFACE AIRCRAFT GEOMETRY OPTION 6 SPANWISE ROWS, 11 ELEMENTS/ROW (ONE DUMMY EACH ROW) PLUS DUMMY STATUS 3 ELEMENT
3	WNGL	67	WING LOWER SURFACE AIRCRAFT GEOMETRY OPTION 6 SPANWISE ROWS, 11 ELEMENTS/ROW (ONE DUMMY EACH ROW) PLUS DUMMY STATUS 3 ELEMENT
4	WBU	11	WING/BODY FILLET, UPPER SURFACE ELEMENT INPUT 1 ROW, 11 ELEMENTS/ROW (ONE DUMMY ELEMENT)
5	WBL	11	WING/BODY FILLET, LOWER SURFACE ELEMENT INPUT 1 ROW, 11 ELEMENTS/ROW (ONE DUMMY ELEMENT)
6	C	4	AREA ON BODY COVERED BY WING ELEMENT INPUT 2 ROWS, 2 ELEMENTS/ROW

TABLE 2. INPUT PRESSURES

a) FORWARD BODY - Region 1

Row No	ORIGINAL MARK IV ELEMENT NUMBERS
1	1, 3, 5, 7, 9, 10, 12, 14, 16, 18
6	91, 93, 95, 97, 99, 100, 102, 104, 106, 108
8	127, 129, 131, 133, 135, 136, 138, 140, 142, 144
12	199, 201, 203, 205, 207, 208, 210, 212, 214, 216
16	271, 273, 275, 277, 279, 280, 282, 284, 286, 288
21	361, 363, 365, 367, 369, 370, 372, 374, 376, 378

b) AFT BODY - Region 2

Row No.	ORIGINAL MARK IV ELEMENT NUMBERS
22	379, 381, 383, 385, 387, 388, 390, 392, 394, 396
26	451, 453, 455, 457, 459, 460, 462, 464, 466, 468
30	523, 525, 527, 529, 531, 532, 534, 536, 538, 540
33	577, 579, 581, 583, 585, 586, 588, 590, 592, 594
36	631, 633, 635, 637, 639, 640, 642, 644, 646, 648
39	685, 687, 689, 691, 693, 694, 696, 698, 700, 702

c) UPPER WING FORWARD - Region 3

Row No.	ORIGINAL MARK IV ELEMENT NUMBERS
2	12, 13, 14, 15, 16
4	34, 35, 36, 37, 38
6	56, 57, 58, 59, 60

d) UPPER WING AFT - Region 4

Row No.	ORIGINAL MARK IV ELEMENT NUMBERS
2	17, 18, 19, 20, 21
4	39, 40, 41, 42, 43
6	61, 62, 63, 64, 65

TABLE 2. INPUT PRESSURES (CONTD)

e) LOWER WING FORWARD - Region 5

Row No.	ORIGINAL MARK IV ELEMENT NUMBERS
2	133, 134, 135, 136, 137
4	111, 112, 113, 114, 115
6	89, 90, 91, 92, 93

f) LOWER WING AFT - Region 6

Row No.	ORIGINAL MARK IV ELEMENT NUMBERS
2	138, 139, 140, 141, 142
4	116, 117, 118, 119, 120
6	94, 95, 96, 97, 98

TABLE 3. CORRESPONDENCE BETWEEN ORIGINAL MARK IV CALCULATIONS AND MARK IVS INPUT PRESSURES

FLOW REGION		ORIGINAL MARK IV	MARK IVS	
NO.	DESCRIPTION	GEOMETRY ELEMENTS & PRESSURES	GEOMETRY ELEMENTS	PRESSURES INPUT
1	FORWARD BODY	378	198	60
2	AFT BODY	324	144	60
3	UPPER WING FORWARD	35	35	15
4	UPPER WING AFT	35	35	15
5	LOWER WING FORWARD	35	35	15
6	LOWER WING AFT	35	35	15
	TOTALS	842	482	180

TABLE 4. CONFIGURATION 101A FORCE COMPONENTS

FORCE COMPONENT	GEOMETRY PANELS	FLOW REGION	DESCRIPTION
1	1	1	FORWARD FUSELAGE
2	1	2	AFT FUSELAGE
3	2,4	3	UPPER WING, FORWARD
4	2,4	4	UPPER WING, AFT
5	3,5	5	LOWER WING, FORWARD
6	3,5	6	LOWER WING, AFT
7*	6	2	FUSELAGE AREA COVERED BY WING

*SEPARATE ENTRY INTO AERO TO ALLOW NEGATIVE SREF.

TABLE 5. CONFIGURATION 101A SUMMATION DATA SETS

SUMMATION DATA SET	TYPE/NUMBER	DESCRIPTION	LIFT/DRAG	
			MARK IV CALCULATED	MARK IVS INPUT PRESSURES
1	COMPONENTS 1,2	BODY ALONE	4.07619	4.07234
2	COMPONENT 7	BODY AREA COVERED BY WING	0.88936	0.85354
3	SUMMATIONS 1,2	COMPLETE BODY	4.16074	4.14767
4	COMPONENTS 3,4	WING UPPER SURFACE	- -	6.00269
5	COMPONENTS 5,6	WING LOWER SURFACE	- -	3.91434
6	SUMMATIONS 4,5	COMPLETE WING ALONE	4.72091	4.72583
7	SUMMATIONS 3,6	COMPLETE CONFIGURATION	4.41016	4.40507

SECTION 5. DATA-STORAGE TECHNIQUES

Each major part of the MARK IVS Program makes use of data that has been generated and stored by some other program component. The manner in which these data are stored and retrieved is important both to the user and to the programmer seeking to add some new capability to the program. The principal concern for the user is that he know what is stored, and what sequence number, pointer, etc., might be required as subsequent input data to retrieve it. The primary storage units are tabulated below.

Unit Number	Name	Usage
1	TAPEIN	Copy of all input data.
2	TAPEB	Scratch unit.
3	TAPEA	Scratch unit.
4	-	Geometry-data storage (mass storage).
5	TAPEIN	System input-data stream.
6	TAPEOT	System output-print unit.
7	TAPEG	System punch-output unit.
8	-	Element-data geometry-data storage (Type 3 cards).
9	-	Force-data storage (mass storage).
10	-	Flow-field-data storage (mass storage).
11	TAPEC	Scratch unit.
12	TAPEE	Geometry data for interference usage.
13	TAPEF	Geometry data plus local-flow-field data.
14	TAPED	Scratch.
15	-	Coordinates and pointers for C_p data (mass storage).
16	-	C_p data (mass storage).
17	TAPEH	YF-12A aero auto input (alpha case).
18	TAPEI	YF-12A aero auto input (beta case).
19	TAPEJ	YF-12A input-C_p data for auto input.

Unit Number	Name	Usage
20	-	Case data - integrated forces, moments, and shears (mass storage).
21	-	Surface-spline pressure-distribution data (mass storage).

Units 4, 9, 10, 15, 16, 20, and 21 listed above are identified as being "mass-storage" units. That is, these units are used to store data along with the necessary tags or pointers so that a given item can be retrieved directly without using a sequential read. On the CDC machine this process is called "mass storage". On the IBM machine it is identified as "direct-access" storage. The CDC "mass storage" required an index variable to be dimensionalized equal to the number of records on the unit. IBM does not have this requirement and thus suffers no penalty in terms of core required to accommodate large units (i.e., Units 4, 16, and 21). Reference to those units has been recoded using the sub-index capability for the CDC System. Although the two machines use different types of FORTRAN statements to write and read the data, the general procedure is still the same. A pointer or tag must be set or calculated by the program for each FORTRAN write statement. This pointer or tag is the sequential number of the record on the data set. The use of the same tag number in a FORTRAN read will result in the proper retrieval of the record. The specific data stored on the "mass-storage" Units 4, 9, 10, 15, 16, 20, and 21 are described below.

Unit 4. Geometry Data

Unit 4 is the storage unit for geometry data generated in routine GEOM. The unit normally contains both the input-element corner points plus all the characteristics of the plane quadrilateral. When it is requested to do so, the program will also store on Unit 4 the surface-property data calculated in the FORCE routine and the hinge-moment factor calculated in the SHEAR routine. Unit 4 has a master table of contents followed by the data for each element. The unit is set-up and initialized in routine GEOM.

Record Number	Word Number	Description
1	1	Configuration title (CONFIG from the Geometry-Control Card).
2	1	NEXT, pointer to next empty record.
	2	NP, number of PANELS currently stored.
	3	NPMAX, maximum number of panels provided for in the table of contents.
	4	NREM, number of record spaces provided for use on each element (set = 3 in GEOM).
	5	NEHMF, elevon hinge-moment factor storage indicator.

Unit 4. (Continued)

Record Number	Word Number	Description
3	1	Pointer to next record after geometry-data storage is complete.
4		Not used.
5		Panel #1 table array.
	1	Panel number (ISTAT3 = 1).
	2	Panel name (PANEL from the Panel-Identification Card).
	3	Record number of first element (ISTART).
	4	Number of elements in the panel (L).
	5	Panel orientation (IORN).
	6	Symmetry flag (SYMFCT).
	7	Number of columns of elements (N).
	8	Number of rows in each column (I).
	9	Panel number adjacent to side 1 of panel (NADJ1).
	10	Panel number adjacent to side 2 of panel (NADJ2).
	11	Panel number adjacent to side 3 of panel (NADJ3).
	12	Panel number adjacent to side 4 of panel (NADJ4).
6		Blank
7		Blank
8		Blank
9		Blank
10		Panel #2 table array, (same format as record-number 5).
11		Blank
12		Blank
13		Blank
14		Blank
15		Panel #3 table array, (same format as record-number 5).
etc.		
(NP*5)		Panel #NP (last panel loaded).
(NP*5)+1		Blank
(NP*5)+2		Blank
(NP*5)+3		Blank
(NP*5)+4		Blank

Unit 4. (Continued)

Record Number	Word Number	Description
(NPMAX*5)+1		Last panel for which space is provided
(NPMAX*5)+2		(NPMAX may be = NP if desired).
(NPMAX*5)+3		
(NPMAX*5)+4		
[(NPMAX+1)*5]		Element #1. First available record for element data. The array ELEM is written here in GEOM.
[(NPMAX+1)*5+1]		Pointer array (25 words) to where surface output data is written by FORCE for input side of vehicle. First word is for 1st α-β, 2nd for 2nd α-β, etc.
[(NPMAX+1)*5+2]		Same as above for reflected side of vehicle.

The data written on Unit 4 for each element are contained in the element-data array, ELEM, in subroutine GEOM. As indicated above, the 25 word array ELEM for the first element is written on Unit 4 at record number [(NPMAX+1)*5]. In routine GEOM the next two records, [(NPMAX+1)*5+1] and [(NPMAX+1)*5+2], are filled with 0.0 values. These records will be filled with live pointer data in subroutine FORCE as indicated above. These pointers will give the location of the calculated force data. We thus have three records per element on Unit 4.

The element data array, ELEM, contains the following information:

ELEM(1)	= L	Element number.
ELEM(2)	= N	Column number.
ELEM(3)	= I	Row number.
ELEM(4)	= NX	X-component of unit normal.
ELEM(5)	= NY	Y-component of unit normal.
ELEM(6)	= NZ	Z-component of unit normal.
ELEM(7)	= XCENT	X-coordinate of element centroid.
ELEM(8)	= YCENT	Y-coordinate of element centroid.
ELEM(9)	= ZCENT	Z-coordinate of element centroid.
ELEM(10)	= AREA	Element area.
ELEM(11)	= XIN(1)	X-coordinates of input element corner
ELEM(12)	= XIN(2)	points.
ELEM(13)	= XIN(3)	
ELEM(14)	= XIN(4)	
ELEM(15)	= YIN(1)	Y-coordinates of input element corner
ELEM(16)	= YIN(2)	points.
ELEM(17)	= YIN(3)	
ELEM(18)	= YIN(4)	
ELEM(19)	= ZIN(1)	Z-coordinates of input element corner
ELEM(20)	= ZIN(2)	points.
ELEM(21)	= ZIN(3)	
ELEM(22)	= ZIN(4)	
ELEM(23)	= HMF	Elevon hinge-moment factor.
ELEM(24)	= 0.0	
ELEM(25)	= 0.0	

Unit 4. (Continued)

When directed by the input parameter ISAVE subroutine FORCE will write certain results from the pressure and force calculations out on Unit 4. These surface-property data are in the form of single records containing the array El in FORCE. Since up to 20 α-β conditions can be analyzed on a pass into FORCE, pointers are required on Unit 4 to indicate where each El record is written. These pointers are written in the vacant record locations provided right behind the original element-data record, ELEM, for each element. The surface-property data contained in the array El in FORCE are as follows.

El(1)	= IG4	Pointer to original element data on Unit 4.
El(2)	= E(1)	Element number.
El(3)	= 0.0	Shear.
El(4)	= 0.0	
El(5)	= CP	Element-pressure coefficient
El(6)	= DELTA	Element impact angle
El(7)	= MACH	
El(8)	= SX	X-component of surface unit-velocity vector.
El(9)	= XY	Y-component of surface unit-velocity vector.
El(10)	= SZ	Z-component of surface unit-velocity vector.
El(11)	= BS(2)/PFS	Local pressure divided by freestream pressure
El(12)	= BS(3)/TFS	Local temperature divided by freestream temperature.
El(13)	= DELCA	ΔC_A
El(14)	= DELCY	ΔC_Y
El(15)	= DELCN	ΔC_N
El(16)	= DELCLL	ΔC_ℓ
El(17)	= DELCLM	ΔC_m
El(18)	= DELCLN	ΔC_n
El(19)	= 0.0	
El(20)	= 0.0	
El(21)	= 0.0	
El(22)	= 0.0	
El(23)	= 0.0	
El(24)	= 0.0	
El(25)	= 0.0	

The total number of records that can be stored on Unit 4 is controlled by the pointer-array size provided for in the MAIN program. In the MARK IVS release of the program this was set at 20,000 records. A check is made in routine FORCE to prevent the storage of more than this number of records. Thus, the total number of elements that can be analyzed at one time depends upon whether or not surface-property data are saved.

The total number or record spaces required on Unit 4 is given by the following relationship.

$$L_{total} = 4 + NPMAX*5 + L_{max}*3 + L_{save}$$

Unit 4. (Continued)

where

NPMAX = Maximum number of panels provided for on the unit (input on the Geometry-Control Card).

L_{max} = The total number of elements for all panels.

L_{save} = The total number of elements for which local surface-property data will be saved in routine FORCE. Must include the total element count for both sides of the vehicle if it is yawed and the symmetry reflection process is used.

Unit 9. Force-Data

Unit 9 is used to store component-force data generated in the invicid routine PRES. Unit 9 contains a table of contents followed by the data for each vehicle component.

Record Number	Word Number	Description
1	1	NTOT, total number of components.
	2	ISUM, total number of summations.
	3	NEXT, next empty record space.
	4	TITLE9(1) = 10HFORCE DATA for CDC.
		= 4HFORC for IBM.
	5	TITLE9(2) = 10HSAVE UNIT for CDC.
		= 4HE DA for IBM.
	6	TITLE9(3) = blank for CDC.
		= 4HTA S for IBM.
	7	TITLE9(4) = blank for CDC.
		= 4HAVE for IBM.
	8	TITLE9(5) = blank for CDC.
		= 4HUNIT for IBM.
2	1	Identification flag for component data.
	2-41	LCOM, pointers to location of component data (80 total, 41 through 80 stored at record-number 5).
3	1	Identification flag for summation data (=0.0 for force data only).
	2-41	LSUM, pointers to location of summation data (40 total).

Unit 9. (Continued)

Record Number	Word Number	Description
4	1	Mach, freestream Mach number.
	2	V, freestream velocity.
	3	RENO, freestream Reynolds number per foot.
	4	ALT, altitude.
	5	PSTAG, stagnation pressure in atmospheres.
	6	TSTAG, stagnation temperature.
	7	GTYPE(1), gas type name.
	8	GTYPE(2), gas type name.
	9	SREF, reference wing area.
	10	SPAN, reference span length.
	11	MAC, reference mean aerodynamic chord.
	12	XCG, X-location of cg.
	13	YCG, Y-location of cg.
	14	ZCG, Z-location of cg.
5	1-40	LCOM(I), I = 41, 80
6-9		Not used.
10		Force-data table of contents for 1st component (record number is = LCOM(1) = 10).
	1	Component number = 1.
	2-11	IPANL array, panel numbers used to make up this component.
11	1	NAB, number of α-β sets.
	2	Value of α for 1st α-β set.
	3	Value of α for 2nd α-β set.
	etc.	
	22	Value of β for 1st α-β set.
	23	Value of β for 2nd α-β set.
	etc.	
	41	
12		Force data for 1st component at 1st α-β. The array F in routine PRES.
	1	F(1) = α
	2	F(2) = C_D
	3	F(3) = C_L
	4	F(4) = C_A
	5	F(5) = C_Y
	6	F(6) = C_N
	7	F(7) = β
	8	F(8) = C_L/C_D

Unit 9. (Continued)

Record Number	Word Number	Description
	9	F(9) = C_m
	10	F(10) = C_ℓ
	11	F(11) = C_n
13	1-11	Force derivatives for 1st component at 1st α-β. Not used.
14	1-11	Force data for 1st component at 2nd α-β. Array F.
15	1-11	Force derivatives for 1st component at 2nd α-β. Not used.
16	1-11	Force data for 1st component at 3rd α-β. Array F.
17	1-11	Force derivatives for 1st component at 3rd α-β. Not used.

etc.

The above storage process starting with record 10 is repeated for each vehicle component. The starting record number for each component is placed in the LCOM array that is stored in Record-Number 2. The pointers to summation-data sets are placed in LSUM and stored in Record-Number 3. Normally, Record-Number 4 containing the Mach number and other data is based on the last component placed on the unit.

Unit 10. Flow-Field-Data

Unit 10 is the most complex of the mass-storage (direct-access) units used by the program. Although Unit 10 is called the FLOW FIELD Unit (and is used for many purposes in the MARK IV program), it is used in the MARK IVS program to store the input surface-pressure-coefficient data. The unit contains a number of directories and tables that hold not only basic-identification information, but pointers that provide storage-location-record numbers pointing to lower-level directories and the various types of data.

The general format of Unit 10 is shown in Figure 18. Starting with the Master Directoryat the top we see that it contains a set of 5 pointers in the array IMTAB. These point to what are called major-data sets. For example, each set could be for a completely different vehicle or for a different Mach number for the same vehicle. Each of these sets has a Set Directory containing the Mach number and a pointer array, LOAB, pointing to a different lower level directory for each α-β combination. In turn, each α-β Directory has pointers, LORG, that point to the Region Directories. The Region Directories contain some basic-

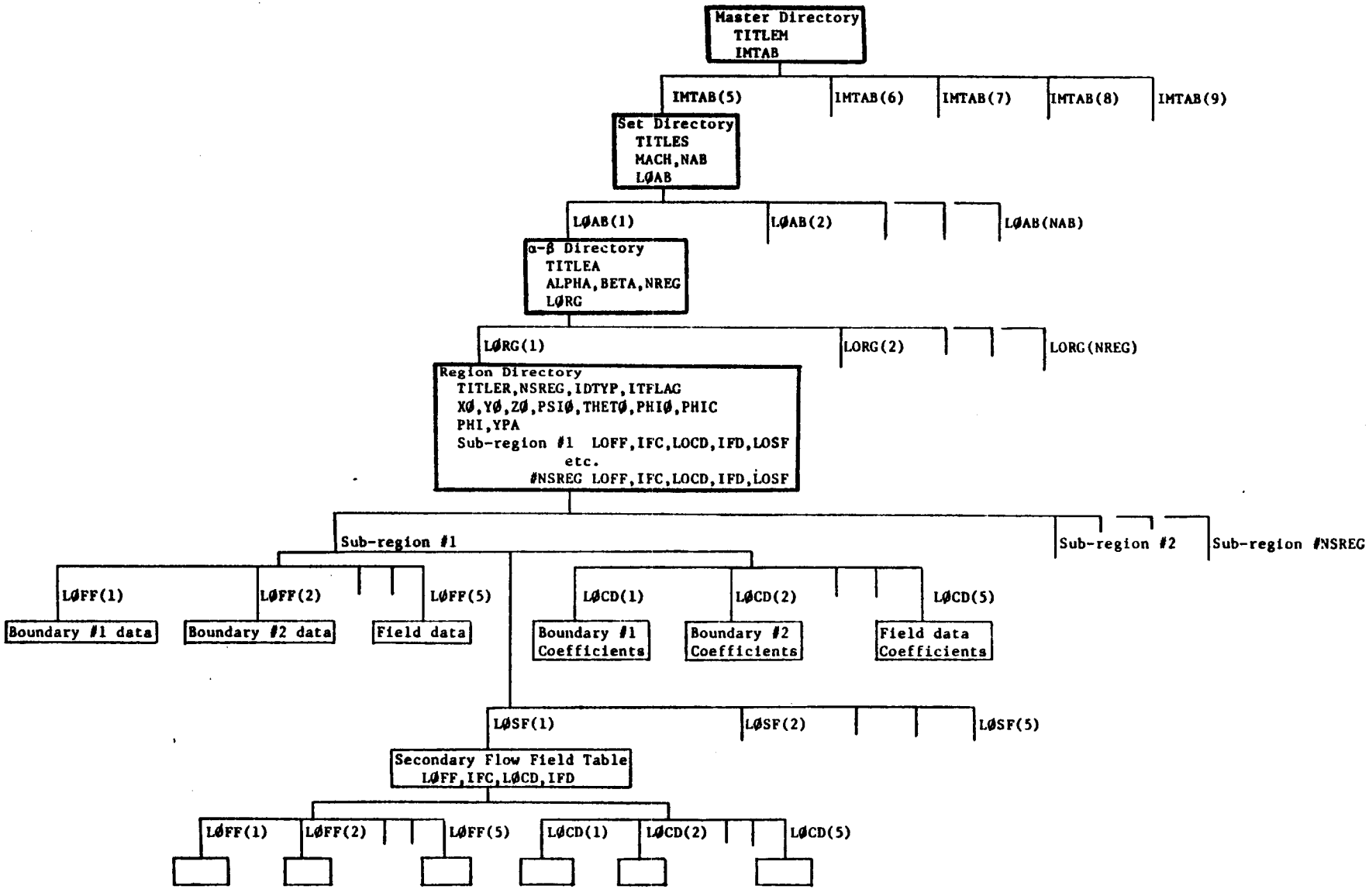


FIGURE 18. FLOW-FIELD DATA-STORAGE FORMAT

flag and interpolation parameters plus a set of pointers used to reach and retrieve data in the still lower-level sub-regions and secondary-flow-field tables.

The basic format for each of the directory tables is listed below.

Record Number	Description
	Master Directory
1	TITLEM = Unit title (10 words).
2	IMTAB(1) = Number of separate data sets stored on the unit (NDSET). IMTAB(2) = Pointer to next empty record (LCNEXT). IMTAB(3) = Empty IMTAB(4) = Empty IMTAB(5) = Pointer to start of 1st set = LOSET(1). IMTAB(6) = Pointer to start of 2nd set = LOSET(2). IMTAB(7) = Pointer to start of 3rd set = LOSET(3). IMTAB(8) = Pointer to start of 4th set = LOSET(4). IMTAB(9) = Pointer to start of 5th set = LOSET(5).
3	Empty
	Set Directory (First set is at record-number 4).
LOSET	TITLES, MACH, NAB (number of α-β sets).
LOSET+1	LOAB array pointers to α-β sets.
LOSET+2	Empty
	α-β Directory
LOAB	TITLEA, ALPHA, BETA, NREG (numbers of regions).
LOAB+1	LORG array pointers to Region Directories.
LOAB+2	Empty
	Region Directory
LORG	TITLER NSREG = Number of sub-regions. IDTYP(1) = Data-type flag. = 1 flow-field data. = 2 surface-property data. = 3 streamline data. = 4 shock-shape data (not used at present time).

Unit 10. (Continued)

Record Number	Description
	Region Directory (Continued)
	IDTYP(2) = Uniform - non-uniform flag. = 1 uniform flow-field data. = 2 non-uniform data. ITFLAG Not used.
LORG+1	XO, YO, ZO, PSIO, etc.
LORG+2	PHI or YPA (not active).
LORG+3	PHI or YPA (not active).
LORG+4	Empty
LORG+5	Pointers for sub-region #1. LOFF, IFC, LOCD, IFD, LOSF.
LORG+6	Pointers for sub-region #2. LOFF, IFC, LOCD, IFD, LOSF.
etc.	etc.
LORG+(4+NSRG)	Pointers for last sub-region (NSREG).

Unit 15. Coordinates and Pointers for C_p Data

Unit 16. C_p Data

Units 15 and 16 are used together to store coordinate data and interpolated C_p data for subsequent access by a plotting program. The coordinate data is arranged as contours or rows on the vehicle surface and must have been loaded and stored on Unit 15 prior to a MARK IVS run. The interpolated C_p data are stored on Unit 16 while the directories or pointers to this data are stored on Unit 15. Units 15 and 16 are used if the parameter IPLTCP≠0 on the Aero-System Title and Control Card.

Unit 15. Coordinates and pointers for C_p data. -

Record Number	Word Number	Description
1	1	Title, Unit 15 title.
	2	NROW, number of rows stored.
	3	NEXTCØ, pointer to next empty row - coordinate record.
	4	NRUN, number of runs stored.
	5	NEXTRN, pointer to next empty run - parameter record.
	6	NEXTRW, pointer to next empty row C_p-data-pointer record.
	7	NEXTCP, pointer to next empty row C_p-data record (Unit 16).
2		Blank.
3-4	1-200	IDRW, row ID-number directory.
5-6	1-200	ITYP, row type-flag directory.
7-8	1-200	INRW, number of points per row directory.
9-10	1-200	IRUL, upper/lower surface-flag directory.
11-12	1-200	IPCØ, pointers to row-coordinate data.
13-20		Blank.
21	1-100	IDRN, run ID-number directory.
22	1-100	NCRN, number of correlation-run directory.
23	1-100	IRNP, pointers to correlation-directory table.
24	1-100	FMACH, run Mach-number directory.
25	1-100	AMPB, by-pass mass-flow-ratio directory.
26-30		Blank.
31		First-run ID-directory table.
	1-4	IDCN, correlation ID-number directory.
	5-8	IRWP, pointers to the row C_p-data-pointer directory.
	9-12	ALPHA, angle-of-attack directory.
	13-16	BETA, sideslip-angle directory.
32		Second-run ID-directory table (same format as record-number 31).
.		.
.		.
.		.

Unit 15. (Continued)

Record Number	Word Number	Description
130		Hundredth-run ID-directory table (same format as record-number 31). Data Pointers For First Correlation Stored.
131-132	1-200	IRCP, pointers to the C_p data.
133-134	1-200	IRCT, row count of C_p-data-saved directory. Data Pointers For Second Correlation Stored.
135-136	1-200	(Same format as first correlation stored).
137-138	1-200	(Same format as first correlation stored).
.		.
.		.
.		.
.		Data Pointers For The Hundreth Correlation Stored.
1327-1328	1-200	(Same format as first correlation stored).
1329-1330	1-200	(Same format as first correlation stored).
1331-2000		Row-coordinate data. I=1, INRW X(I), X-row coordinate. Y(I), Y-row coordinate. Z(I), Z-row coordinate. IØ(I), point-type input/output IP(I), panel number. Note: IBM - Five words per data point and twenty data points per record. If the number of points is greater than twenty, then the program will span the next record. Each row starts on a new record. CDC - Five words per data point and one row per record. The length of each record is: Length = INRW* words.

Unit 16. C_p data. -

Record Number	Word Number	Description
1-92720		Interpolated C_p data.
		Note: IBM - Twenty data points per record. If the number of points is greater than twenty, then the program will span the next record. Each row starts on a new record.
		CDC - One row per record. The length of each record is: Length = INRW words.

Unit 20. Case Data-Integrated Forces, Moments, and Shears

Unit 20 is used to store the vehicle forces and moments, the shears, bending moments, torques, and elevon hinge moments for subsequent access by a plotting program. Unit 20 is used if the parameter ISVFMS≠0 on the Aero-System Title and Control Card and the data are saved in the Shear Option. The vehicle forces and moments are retrieved from the Force Data Storage (Unit 9) and thus must have been previously saved (ISAVE=0 ,1 on the Pressure Title Card).

Record Number	Word Number	Description
1	1	NEXRUN, pointer to the next empty space in the IRUNID array.
	2	NEXDAT, pointer to the next empty data record.
2		Blank.
3	1-216	IRUNID, run ID-number directory (for IBM - IRUNID will span the next record - 108 words per record).
5	1-216	NCORR, number of correlations directory (for IBM - NCORR will span the next record - 108 words per record).
7	1-108	ICORID, first correlation ID-number directory.
.	.	
.	.	
28	1-108	ICORID, 22nd case-data-pointer directory.
29	1-108	IDAPT, first case-data-pointer directory.
.	.	
.	.	
.	.	

Unit 20. (Continued)

Record Number	Word Number	Description
50	1-108	IDAPT, 22nd case-data-pointer directory.
51	1	IRSQ, run sequence number.
	2	IAB, alpha-beta flag.
	3	EMACH, run Mach number.
	4-14	Force data F-array.
	15-108	Shears, bending moments, torques, and elevon hinge moments (see subroutine SHEAR for details).
52	1-108	Same format as record-number 51.
.	.	
.	.	
1150	1-108	Same format as record-number 51.

Unit 21. Surface-Spline Pressure-Distribution Data

Unit 21 is used to store the surface-spline definitions and coefficients for each of the pressure regions. This includes all the normalization parameters, boundary definitions, normalized coordinates used to define the fit, and the spline coefficients necessary to recreate the pressure distribution. Unit 21 is used if ISVSPL≠0 on the Aero-System Title and Control Card. This is a special option which provides the capability to determine the pressure at any arbitrary point on the vehicle, subsequent to the MARK IVS run, without having to redetermine the spline fit. Since all of the test conditions analyzed for the YF-12A were saved, the unit is very large.

The format of Unit 21 is very complex, consisting of set locations for directories and a "free form" for the data storage. For efficiency, the region definitions are saved only once and as they occur during execution of the MARK IVS. Consequently, the first portion of the unit has, in addition to the set directories, coordinate and spline-coefficient-data storage intermixed. Eventually, after all the regions have been considered, the information stored will be wholly spline-coefficient data.

Record Number	Word Number	Description
Master Directory:		
1	1	NEXPAN, pointer to next empty record available for storing Region definitions.
	2	NUMPAN, number of Region definitions stored.

Record Number	Word Number	Description
	3	NEXSPL, pointer to next empty record for storing spline-coefficient data.
	4	NUMCOR number of correlations stored.
Region Directory:		
2	1-72	IPAN, Region ID number.
4	1-72	IPDP, pointers to Region definitions.
		IPAN(I) gives the Region ID number and the corresponding IPDP(I) gives the record number where the Region definition is stored. Provision for 72 regions (for IBM-36 words per record, IPAN and IPDP will span the next record).
Region Definitions:		Records 6 through 149 are provided for storing the Region definitions. IBM requires two records per Region (144 total for 72 regions). A new Region always begins with an even record number.
6	1-48	ICØP, IC, INØRM, ISURF, IX, IY, IZ, U1, UL, V1, VL, CR, CT, CHX, N2, N3, XØ, YØ, ZØ, AP11, AP12, AP13, AP21, AP22, AP31, AP33, XB(1), XB(2), XB(3), XB(4), YB(1), YB(2), YB(3), YB(4), ZB(1), ZB(2), ZB(3), ZB(4), BX(1,1), BY(1,1), BX(2,1), BY(2,1), BX(1,2), BY(1,2), BX(2,2), BY(2,2).
8	1-48	Same as record-number 6.
.	.	
.	.	
148	1-48	Same as record-number 6.
		ICØP is the pointer to the Region-normalized-coordinate data and IC is the number of coordinates. The remaining parameters may be determined by reference to subroutine SFNTR2.
Correlation Directory:		Corelation number was used in the YF-12A study to designate a particular test condition. This directory uses three variables to identify the correlation number, ICID(J), to give the record number of the Spline-Pointer Directory, ICPP(J). Provision provided for 1260 correlation numbers (IBM requires 35 records for each variable).

Unit 21. (Continued)

Record Number	Word Number	Description
Correlation Directory: (Continued)		
150	1-1260	ICID
185	1-1260	ICPP
220	1-1260	IBTA

The remaining records of Unit 21 are in a "free-form" format. The first available record is number 255 at which location the Spline-Pointer Directory, IPSP1, for the first correlation number will be located. If this is a beta case (IBTA=1), a second Spline-Pointer Directory, IPSP2, will be stored in the next location (record-number 256). Record-number 257 will contain the normalized-coordinate data for the first Region. This will be followed by the spline-coefficient data for the first correlation number associated with the first Region. Data storage at subsequent locations is entirely dependent on the MARK IVS run. Normally, normalized coordinates for the second Region will be the next data stored.

255	1-36	IPSP1, spline-pointer directory.
256	1-36	IPSP2, second spline-pointer directory for beta case (IBTA=1). The location of the IPSP1 array is given by ICPP(1).
257		Normalized coordinates for first Region: IC pairs of XI, YI coordinates. This location is given by ICØP(1). The number of records required is: CDC, NCREC = 1 IBM, NCREC = (2*IC-1)/36 + 1
257+NCREC		Spline-coefficient data for first correlation and first Region: NC, N3, IPNSEQ, FLOWC(5), (IG(L), L=1, NC), (B(M,5), M=1, N3). This location is given by IPSP1(1). The number of records required is: CDC, NSREC = 1 IBM, NSREC = (NC+N3+12-1)/36 +1

Unit 21. (Continued)

Record Number	Word Number	Description
257+NCREC +NSREC		Normalized coordinates for second Region. This location is given by ICØP(2).
etc.		

APPENDIX A

ILLUSTRATED EXAMPLE GEOMETRY DEFINITION

This section provides a listing of the card input required to set up the geometry for Configuration 101A used in the illustrated example (Section 4). The listing is output by the program (INMONT = 0) and only the first card in each job, the Executive-Flag Card, is not included. It should be added to the deck in any attempt to recreate the run from these listings.

Each page of the input listing contains a header which includes two lines indicating the card columns. Identification of several of the card types has been added on the right-hand margin. The input listing is given on the first three pages. The remaining pages present selected portions of the printed output from the geometry definition stage. Various annotations are given on the right-hand margin of these pages as additional aids to the user.

```
SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0
CASE      0                                                                                              PAGE    1

SUPERSONIC-HYPERSONIC ARBITRARY-BODY PROGRAM INPUT DATA

0         1         2         3         4         5         6         7         9
12345678901234567890123456789012345678901234567890123456789012345678901234567890

1                                                                                          SYSTEM CONTROL CARD
 BOI060   CASE IOIA, WING AFT                                   101                        GEOMETRY CONTROL CARD
BODY302       01                                                                           PANEL ID CARD
CASE IOIA  BODY                                                 11         B 4       1
0.0        0.0    180.0   180.0       0.0        0.0     0.0    0          B 5       2
-1.016     0.0    180.0   180.203     .203       0.0     0.0    0          B 5       4
-2.032     0.0    180.0   180.401     .401       0.0     0.0    0          B 5       6
-5.080     0.0    180.0   180.972     .972       0.0     0.0    0          B 5       8
-10.160    0.0    180.0   181.841     1.841      0.0     0.0    0          B 5      10
-15.240    0.0    180.0   182.603     2.603      0.0     0.0    0          B 5      12
-20.320    0.0    180.0   183.263     3.263      0.0     0.0    0          B 5      14
-25.400    0.0    180.0   183.820     3.820      0.0     0.0    0          B 5      16   ELLIPSE GENERATION
-30.480    0.0    180.0   184.274     4.274      0.0     0.0    0          B 5      18        CARDS
-35.560    0.0    180.0   184.627     4.627      0.0     0.0    0          B 5      20
-40.640    0.0    180.0   184.879     4.879      0.0     0.0    0          B 5      22
-45.720    0.0    180.0   185.029     5.029      0.0     0.0    0          B 5      24
-50.800    0.0    180.0   185.080     5.080      0.0     0.0    0          B 5      26
-55.880    0.0    180.0   185.029     5.029      0.0     0.0    0          B 5      28
-60.960    0.0    180.0   184.879     4.879      0.0     0.0    0          B 5      30
-66.040    0.0    180.0   184.627     4.627      0.0     0.0    0          B 5      32
-71.120    0.0    180.0   184.274     4.274      0.0     0.0    0          B 5      34
-76.200    0.0    180.0   183.820     3.820      0.0     0.0    0          B 5      36
-83.820    0.0    180.0   182.946     2.946      0.0     0.0    0          B 5      39
-88.900    0.0    180.0   182.235     2.235      0.0     0.0    4          B 5      41
WNGU014       01                                                                           PANEL ID CARD
CASE IOIA, WING UPPER SURFACE  T/C = 0.10                       0  0
 0  0  0  0  0  0  01 7 11
WING UPPER SURFACE     +1     7    11    0     0     0    0     0     1     0  WNGU
0.0      10.0     20.0    30.0    40.0    50.0    60.0    70.0    80.0    90.0              7
100.0                                                                                       8
-42.7    5.08      0.0    -80.30   5.08    0.0     0.0                                       AIRCRAFT GEOMETRY
-47.5    7.00      0.0    -78.40   7.00    0.0     0.0                                         OPTION CARDS
-51.25   8.50      0.0    -76.525  8.50    0.0     0.0
-53.1749.26984     0.0    -75.5629.2684    0.0     0.0
-57.50   11.0      0.0    -73.40   11.0    0.0     0.0
-62.50   13.0      0.0    -70.90   13.0    0.0     0.0
-68.10   15.24     0.0    -68.101  15.24   0.0     0.0
0.0      1.0      2.0     3.0     4.0     5.0     4.0     3.0     2.0     1.0               10
0.0
                                                                          99
WNGL014       01                                                                           PANEL ID CARD
```

SUPERSONIC-HYPERSONIC ARBITRARY-BODY PROGRAM INPUT DATA

```
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CASE 101A, WING LOWER SURFACE  T/C = 0.10                    0  0
  0  0  0  0  0  0  01 7 11
WING LOWER SURFACE      -1    7   11    0    0    0    0    0    1    0  WNGL
0.0     10.0    20.0    30.0    40.0    50.0    60.0    70.0    80.0    90.0                7
100.0                                                                                       8
-42.7  5.08     0.0    -80.80  5.08    0.0     0.0
-47.5  7.00     0.0    -78.40  7.00    0.0     0.0
-51.25 8.50     0.0    -76.525 8.50    0.0     0.0
-53.1749.26984  0.0    -75.5429.2684   0.0     0.0
-57.50 11.0     0.0    -73.40  11.0    0.0     0.0
-62.50 13.0     0.0    -70.00  13.0    0.0     0.0
-68.10 15.24    0.0    -68.101 15.24   0.0     0.0
0.0     1.0     2.0     3.0     4.0     5.0     4.0     3.0     2.0     1.0                10
0.0
                                                                                   99
 WBU011          01
 1.000
-42.3425    4.936643   0.0       2-46.38      5.02       0.385     0       WBU 3        1
-50.2       5.00       0.765     0-53.43      4.91       1.11      0       WBU 3        2
-57.58      4.73       1.577     0-61.31      4.40       2.03      0       WBU 3        3
-65.38      4.35       1.63      0-69.49      4.19       1.24      0       WBU 3        4
-73.74      3.96       0.848     0-78.36      3.55       0.438     0       WBU 3        5
-83.402359  2.99812    0.0       0-83.402360  2.99812    0.0       0       WBU 3        6
  -42.7000     5.0800     0.0     1  -46.5100     5.0800     0.38100      WBU 3        7
  -50.3200     5.0800     0.76200   -54.1300     5.0800     1.14300      WBU 3        8
  -57.9400     5.0800     1.52400   -61.7500     5.0800     1.90500      WBU 3        9
  -65.5600     5.0800     1.52400   -69.3700     5.0800     1.14300      WBU 3       10
  -73.1800     5.0800     0.76200   -76.9900     5.0800     0.38100      WBU 3       11
  -80.8000     5.0800     0.0     0 -80.8000     5.0800     0.0     3    WBU 3       12
 WBL011          01
 1.000
  -42.7000     5.0800     0.0     2  -46.5100     5.0800    -0.38100      WBL 3        1
  -50.3200     5.0800    -0.76200   -54.1300     5.0800    -1.14300      WBL 3        2
  -57.9400     5.0800    -1.52400   -61.7500     5.0800    -1.90500      WBL 3        3
  -65.5600     5.0800    -1.52400   -69.3700     5.0800    -1.14300      WBL 3        4
  -73.1800     5.0800    -0.76200   -76.9900     5.0800    -0.38100      WBL 3        5
  -80.8000     5.0800     0.0     0  -80.8000     5.0800     0.0     0    WBL 3        6
-42.3425    4.936643   0.0       1-46.38      5.02      -0.385     0       WBL 3        7
-50.2       5.00      -0.765     0-53.43      4.91      -1.11      0       WBL 3        8
-57.58      4.73      -1.577     0-61.31      4.40      -2.03      0       WBL 3        9
-65.38      4.35      -1.63      0-69.49      4.19      -1.24      0       WBL 3       10
```

AIRCRAFT GEOMETRY OPTION CARDS

PANEL ID CARD

ELEMENT INPUT CARDS

PANEL ID CARD

ELEMENT INPUT CARDS

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0 PAGE 3

SUPERSONIC-HYPERSONIC ARBITRARY-BODY PROGRAM INPUT DATA

```
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

-73.74      3.96      -0.948    0-78.36     3.55      -0.438    0      WBL 3      11
-83.402359  2.99812   0.0       0-83.402360 2.99812   0.0       3       BL 3      12
   C111          01
 1 000
-42.3425    4.936643  0.0       2-42.3425   4.936643  0.0       0        C 3       1
-42.3425    4.936643  0.0       0-61.31     4.40      -2.03     1        C 3       2
-61.31      4.85      0.0       0-61.31     4.40       2.03     0        C 3       3
-83.40236   2.99812   0.0       1-83.40236  2.99812   0.0       0        C 3       4
-83.40236   2.99812   0.0       0-83.40236  2.99812   0.0       3        C 3       5
1.124859391.124859391.12485939    0.0       0.0       0.0
1.124859391.124859391.12485939    0.0       0.0       0.0
1.124859391.124859391.12485939    0.0       0.0       0.0
1.124859391.124859391.12485939    0.0       0.0       0.0
1.124859391.124859391.12485939    0.0       0.0       0.0
1.124859391.124859391.12485939    0.0       0.0       0.0
```

PANEL ID CARD

ELEMENT INPUT CARDS

SCALE FACTOR CARDS

```
***SUPERSONIC-HYPERSONIC ARBITRARY-BODY AERODYNAMIC COMPUTER PROGRAM SYSTEM *************

          PROGRAM OPTIONS ARE IN THE FOLLOWING ORDER....

                1  GEOMETRY PROGRAM  (OPTION 1)
```

ELLIPTICAL GEOMETRY DATA IS BEING GENERATED **********

ANALYTICALLY GENERATED ELEMENT DATA

CASE 0 CASE 101A BODY PAGE 4

X	Y	Z	S	X	Y	Z	S	CASE	SECT	SEQ
0.0	0.0	0.0	2	0.0	0.0	0.0	0	0	B 3AERO	1
0.0	0.0	0.0	0	0.0	0.0	0.0	0	0	B 3AERO	2
0.0	0.0	0.0	0	0.0	0.0	0.0	0	0	B 3AERO	3
0.0	0.0	0.0	0	0.0	0.0	0.0	0	0	B 3AERO	4
0.0	0.0	0.0	0	0.0	0.0	0.0	0	0	B 3AERO	5
0.0	0.0	0.0	0	0.0	0.0	0.0	0	0	B 3AERO	6
0.0	0.0	0.0	0	0.0	0.0	0.0	0	0	B 3AERO	7
0.0	0.0	0.0	0	0.0	0.0	0.0	0	0	B 3AERO	8
0.0	0.0	0.0	0	0.0	0.0	0.0	0	0	B 3AERO	9
0.0	0.0	0.0	0	-1.0160	0.0	-0.20301	0		B 3AERO	10
-1.0160	0.0353	-0.19990		-1.0160	0.0694	-0.19080	0		B 3AERO	11
-1.0160	0.1015	-0.17580		-1.0160	0.1305	-0.15550	0		B 3AERO	12
-1.0160	0.1555	-0.13050		-1.0160	0.1758	-0.10150	0		B 3AERO	13
-1.0160	0.1908	-0.06940		-1.0160	0.1999	-0.03530	0		B 3AERO	14
-1.0160	0.2030	-0.00000		-1.0160	0.1999	0.03530	0		B 3AERO	15
-1.0160	0.1908	0.06940		-1.0160	0.1758	0.10150	0		B 3AERO	16
-1.0160	0.1555	0.13050		-1.0160	0.1305	0.15550	0		B 3AERO	17
-1.0160	0.1015	0.17580		-1.0160	0.0694	0.19080	0		B 3AERO	18
-1.0160	0.0353	0.19990		-1.0160	0.0000	0.20300	0		B 3AERO	19
-2.0320	0.0	-0.40101		-2.0320	0.0696	-0.39490	0		B 3AERO	20
-2.0320	0.1371	-0.37680		-2.0320	0.2005	-0.34730	0		B 3AERO	21
-2.0320	0.2578	-0.30720		-2.0320	0.3072	-0.25780	0		B 3AERO	22
-2.0320	0.3473	-0.20050		-2.0320	0.3768	-0.13720	0		B 3AERO	23

FIRST PAGE OF BODY ELEMENT DATA CREATED BY ELLIPSE GENERATION

ANALYTICALLY GENERATED ELEMENT DATA

CASE 0 CASE 101A BODY PAGE 12

X	Y	Z	S	X	Y	Z	S	CASE	SECT	SEQ
-88.9000	2.1002	-0.76440		-88.9000	2.2010	-0.38810		0	B 3AERO	185
-88.9000	2.2350	-0.00000		-88.9000	2.2010	0.38810		0	B 3AERO	186
-88.9000	2.1002	0.76440		-88.9000	1.9356	1.11750		0	B 3AERO	187
-88.9000	1.7121	1.43660		-88.9000	1.4366	1.71210		0	B 3AERO	188
-88.9000	1.1175	1.93560		-88.9000	0.7644	2.10020		0	B 3AERO	189
-88.9000	0.3891	2.20100		-88.9000	0.0000	2.23503		0	B 3AERO	190

LAST PAGE OF BODY ELEMENT DATA

*** OUTPUT GEOMETRY DATA FROM AIRCRAFT GEOMETRY OPTION ****

CONFIGURATION CASE 101A, WING UPPER SURFACE T/C = 0.10 CASE 0

X	Y	Z	S	X	Y	Z	S		
-42.7000	5.0800	0.0	2	-46.5100	5.0800	0.38100	0	3WNGU	1
-50.3200	5.0800	0.76200		-54.1300	5.0800	1.14300	0	3WNGU	2
-57.9400	5.0800	1.52400		-61.7500	5.0800	1.90500	0	3WNGU	3
-65.5600	5.0800	1.52400		-69.3700	5.0800	1.14300	0	3WNGU	4
-73.1800	5.0800	0.76200		-76.9900	5.0800	0.38100	0	3WNGU	5
-80.8000	5.0800	0.0	0	-80.8000	5.0800	0.0	0	3WNGU	6
-47.5000	7.0000	0.0	1	-50.5900	7.0000	0.30900	0	3WNGU	7
-53.6800	7.0000	0.61800		-56.7700	7.0000	0.92700	0	3WNGU	8
-59.8600	7.0000	1.23600		-62.9500	7.0000	1.54500	0	3WNGU	9
-66.0400	7.0000	1.23600		-69.1300	7.0000	0.92700	0	3WNGU	10
-72.2200	7.0000	0.61800		-75.3100	7.0000	0.30900	0	3WNGU	11
-78.4000	7.0000	0.0	0	-78.4000	7.0000	0.0	0	3WNGU	12
-51.2500	8.5000	0.0	1	-53.7775	8.5000	0.25270	0	3WNGU	13
-56.3050	8.5000	0.50550		-58.8325	8.5000	0.75820	0	3WNGU	14
-61.3600	8.5000	1.01100		-63.8875	8.5000	1.26370	0	3WNGU	15
-66.4150	8.5000	1.01100		-68.9425	8.5000	0.75820	0	3WNGU	16
-71.4700	8.5000	0.50550		-73.9975	8.5000	0.25270	0	3WNGU	17
-76.5250	8.5000	0.0	0	-76.5250	8.5000	0.0	0	3WNGU	18
-53.1740	9.2698	0.0	1	-55.4128	9.2697	0.22390	0	3WNGU	19
-57.6516	9.2696	0.44780		-59.8904	9.2694	0.67160	0	3WNGU	20
-62.1292	9.2693	0.89550		-64.3680	9.2691	1.11940	0	3WNGU	21
-66.6068	9.2690	0.89550		-68.8456	9.2688	0.67160	0	3WNGU	22
-71.0844	9.2687	0.44780		-73.3232	9.2685	0.22390	0	3WNGU	23
-75.5620	9.2684	0.0	0	-75.5620	9.2684	0.0	0	3WNGU	24
-57.5000	11.0000	0.0	1	-59.0900	11.0000	0.15900	0	3WNGU	25
-60.6800	11.0000	0.31800		-62.2700	11.0000	0.47700	0	3WNGU	26
-63.8600	11.0000	0.63600		-65.4500	11.0000	0.79500	0	3WNGU	27
-67.0400	11.0000	0.63600		-68.6300	11.0000	0.47700	0	3WNGU	28
-70.2200	11.0000	0.31800		-71.8100	11.0000	0.15900	0	3WNGU	29
-73.4000	11.0000	0.0	0	-73.4000	11.0000	0.0	0	3WNGU	30
-62.5000	13.0000	0.0	1	-63.3400	13.0000	0.08400	0	3WNGU	31
-64.1800	13.0000	0.16800		-65.0200	13.0000	0.25200	0	3WNGU	32
-65.8600	13.0000	0.33600		-66.7000	13.0000	0.42000	0	3WNGU	33
-67.5400	13.0000	0.33600		-68.3800	13.0000	0.25200	0	3WNGU	34
-69.2200	13.0000	0.16800		-70.0600	13.0000	0.08400	0	3WNGU	35
-70.9000	13.0000	0.0	0	-70.9000	13.0000	0.0	0	3WNGU	36
-68.1000	15.2400	0.0	1	-68.1001	15.2400	0.00000	0	3WNGU	37
-68.1002	15.2400	0.00000		-68.1003	15.2400	0.00000	0	3WNGU	38
-68.1004	15.2400	0.00000		-68.1005	15.2400	0.00000	0	3WNGU	39
-68.1006	15.2400	0.00000		-68.1007	15.2400	0.00000	0	3WNGU	40
-68.1008	15.2400	0.00000		-68.1009	15.2400	0.00000	0	3WNGU	41
-68.1010	15.2400	0.0	0	-68.1010	15.2400	0.0	0	3WNGU	42
0.0	0.0	0.0	2	0.0	0.0	0.00010		3	
-0.0001	0.0	0.0	1	-0.0001	0.0	0.00013		3	

ELEMENT DATA FOR WING UPPER SURFACE

*** OUTPUT GEOMETRY DATA FROM AIRCRAFT GEOMETRY OPTION ****

CONFIGURATION CASE 101A, WING LOWER SURFACE T/C = 0.10 CASE 0

ELEMENT DATA FOR WING LOWER SURFACE

X	Y	Z	S	X	Y	Z	S			
-68.1000	15.2400	0.0	2	-68.1001	15.2400	-0.00000	0	3WNGL	1
-68.1002	15.2400	-0.00000		-68.1003	15.2400	-0.00000	0	3WNGL	2
-68.1004	15.2400	-0.00000		-68.1005	15.2400	-0.00000	0	3WNGL	3
-68.1006	15.2400	-0.00000		-68.1007	15.2400	-0.00000	0	3WNGL	4
-68.1008	15.2400	-0.00000		-68.1009	15.2400	-0.00000	0	3WNGL	5
-68.1010	15.2400	0.0	0	-68.1010	15.2400	0.0	0	3WNGL	6
-62.5000	13.0000	0.0	1	-63.3400	13.0000	-0.08400	0	3WNGL	7
-64.1800	13.0000	-0.16800		-65.0200	13.0000	-0.25200	0	3WNGL	8
-65.8500	13.0000	-0.33600		-66.7000	13.0000	-0.42000	0	3WNGL	9
-67.5400	13.0000	-0.33600		-68.3800	13.0000	-0.25200	0	3WNGL	10
-68.2200	13.0000	-0.16800		-70.0600	13.0000	-0.08400	0	3WNGL	11
-70.9000	13.0000	0.0	0	-70.9000	13.0000	0.0	0	3WNGL	12
-57.5000	11.0000	0.0	1	-59.0900	11.0000	-0.15900	0	3WNGL	13
-60.6800	11.0000	-0.31800		-62.2700	11.0000	-0.47700	0	3WNGL	14
-63.8600	11.0000	-0.63600		-65.4500	11.0000	-0.79500	0	3WNGL	15
-67.0400	11.0000	-0.63600		-68.6300	11.0000	-0.47700	0	3WNGL	16
-70.2200	11.0000	-0.31800		-71.8100	11.0000	-0.15900	0	3WNGL	17
-73.4000	11.0000	0.0	0	-73.4000	11.0000	0.0	0	3WNGL	18
-53.1740	9.2698	0.0	1	-55.4128	9.2697	-0.22300	0	3WNGL	19
-57.6516	9.2696	-0.44780		-59.8904	9.2694	-0.67160	0	3WNGL	20
-62.1292	9.2693	-0.89550		-64.3680	9.2691	-1.11940	0	3WNGL	21
-66.6068	9.2690	-0.89550		-68.8456	9.2688	-0.67160	0	3WNGL	22
-71.0844	9.2687	-0.44780		-73.3232	9.2685	-0.22300	0	3WNGL	23
-75.5620	9.2684	0.0	0	-75.5620	9.2684	0.0	0	3WNGL	24
-51.2500	8.5000	0.0	1	-53.7775	8.5000	-0.25270	0	3WNGL	25
-56.3050	8.5000	-0.50550		-58.8325	8.5000	-0.75820	0	3WNGL	26
-61.3600	8.5000	-1.01100		-63.8875	8.5000	-1.26370	0	3WNGL	27
-66.4150	8.5000	-1.01100		-68.9425	8.5000	-0.75820	0	3WNGL	28
-71.4700	8.5000	-0.50550		-73.9975	8.5000	-0.25270	0	3WNGL	29
-76.5250	8.5000	0.0	0	-76.5250	8.5000	0.0	0	3WNGL	30
-47.5000	7.0000	0.0	1	-50.5900	7.0000	-0.30900	0	3WNGL	31
-53.6800	7.0000	-0.61800		-56.7700	7.0000	-0.92700	0	3WNGL	32
-59.8600	7.0000	-1.23600		-62.9500	7.0000	-1.54500	0	3WNGL	33
-66.0400	7.0000	-1.23600		-69.1300	7.0000	-0.92700	0	3WNGL	34
-72.2200	7.0000	-0.61800		-75.3100	7.0000	-0.30900	0	3WNGL	35
-78.4000	7.0000	0.0	0	-78.4000	7.0000	0.0	0	3WNGL	36
-42.7000	5.0800	0.0	1	-46.5100	5.0800	-0.38100	0	3WNGL	37
-50.3200	5.0800	-0.76200		-54.1300	5.0800	-1.14300	0	3WNGL	38
-57.9400	5.0800	-1.52400		-61.7500	5.0800	-1.90500	0	3WNGL	39
-65.5600	5.0800	-1.52400		-69.3700	5.0800	-1.14300	0	3WNGL	40
-73.1800	5.0800	-0.76200		-76.9900	5.0800	-0.38100	0	3WNGL	41
-80.8000	5.0800	0.0	0	-80.8000	5.0800	0.0	0	3WNGL	42
0.0	0.0	0.0	2	0.0	0.0	0.00010		3	
-0.0001	0.0	0.0	1	-0.0001	0.0	0.00013		3	

TYPE 3 CARDS ARE BEING TRANSFERED FROM UNIT 1 TO UNIT 8 I3MAX= 1 WING-BODY UPPER

TYPE 3 CARDS ARE BEING TRANSFERED FROM UNIT 1 TO UNIT 8 I3MAX= 1 WING-BODY LOWER

TYPE 3 CARDS ARE BEING TRANSFERED FROM UNIT 1 TO UNIT 8 I3MAX= 1 BODY AREA COVERED BY WING

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

CASE 101A BODY

INPUT SURFACE ELEMENT DATA

N	M	X Y Z	X Y Z	X Y Z	X Y Z	NX NY NZ	XCENT YCENT ZCENT	AREA DELTA V VOLUME	L
1	1	0.0	0.0	-1.14286E+00	-1.14286E+00	0.195208	-7.61904E-01	2.32241E-02	1
		0.0	0.0	3.97076E-02	0.0	0.085800	1.32358E-02	2.63743E-05	
		0.0	0.0	-2.24859E-01	-2.28347E-01	-0.977002	-1.51069E-01	2.63743E-05	
	2	0.0	0.0	-1.14286E+00	-1.14286E+00	0.195228	-7.61904E-01	2.31307E-02	2
		0.0	0.0	7.80653E-02	3.97076E-02	0.252877	3.92576E-02	2.29627E-04	
		0.0	0.0	-2.14623E-01	-2.24859E-01	-0.947597	-1.46494E-01	2.56001E-04	
	3	0.0	0.0	-1.14286E+00	-1.14286E+00	0.195224	-7.61904E-01	2.32215E-02	3
		0.0	0.0	1.14173E-01	7.80653E-02	0.415203	6.40794E-02	6.17830E-04	
		0.0	0.0	-1.97750E-01	-2.14623E-01	-0.888535	-1.37458E-01	8.73831E-04	
	4	0.0	0.0	-1.14286E+00	-1.14286E+00	0.195213	-7.61904E-01	2.32000E-02	4
		0.0	0.0	1.46794E-01	1.14173E-01	0.562430	8.69890E-02	1.13507E-03	
		0.0	0.0	-1.74916E-01	-1.97750E-01	-0.803470	-1.24222E-01	2.00890E-03	
	5	0.0	0.0	-1.14286E+00	-1.14286E+00	0.195218	-7.61904E-01	2.31713E-02	5
		0.0	0.0	1.74916E-01	1.46794E-01	0.693502	1.07237E-01	1.72322E-03	
		0.0	0.0	-1.46794E-01	-1.74916E-01	-0.693502	-1.07237E-01	3.73212E-03	
	6	0.0	0.0	-1.14286E+00	-1.14286E+00	0.195213	-7.61904E-01	2.32000E-02	6
		0.0	0.0	1.97750E-01	1.74916E-01	0.803470	1.24222E-01	2.31556E-03	
		0.0	0.0	-1.14173E-01	-1.46794E-01	-0.562430	-8.69890E-02	6.04768E-03	
	7	0.0	0.0	-1.14286E+00	-1.14286E+00	0.195224	-7.61904E-01	2.32214E-02	7
		0.0	0.0	2.14623E-01	1.97750E-01	0.888535	1.37458E-01	2.83617E-03	
		0.0	0.0	-7.80653E-02	-1.14173E-01	-0.415203	-6.40795E-02	8.88385E-03	
	8	0.0	0.0	-1.14286E+00	-1.14286E+00	0.195228	-7.61904E-01	2.31308E-02	8
		0.0	0.0	2.24859E-01	2.14623E-01	0.947597	1.46494E-01	3.21096E-03	
		0.0	0.0	-3.97076E-02	-7.80653E-02	-0.252877	-3.92576E-02	1.20948E-02	
	9	0.0	0.0	-1.14286E+00	-1.14286E+00	0.195208	-7.61904E-01	2.32241E-02	9
		0.0	0.0	2.28347E-01	2.24859E-01	0.977002	1.51069E-01	3.42775E-03	
		0.0	0.0	0.0	-3.97076E-02	-0.085800	-1.32358E-02	1.55225E-02	
	10	0.0	0.0	-1.14286E+00	-1.14286E+00	0.195208	-7.61904E-01	2.32241E-02	10
		0.0	0.0	2.24859E-01	2.28347E-01	0.977002	1.51069E-01	3.42775E-03	
		0.0	0.0	3.97076E-02	0.0	0.085800	1.32358E-02	1.89503E-02	

FIRST PAGE OF BODY QUADRILATERAL DATA

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0 PAGE 47

CASE 101A BODY

INPUT SURFACE ELEMENT DATA

N	M	X Y Z	X Y Z	X Y Z	X Y Z	NX NY NZ	XCENT YCENT ZCENT	AREA DELTA V VOLUME	L
19	17	-9.42858E+01	-9.42858E+01	-1.00000E+02	-1.00000E+02	-0.138093	-9.70121E+01	2.93040E+00	341
		1.13341E+00	5.75478E-01	4.36558E-01	8.59843E-01	0.256307	7.56039E-01	5.67848E-01	
		3.11395E+00	3.26344E+00	2.47582E+00	2.36243E+00	0.956680	2.82151E+00	3.05506E+03	
	18	-9.42858E+01	-9.42858E+01	-1.00000E+02	-1.00000E+02	-0.138093	-9.70121E+01	2.93067E+00	342
		5.75478E-01	0.0	0.0	4.36558E-01	0.086415	2.54598E-01	6.44779E-02	
		3.26344E+00	3.31384E+00	2.51406E+00	2.47582E+00	0.986642	2.90995E+00	3.05512E+03	

SECTION= 8 TOTAL AREA OF INPUT ELEMENTS = 1310.990 TOTAL NUMBER OF ELEMENTS = 342
TOTAL VOLUME OF INPUT ELEMENTS = 3055.123

********* ********* *********

LAST PAGE OF BODY QUADRILATERAL DATA

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

CASE 101A BODY

INPUT SURFACE ELEMENT DATA

N	M	X Y Z	X Y Z	X Y Z	X Y Z	NX NY NZ	XCENT YCENT ZCENT	AREA DELTA V VOLUME	L
1	1	-4.80315E+01 5.71429E+00 0.0	-5.23172E+01 5.71429E+00 4.28572E-01	-5.69066E+01 7.87402E+00 3.47582E-01	-5.34308E+01 7.87402E+00 0.0	0.096561 0.241403 0.965609	-5.25847E+01 6.75659E+00 1.94743E-01	8.67991E+00 1.41574E+01 1.41574E+01	1
	2	-5.23172E+01 5.71429E+00 4.28572E-01	-5.66030E+01 5.71429E+00 8.57143E-01	-6.03825E+01 7.87402E+00 6.95163E-01	-5.69066E+01 7.87402E+00 3.47582E-01	0.096561 0.241402 0.965609	-5.64795E+01 6.75658E+00 5.84227E-01	8.67993E+00 1.41574E+01 2.83148E+01	2
	3	-5.66030E+01 5.71429E+00 8.57143E-01	-6.08837E+01 5.71429E+00 1.28571E+00	-6.38583E+01 7.87402E+00 1.04274E+00	-6.03825E+01 7.87402E+00 6.95163E-01	0.096561 0.241402 0.965609	-6.03744E+01 6.75659E+00 9.73713E-01	8.67991E+00 1.41574E+01 4.24722E+01	3
	4	-6.08887E+01 5.71429E+00 1.28571E+00	-6.51744E+01 5.71429E+00 1.71429E+00	-6.73341E+01 7.87402E+00 1.39033E+00	-6.38583E+01 7.87402E+00 1.04274E+00	0.096561 0.241402 0.965609	-6.42692E+01 6.75658E+00 1.36320E+00	8.67992E+00 1.41574E+01 5.66296E+01	4
	5	-6.51744E+01 5.71429E+00 1.71429E+00	-6.94601E+01 5.71429E+00 2.14286E+00	-7.08099E+01 7.87402E+00 1.73791E+00	-6.73341E+01 7.87402E+00 1.39033E+00	0.096561 0.241402 0.965609	-6.81640E+01 6.75658E+00 1.75268E+00	8.67990E+00 1.41574E+01 7.07870E+01	5
	6	-6.94601E+01 5.71429E+00 2.14286E+00	-7.37458E+01 5.71429E+00 1.71429E+00	-7.42857E+01 7.87402E+00 1.39033E+00	-7.08099E+01 7.87402E+00 1.73791E+00	-0.098743 0.123429 0.987429	-7.20589E+01 6.75658E+00 1.75268E+00	8.48810E+00 7.07870E+00 7.78657E+01	6
	7	-7.37458E+01 5.71429E+00 1.71429E+00	-7.80315E+01 5.71429E+00 1.28571E+00	-7.77616E+01 7.87402E+00 1.04274E+00	-7.42857E+01 7.87402E+00 1.39033E+00	-0.098743 0.123429 0.987429	-7.59537E+01 6.75659E+00 1.36320E+00	8.48812E+00 7.07871E+00 8.49444E+01	7
	8	-7.80315E+01 5.71429E+00 1.28571E+00	-8.23172E+01 5.71429E+00 8.57143E-01	-8.12374E+01 7.87402E+00 6.95163E-01	-7.77616E+01 7.87402E+00 1.04274E+00	-0.098743 0.123423 0.987429	-7.98486E+01 6.75658E+00 9.73713E-01	8.48810E+00 7.07868E+00 9.20231E+01	8
	9	-8.23172E+01 5.71429E+00 8.57143E-01	-8.66030E+01 5.71429E+00 4.28572E-01	-8.47132E+01 7.87402E+00 3.47582E-01	-8.12374E+01 7.87402E+00 6.95163E-01	-0.098743 0.123429 0.987429	-8.37435E+01 6.75658E+00 5.84228E-01	8.48812E+00 7.07871E+00 9.91018E+01	9
	10	-8.66030E+01 5.71429E+00 4.28572E-01	-9.08887E+01 5.71429E+00 0.0	-8.81890E+01 7.87402E+00 0.0	-8.47132E+01 7.87402E+00 3.47582E-01	-0.098743 0.123429 0.987429	-8.76383E+01 6.75658E+00 1.94745E-01	8.48811E+00 7.07872E+00 1.06180E+02	10

FIRST PAGE OF WING UPPER SURFACE QUADRILATERAL DATA

CASE 101A BODY

INPUT SURFACE ELEMENT DATA

N	M	X Y Z	X Y Z	X Y Z	X Y Z	NX NY NZ	XCENT YCENT ZCENT	AREA DELTA V VOLUME	L
6	6	-7.50281E+01	-7.59730E+01	-7.66036E+01	-7.66035E+01	-0.099730	-7.58683E+01	1.20571E+00	61
		1.46232E+01	1.46232E+01	1.71429E+01	1.71429E+01	0.123411	1.54632E+01	2.30088E+00	
		4.72441E-01	3.77953E-01	0.0	0.0	0.987432	2.83446E-01	4.28106E+02	
	7	-7.59730E+01	-7.69179E+01	-7.66037E+01	-7.66036E+01	-0.098732	-7.64982E+01	1.20568E+00	62
		1.46232E+01	1.46232E+01	1.71429E+01	1.71429E+01	0.123402	1.54632E+01	2.30065E+00	
		3.77953E-01	2.83465E-01	0.0	0.0	0.987433	2.20461E-01	4.30407E+02	
	8	-7.69179E+01	-7.78628E+01	-7.66039E+01	-7.66037E+01	-0.098730	-7.71281E+01	1.20570E+00	63
		1.46232E+01	1.46232E+01	1.71429E+01	1.71429E+01	0.123393	1.54632E+01	2.30054E+00	
		2.83465E-01	1.88976E-01	0.0	0.0	0.987434	1.57470E-01	4.32707E+02	
	9	-7.78628E+01	-7.88077E+01	-7.66040E+01	-7.66039E+01	-0.098734	-7.77580E+01	1.20566E+00	64
		1.46232E+01	1.46232E+01	1.71429E+01	1.71429E+01	0.123385	1.54632E+01	2.30032E+00	
		1.88976E-01	9.44882E-02	0.0	0.0	0.987435	9.44834E-02	4.35007E+02	
	10	-7.88077E+01	-7.97525E+01	-7.66041E+01	-7.66040E+01	-0.098732	-7.83879E+01	1.20568E+00	65
		1.46232E+01	1.46232E+01	1.71429E+01	1.71429E+01	0.123376	1.54632E+01	2.30017E+00	
		9.44882E-02	0.0	0.0	0.0	0.987436	3.14956E-02	4.37307E+02	
	11	-7.97525E+01	-7.97525E+01	-7.66041E+01	-7.66041E+01	0.0	-7.81783E+01	0.0	66
		1.46232E+01	1.46232E+01	1.71429E+01	1.71429E+01	0.0	1.58830E+01	0.0	
		0.0	0.0	0.0	0.0	0.0	0.0	4.37307E+02	

SECTION= TOTAL AREA OF INPUT ELEMENTS = 250.815 TOTAL NUMBER OF ELEMENTS = 66
TOTAL VOLUME OF INPUT ELEMENTS = 437.307

********* ********* *********

1	1	0.0	0.0	-1.12486E-04	-1.12486E-04	0.0	-5.62430E-05	1.26531E-08	67
		0.0	0.0	0.0	0.0	1.000000	0.0	0.0	
		0.0	1.12486E-04	1.12486E-04	0.0	0.0	5.62430E-05	4.37307E+02	

SECTION= TOTAL AREA OF INPUT ELEMENTS = 250.815 TOTAL NUMBER OF ELEMENTS = 67
TOTAL VOLUME OF INPUT ELEMENTS = 437.307

********* ********* *********

LAST PAGE OF WING UPPER SURFACE QUADRILATERAL DATA.

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

CASE 101A BODY

INPUT SURFACE ELEMENT DATA

N	M	X Y Z	X Y Z	X Y Z	X Y Z	NX NY NZ	XCENT YCENT ZCENT	AREA DELTA V VOLUME	L
1	1	-7.66030E+01 1.71429E+01 0.0	-7.66031E+01 1.71429E+01 0.0	-7.12486E+01 1.46232E+01 -9.44882E-02	-7.03037E+01 1.46232E+01 0.0	0.096550 0.241381 -0.965615	-7.27186E+01 1.54631E+01 -3.14943E-02	1.23293E+00 4.60192E+00 4.60192E+00	1
	2	-7.66031E+01 1.71429E+01 0.0	-7.66032E+01 1.71429E+01 0.0	-7.21935E+01 1.46232E+01 -1.88976E-01	-7.12486E+01 1.46232E+01 -9.44882E-02	0.096550 0.241389 -0.965614	-7.33485E+01 1.54632E+01 -9.44814E-02	1.23293E+00 4.60209E+00 9.20401E+00	2
	3	-7.66032E+01 1.71429E+01 0.0	-7.66033E+01 1.71429E+01 0.0	-7.31384E+01 1.46232E+01 -2.83465E-01	-7.21935E+01 1.46232E+01 -1.88976E-01	0.096550 0.241397 -0.965611	-7.39784E+01 1.54631E+01 -1.57472E-01	1.23293E+00 4.60223E+00 1.38062E+01	3
	4	-7.66033E+01 1.71429E+01 0.0	-7.66034E+01 1.71429E+01 0.0	-7.40833E+01 1.46232E+01 -3.77953E-01	-7.31384E+01 1.46232E+01 -2.83465E-01	0.096548 0.241403 -0.965610	-7.46084E+01 1.54632E+01 -2.20459E-01	1.23295E+00 4.60242E+00 1.84087E+01	4
	5	-7.66034E+01 1.71429E+01 0.0	-7.66035E+01 1.71429E+01 0.0	-7.50281E+01 1.46232E+01 -4.72441E-01	-7.40833E+01 1.46232E+01 -3.77953E-01	0.096551 0.241414 -0.965607	-7.52383E+01 1.54632E+01 -2.83448E-01	1.23292E+00 4.60251E+00 2.30112E+01	5
	6	-7.66035E+01 1.71429E+01 0.0	-7.66036E+01 1.71429E+01 0.0	-7.59730E+01 1.46232E+01 -3.77953E-01	-7.50281E+01 1.46232E+01 -4.72441E-01	-0.098730 0.123411 -0.987432	-7.58683E+01 1.54632E+01 -2.83447E-01	1.20571E+00 2.30088E+00 2.53120E+01	6
	7	-7.66036E+01 1.71429E+01 0.0	-7.66037E+01 1.71429E+01 0.0	-7.69179E+01 1.46232E+01 -2.83465E-01	-7.59730E+01 1.46232E+01 -3.77953E-01	-0.098732 0.123402 -0.987433	-7.64982E+01 1.54632E+01 -2.20461E-01	1.20569E+00 2.30066E+00 2.76127E+01	7
	8	-7.66037E+01 1.71429E+01 0.0	-7.66039E+01 1.71429E+01 0.0	-7.78628E+01 1.46232E+01 -1.88976E-01	-7.69179E+01 1.46232E+01 -2.83465E-01	-0.098730 0.123393 -0.987434	-7.71281E+01 1.54632E+01 -1.57470E-01	1.20570E+00 2.30053E+00 2.99132E+01	8
	9	-7.66039E+01 1.71429E+01 0.0	-7.66040E+01 1.71429E+01 0.0	-7.88077E+01 1.46232E+01 -9.44882E-02	-7.78628E+01 1.46232E+01 -1.88976E-01	-0.098734 0.123385 -0.987435	-7.77580E+01 1.54632E+01 -9.44837E-02	1.20567E+00 2.30032E+00 3.22135E+01	9
	10	-7.66040E+01 1.71429E+01 0.0	-7.66041E+01 1.71429E+01 0.0	-7.97525E+01 1.46232E+01 0.0	-7.88077E+01 1.46232E+01 -9.44882E-02	-0.098732 0.123376 -0.987436	-7.83879E+01 1.54632E+01 -3.14959E-02	1.20568E+00 2.30018E+00 3.45137E+01	10

FIRST PAGE OF WING LOWER SURFACE QUADRILATERAL DATA

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

CASE 101A BODY

INPUT SURFACE ELEMENT DATA

N	M	X Y Z	X Y Z	X Y Z	X Y Z	NX NY NZ	XCENT YCENT ZCENT	AREA DELTA V VOLUME	L
6	6	-7.08099E+01	-7.42857E+01	-7.37458E+01	-6.94601E+01	-0.098743	-7.20589E+01	8.48809E+00	61
		7.87402E+00	7.87402E+00	5.71429E+00	5.71429E+00	0.123429	6.75658E+00	7.07869E+00	
		-1.73791E+00	-1.39033E+00	-1.71429E+00	-2.14286E+00	-0.987429	-1.75268E+00	4.08994E+02	
	7	-7.42857E+01	-7.77616E+01	-7.80315E+01	-7.37458E+01	-0.098743	-7.59537E+01	8.48812E+00	62
		7.87402E+00	7.87402E+00	5.71429E+00	5.71429E+00	0.123429	6.75659E+00	7.07871E+00	
		-1.39033E+00	-1.04274E+00	-1.28571E+00	-1.71429E+00	-0.987429	-1.36320E+00	4.16073E+02	
	8	-7.77616E+01	-8.12374E+01	-8.23172E+01	-7.80315E+01	-0.098743	-7.98486E+01	8.48812E+00	63
		7.87402E+00	7.87402E+00	5.71429E+00	5.71429E+00	0.123428	6.75658E+00	7.07869E+00	
		-1.04274E+00	-6.95163E-01	-8.57143E-01	-1.28571E+00	-0.987429	-9.73712E-01	4.23152E+02	
	9	-8.12374E+01	-8.47132E+01	-8.66030E+01	-8.23172E+01	-0.098743	-8.37435E+01	8.48812E+00	64
		7.87402E+00	7.87402E+00	5.71429E+00	5.71429E+00	0.123429	6.75658E+00	7.07871E+00	
		-6.95163E-01	-3.47582E-01	-4.28572E-01	-8.57143E-01	-0.987429	-5.84228E-01	4.30230E+02	
	10	-8.47132E+01	-8.81890E+01	-9.08887E+01	-8.66030E+01	-0.098743	-8.76383E+01	8.48811E+00	65
		7.87402E+00	7.87402E+00	5.71429E+00	5.71429E+00	0.123429	6.75658E+00	7.07872E+00	
		-3.47582E-01	0.0	0.0	-4.28572E-01	-0.987429	-1.94745E-01	4.37309E+02	
	11	-8.81890E+01	-8.81890E+01	-9.08887E+01	-9.08887E+01	0.0	-8.95388E+01	0.0	66
		7.87402E+00	7.87402E+00	5.71429E+00	5.71429E+00	0.0	6.79415E+00	0.0	
		0.0	0.0	0.0	0.0	0.0	0.0	4.37309E+02	

SECTION= TOTAL AREA OF INPUT ELEMENTS = 250.816 TOTAL NUMBER OF ELEMENTS = 66
TOTAL VOLUME OF INPUT ELEMENTS = 437.309

********* ********* *********

1	1	0.0	0.0	-1.12486E-04	-1.12486E-04	0.0	-5.62430E-05	1.26531E-08	67
		0.0	0.0	0.0	0.0	1.000000	0.0	0.0	
		0.0	1.12486E-04	1.12486E-04	0.0	0.0	5.62430E-05	4.37309E+02	

SECTION= TOTAL AREA OF INPUT ELEMENTS = 250.816 TOTAL NUMBER OF ELEMENTS = 67
TOTAL VOLUME OF INPUT ELEMENTS = 437.309

********* ********* *********

LAST PAGE OF WING LOWER SURFACE QUADRILATERAL DATA

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0 PAGE 63

CASE 101A BODY

INPUT SURFACE ELEMENT DATA

N	M	X Y Z	X Y Z	X Y Z	X Y Z	NX NY NZ	XCENT YCENT ZCENT	AREA DELTA V VOLUME	L
1	11	-9.38160E+01 3.37244E+00 0.0	-9.38160E+01 3.37244E+00 0.0	-9.08887E+01 5.71429E+00 0.0	-9.08887E+01 5.71429E+00 0.0	0.0 0.0 0.0	-9.23523E+01 4.54336E+00 0.0	0.0 0.0 2.50611E+01	11

SECTION= WBU TOTAL AREA OF INPUT ELEMENTS = 34.769 TOTAL NUMBER OF ELEMENTS = 11
TOTAL VOLUME OF INPUT ELEMENTS = 25.061

********* ********* *********

LAST PAGE OF WING-BODY UPPER-SURFACE QUADRILATERAL

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

CASE 101A BODY

INPUT SURFACE ELEMENT DATA

N	M	X Y Z	X Y Z	X Y Z	X Y Z	NX NY NZ	XCENT YCENT ZCENT	AREA DELTA V VOLUME	L
1	11	-9.08887E+01 5.71429E+00 0.0	-9.08887E+01 5.71429E+00 0.0	-9.38160E+01 3.37244E+00 0.0	-9.38160E+01 3.37244E+00 0.0	0.0 0.0 0.0	-9.23523E+01 4.54336E+00 0.0	0.0 0.0 2.50611E+01	11

SECTION= WBL TOTAL AREA OF INPUT ELEMENTS = 34.769 TOTAL NUMBER OF ELEMENTS = 11
TOTAL VOLUME OF INPUT ELEMENTS = 25.061

********* ********* *********

LAST PAGE OF WING-BODY LOWER SURFACE QUADRILATERAL DATA

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

CASE 101A BODY

INPUT SURFACE ELEMENT DATA

N	M	X Y Z	X Y Z	X Y Z	X Y Z	NX NY NZ	XCENT YCENT ZCENT	AREA DELTA V VOLUME	L
1	1	-4.76294E+01	-4.76294E+01	-6.89651E+01	-6.89651E+01	-0.004457	-6.18532E+01	2.49513E+01	1
		5.55298E+00	5.55298E+00	5.45557E+00	4.94938E+00	0.976290	5.31931E+00	1.29577E+02	
		0.0	0.0	0.0	-2.28346E+00	-0.216420	-7.61155E-01	1.29577E+02	
	2	-4.76294E+01	-4.76294E+01	-6.89651E+01	-6.89651E+01	-0.004457	-6.18532E+01	2.49513E+01	2
		5.55298E+00	5.55298E+00	4.94938E+00	5.45557E+00	0.976290	5.31931E+00	1.29577E+02	
		0.0	0.0	2.28346E+00	0.0	0.216420	7.61155E-01	2.59153E+02	
2	1	-6.89651E+01	-6.89651E+01	-9.38160E+01	-9.38160E+01	-0.081566	-7.72487E+01	2.91589E+01	3
		4.94938E+00	5.45557E+00	3.37244E+00	3.37244E+00	0.973047	4.59246E+00	1.30302E+02	
		-2.28346E+00	0.0	0.0	0.0	-0.215701	-7.61156E-01	3.89455E+02	
	2	-6.89651E+01	-6.89651E+01	-9.38160E+01	-9.38160E+01	-0.081566	-7.72487E+01	2.91589E+01	4
		5.45557E+00	4.94938E+00	3.37244E+00	3.37244E+00	0.973047	4.59246E+00	1.30302E+02	
		0.0	2.28346E+00	0.0	0.0	0.215701	7.61155E-01	5.19756E+02	

SECTION= C TOTAL AREA OF INPUT ELEMENTS = 108.220 TOTAL NUMBER OF ELEMENTS = 4
TOTAL VOLUME OF INPUT ELEMENTS = 519.756

********* ********* *********

BODY AREA COVERED BY THE WING QUADRILATERAL DATA

```
******** MAIN PROGRAM NOW HAS CONTROL OF SYSTEM ********

***** PROGRAM HAS REACHED NORMAL TERMINATION ******
```

APPENDIX B

ILLUSTRATED EXAMPLE PRESSURE INPUT, FORCE CALCULATION, AND SUMMATIONS

This section provides a listing of the card input required to input surface pressures, calculate the forces, and perform the summations for Configuration 101A to duplicate the results of the illustrated example (Section 4). The listing is output by the program (INMONT = 0) and does not include the Executive-Flag Card. This card should be added to the deck in any attempt to recreate the run from the listing. Further, the input assumes the geometry has been previously loaded and stored on Unit 4 (Appendix A).

Each page of the input listing contains a header which includes two lines indicating the card columns. Identification of several of the card types has been added on the right-hand margin. The input listing is given on the first seven pages. The remaining pages present selected portions of the printed output. Various annotations are given on the right-hand margins of these pages as additional aids to the user.

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0 PAGE 1

SUPERSONIC-HYPERSONIC ARBITRARY-BODY PROGRAM INPUT DATA

```
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

222                                                                               SYSTEM CONTROL CARD
 CASE 101A, WING AFT (NASA TN D-6441)               00             1000  0.0      AERO TITLE CARD
113                          0                                                    AERO FLAG CARD
4.5                                               0 1                             FLIGHT CONDITION CARD
 102.58262 100.0       34.2857    0.0          0.0         0.0                    REFERENCE DIMENSION
 6.0       0.0                                                                    α-β CARD
0                              SURFACE CP DATA FOR CASE 101A
10 1 4.5                       MACH = 4.5
10 1 6.0   0.0                 ALPHA = 6.0
00 12 0                        REGION 1, BODY FORWARD OF WING
1  601     0.0                 C.0         C.0         C.0        0.0        0.0
   -0.38095    .0065992  -.075553   0.20200
   -0.38095    .032021   -.068729   0.19801
   -0.38095    .053618   -.053618   0.16646
   -0.38095    .068729   -.032021   0.13517
   -0.38095    .075553   -.0065992  0.10187
   -0.38095    .075553   +.0065992  0.086903
   -0.38095    .068729    .032021   0.060396
   -0.38095    .053618    .053618   0.041867
   -0.38095    .032058    .068729   0.030717
   -0.38095    .0065992   .075553   0.026202
   -4.4385     .074344   -.84977    0.18601
   -4.4385     .36049    -.77310    0.17514
   -4.4385     .60318    -.60318    0.15212
   -4.4385     .77310    -.36049    0.12200
   -4.4385     .84977    -.074344   0.09068
   -4.4385     .84977    +.074344   0.07615
   -4.4385     .77310     .36049    0.051691
   -4.4385     .60318     .60318    0.034425
   -4.4385     .36049     .77310    0.024272
   -4.4385     .074344    .84977    0.020208
  -10.062      .16009    -1.8297    0.16475
  -10.062      .77624    -1.6647    0.15448
  -10.062     1.2988     -1.2988    0.13282
  -10.062     1.6647     -0.77624   0.10465
  -10.062     1.8297     -0.16009   0.075642
  -10.062     1.8297     +0.16009   0.062337
  -10.062     1.6647      0.77624   0.040233
  -10.062     1.2988      1.2988    0.025013
  -10.062      .77624     1.6647    0.016297
  -10.062      .16009     1.8297    0.012985
```

FIRST AERO OPTION
FLOW FIELD ANALYSIS
INPUT C_p DATA
REGION 1

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0 PAGE 2

SUPERSONIC-HYPERSONIC ARBITRARY-BODY PROGRAM INPUT DATA

```
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

 -21.453    .30347  -3.4686    0.12354
 -21.453   1.4715   -3.1556    0.11457
 -21.453   2.4621   -2.4621    0.095833
 -21.453   3.1556   -1.4715    0.071930
 -21.453   3.4686   -0.30347   0.047964
 -21.453   3.4686   +0.30347   0.037309
 -21.453   3.1556    1.4715    0.020303
 -21.453   2.4621    2.4621    0.0094728
 -21.453   1.4715    3.1556    0.0039227
 -21.453    .30347   3.4686    0.0020485
 -32.869    .40698  -4.6516    0.088772
 -32.869   1.9734   -4.2319    0.081105
 -32.869   3.3018   -3.3018    0.065353
 -32.869   4.2319   -1.9734    0.045631
 -32.869   4.6516   -0.40698   0.026682
 -32.869   4.6516   +0.40698   0.018653
 -32.869   4.2319    1.9734    0.0067271
 -32.869   3.3018    3.3018    0.0007078
 -32.869   1.9734    4.2319   -0.0046350
 -32.869    .40698   4.6516   -0.0085663
 -47.147    .48072  -5.4947    0.051673
 -47.147   2.3310   -4.9989    0.045778
 -47.147   3.9001   -3.9001    0.033903
 -47.147   4.9989   -2.3310    0.019807
 -47.147   5.4947   -0.48072   0.0074304
 -47.147   5.4947   +0.48072   0.0030103
 -47.147   4.9989    2.3310   -0.0047123
 -47.147   3.9001    3.9001   -0.016550
 -47.147   2.3310    4.9989   -0.023782
 -47.147    .48072   5.4947   -0.026783
02 22 0              REGION 2, BODY ALONG WING AND AFT
1  601     0.0       0.0       0.0       0.0       0.0       0.0
 -50.003    .48805  -5.5783    0.046293
 -50.003   2.3665   -5.0750    0.040700
 -50.003   3.9595   -3.9595    0.029514
 -50.003   5.0750   -2.3665    0.016360
 -50.003   5.5783   -0.48805   0.0051931
 -50.003   5.5783   +0.48805   0.0014632
 -50.003   5.0750    2.3665   -0.0083308
 -50.003   3.9595    3.9595   -0.019633
 -50.003   2.3665    5.0750   -0.026538
```

INPUT Cp DATA
REGION 1

INPUT Cp DATA
REGION 2

SUPERSONIC-HYPERSONIC ARBITRARY-BODY PROGRAM INPUT DATA

```
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
 -50.003    .48305   5.5783   -0.029399
 -61.427    .49303  -5.6352    0.024925
 -61.427   2.3906   -5.1267    0.020741
 -61.427   3.9999   -3.9999    0.012687
 -61.427   5.1267   -2.3906    0.0040685
 -61.427   5.6352   -0.49303  -0.0025626
 -61.427   5.6352   +0.49303  -0.010244
 -61.427   5.1267    2.3906   -0.023072
 -61.427   3.9999    3.9999   -0.032163
 -61.427   2.3906    5.1267   -0.037692
 -61.427    .49303   5.6352   -0.039971
 -72.350    .45867  -5.2426    0.0097547
 -72.350   2.2241   -4.7695    0.0069844
 -72.350   3.7212   -3.7212    0.0021718
 -72.350   4.7695   -2.2241   -0.0042776
 -72.350   5.2426   -0.45867  -0.018117
 -72.350   5.2426   +0.45867  -0.024334
 -72.350   4.7695    2.2241   -0.034665
 -72.350   3.7212    3.7212   -0.041958
 -72.350   2.2241    4.7695   -0.046359
 -72.350    .45867   5.2426   -0.048177
 -81.416    .40693  -4.6516    0.0021235
 -81.416   1.9734   -4.2319    0.0007222
 -81.416   3.3018   -3.3018   -0.0045824
 -81.416   4.2319   -1.9734   -0.016400
 -81.416   4.6516   -0.40693  -0.028110
 -81.416   4.6516   +0.40698  -0.033366
 -81.416   4.2319    1.9734   -0.042052
 -81.416   3.3018    3.3018   -0.048158
 -81.416   1.9734    4.2319   -0.049383
 -81.416    .40693   4.6516   -0.049383
 -89.930    .33305  -3.3067   -0.0039767
 -89.930   1.6149   -3.4632   -0.0079396
 -89.930   2.7020   -2.7020   -0.016044
 -89.930   3.4632   -1.6149   -0.026146
 -89.930   3.8067   -0.33305  -0.036109
 -89.930   3.8067   +0.33305  -0.040561
 -89.930   3.4632    1.6149   -0.047906
 -89.930   2.7020    2.7020   -0.049383
 -89.930   1.6149    3.4632   -0.049383
 -89.930    .33305   3.3067   -0.049383
```

INPUT Cp DATA
REGION 2

```
SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0
CASE       0                                                                                        PAGE    4

SUPERSONIC-HYPERSONIC ARBITRARY-BODY PROGRAM INPUT DATA
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
   -93.535    .23670  -2.7056    -0.016066
   -93.535   1.1478   -2.4615    -0.019415
   -93.535   1.9204   -1.9204    -0.026277
   -93.535   2.4615   -1.1478    -0.034802
   -93.535   2.7056   -0.2367    -0.043130
   -93.535   2.7056   +0.2367    -0.046906
   -93.535   2.4615    1.1478    -0.049383
   -93.535   1.9204    1.9204    -0.049383
   -93.535   1.1479    2.4615    -0.049383
   -93.535    .2367    2.7056    -0.049383
02 32 0             REGION 3, WING UPPER FORWARD SECTION
1  151       0.0        0.0       0.0       0.0       0.0       0.0
-  52.535        6.7566     0.19474 -0.023078
-  55.430        6.7566     0.58423 -0.024895
-  60.374        6.7566     0.97371 -0.025654
-  64.269        6.7566     1.3632  -0.025520
-  63.154        6.7566     1.7527  -0.019844
-  60.051        9.9855     0.13419 -0.014405
-  62.735        9.9855     0.40260 -0.016300
-  65.419        9.9355     0.67098 -0.017744
-  68.103        9.9855     0.93936 -0.019734
-  70.787        9.9855     1.2078  -0.019366
-  67.907       13.383      0.07051 -0.006283
-  69.317       13.383      0.21152 -0.007433
-  70.727       13.383      0.35253 -0.008469
-  72.133       13.383      0.49354 -0.001773
-  73.543       13.383      0.63455 -0.003162
02 42 0             REGION 4, WING UPPER AFT SECTION
1  151       0.0        0.0       0.0       0.0       0.0       0.0
-  72.039        6.7566     1.7527  -0.059307
-  75.934        6.7566     1.3632  -0.060071
-  79.349        6.7566     0.97371 -0.060742
-  83.743        6.7566     0.58423 -0.061331
-  87.633        6.7566     0.19474 -0.061946
-  73.471        9.9855     1.2078  -0.057419
-  75.155        9.9855     0.93937 -0.058041
-  78.339        9.9855     0.67098 -0.058602
-  81.523        9.9855     0.40260 -0.059105
-  84.207        9.9855     0.13419 -0.059558
-  74.753       13.383      0.63455 -0.054594
-  75.363       13.383      0.49354 -0.054997
```

INPUT C_p DATA REGION 2

INPUT C_p DATA REGION 3

INPUT C_p DATA REGION 4

SUPERSONIC-HYPERSONIC ARBITRARY-BODY PROGRAM INPUT DATA

```
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
- 77.778     13.383      0.35253 -0.055380
- 79.188     13.383      0.21152 -0.055743
- 80.598     13.383      0.07051 -0.056086
02 52 0                 REGION 5, WING LOWER FORWARD SECTION
1  151      0.0         0.0         0.0         0.0         0.0
- 52.535      6.7566    -0.19474   0.12037
- 55.480      6.7566    -0.58423   0.11110
- 60.374      6.7566    -0.97371   0.10357
- 64.269      6.7566    -1.3632    0.097594
- 68.164      6.7566    -1.7527    0.092979
- 60.051      9.9855    -0.13419   0.12335
- 62.735      9.9855    -0.40260   0.11787
- 65.419      9.9855    -0.67098   0.11306
- 68.103      9.9855    -0.93936   0.10898
- 70.787      9.9855    -1.2078    0.10550
- 67.907     13.383     -0.07051   0.12669
- 69.317     13.383     -0.21152   0.12445
- 70.727     13.383     -0.35253   0.12232
- 72.133     13.383     -0.49354   0.12031
- 73.548     13.383     -0.63455   0.11842
12 52 0                 REGION 6, WING LOWER AFT SECTION
1  151      0.0         0.0         0.0         0.0         0.0
- 72.059      6.7566    -1.7527   -0.017292
- 75.354      6.7566    -1.3632   -0.018803
- 79.849      6.7566    -0.97371  -0.020736
- 83.743      6.7566    -0.58423  -0.023062
- 87.638      6.7566    -0.19474  -0.025726
- 73.471      9.9855    -1.2078   -0.013684
- 75.155      9.9855    -0.93937  -0.015179
- 78.339      9.9855    -0.67098  -0.016695
- 81.523      9.9855    -0.40260  -0.018281
- 84.207      9.9855    -0.13419  -0.019880
- 74.353     13.383     -0.63455  -0.004509
- 75.358     13.383     -0.49354  -0.005817
- 77.778     13.383     -0.35253  -0.007062
- 79.188     13.383     -0.21152  -0.008244
- 80.598     13.383     -0.07051  -0.009367
1
11 1
11 1
01 12 1
```

INPUT Cp DATA
REGION 5

INPUT Cp DATA
REGION 6

FLOW FIELD ANALYSIS

REVIEW INPUT
Cp DATA

```
SUPERSONIC-HYPERSONIC ARBITRARY-BODY PROGRAM INPUT DATA

0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

01 22 1
01 32 1
01 42 1
01 52 1
11 52 1
 50    CASE 101A  SAMPLE PROBLEM USING INPUT PRESSURES
 1                1
    111
  1 1 1400                                   0                 INVISCID PRESSURES
 0.0        0.0       0.0
-43.0       0.0       0.0
 0.00       0.0      -1.0                                      COMPONENT 1
 0.0        0.0      +1.0
 1                1
    111
  1 1 2400                                   0
-43.0       0.0       0.0
-100.0      0.0       0.0
 0.0        0.0      -1.0
 0.0        0.0      +1.0                                      COMPONENT 2
 2 4              1
    111
  1 1 3110                                   0
-47.6234    4.9494    0.0
-63.9531    0.0       0.0
-76.5029   17.1429    0.0                                      COMPONENT 3
-76.5029    0.0       0.0
 2 4              1
    111
  1 1 4110                                   0
-63.9651    3.3725    0.0                                      COMPONENT 4
-73.3159    0.0       0.0
-76.5027   17.1429    0.0
-76.5029    0.0       0.0
 3 5              1
    111
  1 1 5110                                   0
-47.6234    4.9494    0.0
-63.9531    0.0       0.0
-76.5029   17.1429    0.0
-76.5029    0.0       0.0                                      COMPONENT 5
```

SUPERSONIC-HYPERSONIC ARBITRARY-BODY PROGRAM INPUT DATA

```
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
 3 5                  1
   111
  1 1 6110                                  0
-68.9651   3.3725     0.0
-93.8159   0.0        0.0
-76.6029  17.1429     0.0
-76.6029   0.0        0.0
 CASE 101A, WING AFT (NASA TN D-6441)        00              000  0.0
3                          1
 4.5                                     0 1
-102.58262 100.0      34.2857   0.0       0.0        0.0
 6.0       0.0
 11   CASE 101A  SAMPLE PROBLEM USING INPUT PRESSURES
 6                    1
   111
  1 1 2400                                  0
-49.0      0.0        0.0
-100.0     0.0        0.0
 0.0       0.0       -1.0
 0.0       0.0       +1.0
 CASE 101A, WING AFT (NASA TN D-6441)        00              000  0.0
5                          0
 4.5                                     0 1
 102.58262 100.0      34.2857   0.0       0.0        0.0
 6.0       0.0
1
C1010      1 2
C1010      7
C2010      1 2
C1010      3 4
C1010      5 6
C2010      4 5
12010      3 6
```

COMPONENT 6

2ND AERO OPTION

INVISCID PRESSURES

COMPONENT 7

(NEGATIVE SREF)

3RD AERO OPTION

SUMMATIONS

COMPONENT BUILDUP

```
***SUPERSONIC-HYPERSONIC ARBITRARY-BODY AERODYNAMIC COMPUTER PROGRAM SYSTEM *************

PROGRAM OPTIONS ARE IN THE FOLLOWING ORDER....

      1  AERODYNAMIC PROGRAM  (OPTION 2)
      2  AERODYNAMIC PROGRAM  (OPTION 2)
      3  AERODYNAMIC PROGRAM  (OPTION 2)
```

```
***** AERODYNAMIC PROGRAM *****
 CASE 101A, WING AFT (NASA TN D-6441)
     RUN ID =    0      CORRELATION ID =    0
                 1 FLOW FIELD ANALYSIS (OPTION 1)
                 2 FLOW FIELD ANALYSIS (OPTION 1)
                 3 INVISCID PRESSURES  (OPTION 3)
```

1ST AERO OPTION

```
** FLOW-FIELD OPTION **
     ** TITLER=REGION 1, BODY FORWARD OF WING                IDTYP= 2 2 0 0 0   NSREG=  1   ITFLAG=  1
     ** TITLER=REGION 2, BODY ALONG WING AND AFT             IDTYP= 2 2 0 0 0   NSREG=  1   ITFLAG=  1
     ** TITLER=REGION 3, WING UPPER FORWARD SECTION          IDTYP= 2 2 0 0 0   NSREG=  1   ITFLAG=  1
     ** TITLER=REGION 4, WING UPPER AFT SECTION              IDTYP= 2 2 0 0 0   NSREG=  1   ITFLAG=  1
     ** TITLER=REGION 5, WING LOWER FORWARD SECTION          IDTYP= 2 2 0 0 0   NSREG=  1   ITFLAG=  1
     ** TITLER=REGION 6, WING LOWER AFT SECTION              IDTYP= 2 2 0 0 0   NSREG=  1   ITFLAG=  1
** TITLEA=ALPHA = 6.0                          ALPHA= 6.000 BETA= 0.0   NGROUP=   6
** LORG=  10  76 142 163 184 205   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
** TITLES=MACH = 4.5                          MACH= 4.500 NAB=   1
** LTAB=   7   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
**ITAB=   1 226   0   0   4   0   0   0   0
```

1ST FLOW FIELD OPTION.

LOADING OF INPUT PRESSURE
DATA TO UNIT 10.

```
** FLOW-FIELD OPTION **

SURFACE DATA ARE BEING READ FROM FLOW-FIELD UNIT

TITLE4=SURFACE CP DATA FOR CASE 101A                    IMTAB=    1 226    0    0    4    0    0    0    0
TITLE5=MACH = 4.5                                       MACH =  4.500 NAB=   1
    LOAB=    7    0    0    0    0    0    0    0    0   0    0    0    0    0    0    0    0    0    0    0
TITLE6=ALPHA = 6.0                                      ALPHA=  6.000 BETA=  0.0    NREG  =   6
    LORS=   10   76  142  163  184  205    0    0    0   0    0    0    0    0    0    0    0    0    0    0
TITLE7=REGION 1, BODY FORWARD OF WING                   IDTYP=  2  2  0  0  0

XO=   0.0      YO=   0.0      ZO=   0.0      PSIO=   0.0      THETO=   0.0      PHIO=   0.0
```

2ND FLOW FIELD OPTION.

IMAGING OF PRESSURE
DATA ON UNIT 10.

1st PAGE OF REGION L PRESSURE DATA

FLOW REGION 1

SUB-REGION 1 NUMBER OF SECONDARY FLOWS 0

LOFF	16	0	0	0	0
IFC	60	0	0	0	0
LOCD	0	0	0	0	0
IFD	0	0	0	0	0
LOSF	0	0	0	0	0

BOUNDARY 1 DATA

-0.3309	0.0066	-0.0756	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2020	0.0
-0.3309	0.0320	-0.0687	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1980	0.0
-0.3309	0.0536	-0.0536	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1665	0.0
-0.3309	0.0687	-0.0320	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1352	0.0
-0.3309	0.0756	-0.0066	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1019	0.0
-0.3309	0.0756	0.0066	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0869	0.0
-0.3309	0.0687	0.0320	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0604	0.0
-0.3309	0.0536	0.0536	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0419	0.0
-0.3309	0.0321	0.0687	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0307	0.0
-0.3309	0.0066	0.0756	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0262	0.0
-4.4385	0.0743	-0.8498	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1860	0.0
-4.4385	0.3605	-0.7731	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1751	0.0
-4.4385	0.6032	-0.6032	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1521	0.0
-4.4385	0.7731	-0.3605	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1220	0.0
-4.4385	0.8498	-0.0743	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0907	0.0
-4.4385	0.8498	0.0743	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0762	0.0
-4.4385	0.7731	0.3605	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0517	0.0
-4.4385	0.6032	0.6032	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0344	0.0
-4.4385	0.3605	0.7731	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0243	0.0
-4.4385	0.0743	0.8498	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0202	0.0
-10.0620	0.1601	-1.8297	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1647	0.0
-10.0620	0.7762	-1.6647	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1545	0.0
-10.0620	1.2988	-1.2988	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1328	0.0
-10.0620	1.6647	-0.7762	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1047	0.0
-10.0620	1.8297	-0.1601	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0756	0.0
-10.0620	1.8297	0.1601	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0623	0.0
-10.0620	1.6647	0.7762	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0402	0.0
-10.0620	1.2988	1.2988	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0250	0.0
-10.0620	0.7762	1.6647	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0163	0.0
-10.0620	0.1601	1.8297	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0129	0.0
-21.4533	0.3035	-3.4686	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1235	0.0
-21.4533	1.4715	-3.1556	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1146	0.0
-21.4533	2.4621	-2.4621	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0958	0.0
-21.4533	3.1556	-1.4715	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0719	0.0
-21.4533	3.4686	-0.3035	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0480	0.0
-21.4533	3.4686	0.3035	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0373	0.0
-21.4533	3.1556	1.4715	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0203	0.0
-21.4533	2.4621	2.4621	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0095	0.0
-21.4533	1.4715	3.1556	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0039	0.0
-21.4533	0.3035	3.4686	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0020	0.0
-32.3699	0.4070	-4.6516	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0889	0.0
-32.3699	1.9734	-4.2319	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0911	0.0
-32.3699	3.3018	-3.3018	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0654	0.0

FLOW REGION 6

SUB-REGION 1 NUMBER OF SECONDARY FLOWS 0

FLOW REGION 6
PRESSURE DATA

LOFF	211	0	0	0
IFC	15	0	0	0
LOCD	0	0	0	0
IFD	0	0	0	0
LOSF	0	0	0	0

BOUNDARY 1 DATA

-72.1591	6.7566	-1.7527	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0173	0.0
-73.9543	6.7566	-1.3632	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0188	0.0
-73.349)	6.7566	-0.9737	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0207	0.0
-33.743)	6.7566	-0.5842	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0231	0.0
-67.538)	6.7566	-0.1947	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0257	0.0
-73.471)	9.9855	-1.2078	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0137	0.0
-75.155)	9.9855	-0.9394	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0152	0.0
-79.139)	9.9855	-0.6710	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0167	0.0
-31.123)	9.9855	-0.4026	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0183	0.0
-34.237)	9.9855	-0.1342	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0199	0.0
-74.358)	13.3933	-0.6345	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0045	0.0
-76.358)	13.3833	-0.4935	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0058	0.0
-77.778)	13.3833	-0.3525	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0071	0.0
-79.138)	13.3833	-0.2115	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0082	0.0
-80.598)	13.3933	-0.0705	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0094	0.0

*** PRESSURE CALCULATION PROGRAM

INVISCID PRESSURES OPTION

```
PREVIOUSLY SAVED SURFACE DATA WILL BE USED TO CALCULATE  ORCES
     LASTR= 0    NDSET= 1    IABSET= 1    IR = 1    INORM= 4    ISURF= 0
IPF= 0    ISR=  0  0  0  0  0  0  0  0  0  0    ISF= 0 0 0 0 0
```

ACCESS OF REGION 1
C_p DATA FROM UNIT 10.

BOUNDARY DATA, 2 POINTS.

I	X1	Y1	X2	Y2
1	0.0	0.027732	1.000000	0.027732
2	0.0	1.000000	1.000000	1.000000

NORMALIZATION PARAMETERS

NORMALIZED C_p (%)
DATA FOR REGION 1

NORMALIZED SURFACE DATA. NOSET = 1 IABSET = 1 REGION = 1

I	XI	YI	MACH	MX	MY	MZ	P/P1	T/T1	ZI
1	0.007936	0.027732	0.0	0.0	0.0	0.0	0.202000	0.0	0.075841
2	0.007936	0.138783	0.0	0.0	0.0	0.0	0.198010	0.0	0.075822
3	0.007936	0.250000	0.0	0.0	0.0	0.0	0.166460	0.0	0.075827
4	0.007936	0.361217	0.0	0.0	0.0	0.0	0.135170	0.0	0.075822
5	0.007936	0.472267	0.0	0.0	0.0	0.0	0.101870	0.0	0.075841
6	0.007936	0.527733	0.0	0.0	0.0	0.0	0.086903	0.0	0.075841
7	0.007936	0.638783	0.0	0.0	0.0	0.0	0.060396	0.0	0.075822
8	0.007936	0.750000	0.0	0.0	0.0	0.0	0.041867	0.0	0.075827
9	0.007936	0.861076	0.0	0.0	0.0	0.0	0.030717	0.0	0.075838
10	0.007936	0.972267	0.0	0.0	0.0	0.0	0.026202	0.0	0.075841
11	0.092469	0.027777	0.0	0.0	0.0	0.0	0.196010	0.0	0.853016
12	0.092469	0.138885	0.0	0.0	0.0	0.0	0.175140	0.0	0.853016
13	0.092469	0.250000	0.0	0.0	0.0	0.0	0.152120	0.0	0.853025
14	0.092469	0.361115	0.0	0.0	0.0	0.0	0.122000	0.0	0.853016
15	0.092469	0.472223	0.0	0.0	0.0	0.0	0.090680	0.0	0.853016
16	0.092469	0.527777	0.0	0.0	0.0	0.0	0.076150	0.0	0.853016
17	0.092469	0.638885	0.0	0.0	0.0	0.0	0.051691	0.0	0.853016
18	0.092469	0.750000	0.0	0.0	0.0	0.0	0.034425	0.0	0.853025
19	0.092469	0.861115	0.0	0.0	0.0	0.0	0.024272	0.0	0.853016
20	0.092469	0.972223	0.0	0.0	0.0	0.0	0.020208	0.0	0.853016
21	0.206625	0.027780	0.0	0.0	0.0	0.0	0.164750	0.0	1.836690
22	0.206625	0.138885	0.0	0.0	0.0	0.0	0.154480	0.0	1.836783
23	0.206625	0.250000	0.0	0.0	0.0	0.0	0.132820	0.0	1.836781
24	0.206625	0.361114	0.0	0.0	0.0	0.0	0.104650	0.0	1.836783
25	0.206625	0.472220	0.0	0.0	0.0	0.0	0.075642	0.0	1.836690
26	0.206625	0.527780	0.0	0.0	0.0	0.0	0.062337	0.0	1.836690
27	0.206625	0.638886	0.0	0.0	0.0	0.0	0.040233	0.0	1.836783
28	0.206625	0.750000	0.0	0.0	0.0	0.0	0.025013	0.0	1.836781
29	0.206625	0.861114	0.0	0.0	0.0	0.0	0.016257	0.0	1.836783
30	0.206625	0.972220	0.0	0.0	0.0	0.0	0.012885	0.0	1.836690
31	0.446938	0.027778	0.0	0.0	0.0	0.0	0.123540	0.0	3.481851
32	0.446938	0.138891	0.0	0.0	0.0	0.0	0.114570	0.0	3.481828
33	0.446938	0.250000	0.0	0.0	0.0	0.0	0.095933	0.0	3.481836
34	0.446938	0.361109	0.0	0.0	0.0	0.0	0.071930	0.0	3.481828
35	0.446938	0.472221	0.0	0.0	0.0	0.0	0.047964	0.0	3.481851
36	0.446938	0.527779	0.0	0.0	0.0	0.0	0.037309	0.0	3.481851
37	0.446938	0.638891	0.0	0.0	0.0	0.0	0.020303	0.0	3.481828
38	0.446938	0.750000	0.0	0.0	0.0	0.0	0.009473	0.0	3.481836
39	0.446938	0.861109	0.0	0.0	0.0	0.0	0.003923	0.0	3.481828
40	0.446938	0.972221	0.0	0.0	0.0	0.0	0.002049	0.0	3.481851
41	0.684771	0.027779	0.0	0.0	0.0	0.0	0.089772	0.0	4.669368
42	0.684771	0.138891	0.0	0.0	0.0	0.0	0.081105	0.0	4.669397
43	0.684771	0.250000	0.0	0.0	0.0	0.0	0.065353	0.0	4.669449
44	0.684771	0.361109	0.0	0.0	0.0	0.0	0.045631	0.0	4.669397
45	0.684771	0.472221	0.0	0.0	0.0	0.0	0.026692	0.0	4.669368
46	0.684771	0.527779	0.0	0.0	0.0	0.0	0.013653	0.0	4.669368
47	0.684771	0.638891	0.0	0.0	0.0	0.0	0.006727	0.0	4.669397
48	0.684771	0.750000	0.0	0.0	0.0	0.0	0.000708	0.0	4.669449
49	0.684771	0.861109	0.0	0.0	0.0	0.0	-0.004635	0.0	4.669397
50	0.684771	0.972221	0.0	0.0	0.0	0.0	-0.003566	0.0	4.669368
51	0.993229	0.027778	0.0	0.0	0.0	0.0	0.051673	0.0	5.515687

52	0.982229	0.138888	0.0	0.0	0.0	0.0	0.045778	0.0	5.51556
53	0.982229	0.250000	0.0	0.0	0.0	0.0	0.033903	0.0	5.51557
54	0.982229	0.361112	0.0	0.0	0.0	0.0	0.019807	0.0	5.51566
55	0.982229	0.472222	0.0	0.0	0.0	0.0	0.007480	0.0	5.51569
56	0.982229	0.527778	0.0	0.0	0.0	0.0	0.003010	0.0	5.51568
57	0.982229	0.638888	0.0	0.0	0.0	0.0	-0.004712	0.0	5.51566
58	0.982229	0.750000	0.0	0.0	0.0	0.0	-0.016560	0.0	5.51557
59	0.982229	0.861112	0.0	0.0	0.0	0.0	-0.023782	0.0	5.51566
60	0.982229	0.972222	0.0	0.0	0.0	0.0	-0.026783	0.0	5.51568

SURFACE INTERPOLATION COEFFICIENTS B(I,J)

REGION 1
SURFACE SPLINE
COEFFICIENTS

1	0.0	0.0	0.0	0.0	0.08989	0.0	-1.91339
2	0.0	0.0	0.0	0.0	-0.10763	0.0	5.75452
3	0.0	0.0	0.0	0.0	-0.09335	0.0	-0.00001
4	0.0	0.0	0.0	0.0	-0.14201	0.0	-1.05979
5	0.0	0.0	0.0	0.0	0.44922	0.0	-0.39765
6	0.0	0.0	0.0	0.0	-0.15032	0.0	-0.42768
7	0.0	0.0	0.0	0.0	0.05519	0.0	-0.41352
8	0.0	0.0	0.0	0.0	-0.04803	0.0	-0.29371
9	0.0	0.0	0.0	0.0	0.01917	0.0	-0.29375
10	0.0	0.0	0.0	0.0	-0.04768	0.0	-0.41353
11	0.0	0.0	0.0	0.0	-0.02355	0.0	-0.42765
12	0.0	0.0	0.0	0.0	-0.04697	0.0	-0.39704
13	0.0	0.0	0.0	0.0	0.01546	0.0	-1.06988
14	0.0	0.0	0.0	0.0	-0.00714	0.0	0.85551
15	0.0	0.0	0.0	0.0	-0.19104	0.0	0.42434
16	0.0	0.0	0.0	0.0	0.00619	0.0	0.47749
17	0.0	0.0	0.0	0.0	-0.00244	0.0	0.44284
18	0.0	0.0	0.0	0.0	0.00555	0.0	0.32039
19	0.0	0.0	0.0	0.0	-0.00088	0.0	0.37027
20	0.0	0.0	0.0	0.0	0.02014	0.0	0.44276
21	0.0	0.0	0.0	0.0	0.02330	0.0	0.47776
22	0.0	0.0	0.0	0.0	0.00355	0.0	0.42385
23	0.0	0.0	0.0	0.0	0.02222	0.0	0.85572
24	0.0	0.0	0.0	0.0	-0.02552	0.0	0.36661
25	0.0	0.0	0.0	0.0	0.00741	0.0	-0.29097
26	0.0	0.0	0.0	0.0	0.00512	0.0	0.00335
27	0.0	0.0	0.0	0.0	-0.00165	0.0	-0.02671
28	0.0	0.0	0.0	0.0	-0.00365	0.0	-0.02195
29	0.0	0.0	0.0	0.0	-0.00595	0.0	-0.02224
30	0.0	0.0	0.0	0.0	-0.00736	0.0	-0.02649
31	0.0	0.0	0.0	0.0	-0.00072	0.0	0.00320
32	0.0	0.0	0.0	0.0	-0.02105	0.0	-0.29087
33	0.0	0.0	0.0	0.0	0.01493	0.0	0.36664
34	0.0	0.0	0.0	0.0	-0.05483	0.0	0.40258
35	0.0	0.0	0.0	0.0	0.06570	0.0	-0.40604
36	0.0	0.0	0.0	0.0	-0.00849	0.0	-0.01268
37	0.0	0.0	0.0	0.0	-0.00478	0.0	-0.05827
38	0.0	0.0	0.0	0.0	-0.00530	0.0	-0.03415
39	0.0	0.0	0.0	0.0	-0.00611	0.0	-0.03410
40	0.0	0.0	0.0	0.0	-0.00901	0.0	-0.05843
41	0.0	0.0	0.0	0.0	-0.00772	0.0	-0.01247
42	0.0	0.0	0.0	0.0	-0.02575	0.0	-0.40626
43	0.0	0.0	0.0	0.0	0.03330	0.0	0.40263
44	0.0	0.0	0.0	0.0	-0.04377	0.0	0.73903
45	0.0	0.0	0.0	0.0	0.06675	0.0	-0.16597
46	0.0	0.0	0.0	0.0	-0.00207	0.0	0.20148
47	0.0	0.0	0.0	0.0	-0.00685	0.0	0.16008
48	0.0	0.0	0.0	0.0	-0.00954	0.0	0.11499
49	0.0	0.0	0.0	0.0	-0.00933	0.0	0.11511
50	0.0	0.0	0.0	0.0	-0.03677	0.0	0.16016
51	0.0	0.0	0.0	0.0	0.04321	0.0	0.20120
52	0.0	0.0	0.0	0.0	-0.00728	0.0	-0.16567
53	0.0	0.0	0.0	0.0	0.01154	0.0	0.78889

54	0.0	0.0	0.0	0.0	-0.04951	0.0	-0.56053
55	0.0	0.0	0.0	0.0	0.06524	0.0	-0.02956
56	0.0	0.0	0.0	0.0	-0.00110	0.0	-0.14229
57	0.0	0.0	0.0	0.0	-0.00926	0.0	-0.12043
58	0.0	0.0	0.0	0.0	-0.02824	0.0	-0.08700
59	0.0	0.0	0.0	0.0	-0.01080	0.0	-0.08738
60	0.0	0.0	0.0	0.0	0.00233	0.0	-0.12034
61	0.0	0.0	0.0	0.0	-0.05347	0.0	-0.14219
62	0.0	0.0	0.0	0.0	-0.03895	0.0	-0.02963
63	0.0	0.0	0.0	0.0	0.03940	0.0	-0.56051
	0.0	0.0	0.0	0.0	0.06181	0.0	2.73875

```
INORM,ISURF,IX,IY,IZ =    4    0    4    6    5
U1,UL,V1,VL,CR,CT,CHX =          0.0          -48.00000          0.0          3.14159          -48.00000          0.0          0.0
N2 =    60      N3.=    63
XO =          0.0          YO =          0.0          ZO =          0.0
TRANSFORMATION MATRIX,  APIJ, I=1,3, J=1,3
            1.000000          0.0          0.0
            0.0          1.000000          0.0
            0.0          0.0          1.000000
THE NUMBER OF ELEMENTS IN COMPONENT =   342
THE NUMBER OF ELEMENTS INTERPOLATED =   179
```

DIRECTORY INFORMATION FOR REGION 1

SURFACE SPLINE

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

FIRST PAGE OF FORCE DATA FOR COMPONENT 1 IN REGION 1

AIR

ELEMENT DATA MACH= 4.500 ALT = 0. S REF = 102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
XCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 4.00 YAW ANGLE = 0.0 K = 0.0 ETAC = 0.0 DELTA E = 0.0
IDERIV = 0 Q = 0.0 R = 0.0 P = 0.0

L	DEL CA / CA / XCENT	DEL CY / CY / YCENT	DEL CN / CN / ZCENT	DEL CLL / CLL	DEL CLM / CLM	DEL CLN / CLN	CP / DELTA	AREA
1	0.88615E-05	0.33949E-05	0.44351E-04	-0.40135E-10	-0.35130E-06	-0.89974E-07	0.20051E+00	0.23224E-01
	0.88615E-05	0.33949E-05	0.44351E-04	-0.40135E-10	-0.35130E-06	-0.89974E-07	0.17233E+02	
	-0.76190E+00	0.13236E-01	-0.15107E+00					
2	0.97973E-05	0.11395E-04	0.42700E-04	0.20426E-09	-0.33822E-06	-0.26330E-06	0.19985E+00	0.23131E-01
	0.17650E-04	0.15200E-04	0.87051E-04	0.16412E-09	-0.68952E-06	-0.35327E-06	0.17050E+02	
	-0.76190E+00	0.39258E-01	-0.14640E+00					
3	0.86702E-05	0.18440E-04	0.39461E-04	-0.17627E-09	-0.31257E-06	-0.42598E-06	0.19619E+00	0.23222E-01
	0.26329E-04	0.33730E-04	0.12651E-03	-0.12147E-10	-0.10021E-05	-0.77925E-06	0.16680E+02	
	-0.76190E+00	0.64079E-01	-0.13746E+00					
4	0.80496E-05	0.23189E-04	0.33127E-04	0.32401E-10	-0.26239E-06	-0.53573E-06	0.18230E+00	0.23200E-01
	0.34378E-04	0.55919E-04	0.15964E-03	0.20254E-10	-0.12645E-05	-0.13150E-05	0.16149E+02	
	-0.76190E+00	0.85999E-01	-0.12422E+00					
5	0.72916E-05	0.25867E-04	0.25867E-04	-0.56843E-13	-0.20489E-06	-0.59760E-06	0.16513E+00	0.23171E-01
	0.41659E-04	0.82796E-04	0.18551E-03	0.20197E-10	-0.14694E-05	-0.19126E-05	0.15464E+02	
	-0.76190E+00	0.10724E+00	-0.10724E+00					
6	0.64021E-05	0.27173E-04	0.19021E-04	-0.26716E-10	-0.15067E-06	-0.62777E-06	0.14954E+00	0.23200E-01
	0.48261E-04	0.13936E-03	0.20453E-03	-0.65192E-11	-0.16200E-05	-0.25404E-05	0.14651E+02	
	-0.76190E+00	0.12422E+00	-0.86980E-01					
7	0.59200E-05	0.26944E-04	0.12591E-04	0.12033E-09	-0.99723E-07	-0.62249E-06	0.13396E+00	0.23221E-01
	0.54191E-04	0.13690E-03	0.21712E-03	0.11381E-09	-0.17198E-05	-0.31628E-05	0.13742E+02	
	-0.76190E+00	0.13746E+00	-0.64079E-01					
8	0.51626E-05	0.25058E-04	0.66370E-05	-0.11988E-09	-0.52975E-07	-0.57830E-06	0.11728E+00	0.23131E-01
	0.59344E-04	0.16196E-03	0.22381E-03	-0.60680E-11	-0.17727E-05	-0.37417E-05	0.12744E+02	
	-0.76190E+00	0.14640E+00	-0.39258E-01					
9	0.44542E-05	0.22233E-04	0.19578E-05	0.20172E-10	-0.15506E-07	-0.51503E-06	0.10079E+00	0.23224E-01
	0.63798E-04	0.19425E-03	0.22576E-03	0.14104E-10	-0.17983E-05	-0.42568E-05	0.11719E+02	
	-0.76190E+00	0.15107E+00	-0.13236E-01					

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

LAST PAGE OF FORCE DATA FOR COMPONENT 1 IN REGION 1

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

AIR

ELEMENT DATA MACH= 4.500 ALT = 0. S REF = 102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
XCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 6.00 YAW ANGLE = 0.0 K = 0.0 ETAC = 0.0 DELTA F = 0.0
IDERIV = 0 O = 0.0 R = 0.0 P = 0.0

I	DEL CA / CA / XCENT	DEL CY / CY / YCENT	DEL CN / CN / ZCENT	DEL CLL / CLL	DEL CLM / CLM	DEL CLN / CLN	CP / DELTA	AREA
173	0.90514E-05	0.17692E-03	-0.47399E-04	0.75397E-09	0.20451E-04	-0.22263E-03	0.35282E-02	0.53317E+01
	0.22470E-01	0.11399E+00	0.12018E+00	0.16226E-07	-0.31659E-01	-0.79314E-01	0.12646E+01	
	-0.42882E+02	0.51459E+01	0.13788E+01					
174	-0.27435E-05	-0.50317E-04	0.23467E-04	0.48658E-09	-0.10125E-04	0.63320E-04	-0.10694E-02	0.53322E+01
	0.22367E-01	0.11334E+00	0.12020E+00	0.16713E-07	-0.31669E-01	-0.79250E-01	0.28403E+00	
	-0.42882E+02	0.49232E+01	0.22514E+01					
175	-0.15383E-04	-0.25502E-03	0.17856E-03	-0.16735E-08	-0.77039E-04	0.32092E-03	-0.59967E-02	0.53320E+01
	0.22452E-01	0.11359E+00	0.12039E+00	0.15039E-07	-0.31746E-01	-0.78029E-01	-0.61876E+00	
	-0.42882E+02	0.43639E+01	0.30557E+01					
176	-0.27517E-04	-0.39374E-03	0.39376E-03	0.14552E-13	-0.16989E-03	0.49551E-03	-0.10727E-01	0.53316E+01
	0.22424E-01	0.11319E+00	0.12077E+00	0.15054E-07	-0.31916E-01	-0.78434E-01	-0.14177E+01	
	-0.42882E+02	0.37670E+01	0.37670E+01					
177	-0.37327E-04	-0.43327E-03	0.61882E-03	0.41182E-08	-0.26699E-03	0.54524E-03	-0.14551E-01	0.53320E+01
	0.22387E-01	0.11276E+00	0.12139E+00	0.19172E-07	-0.32183E-01	-0.77888E-01	-0.20885E+01	
	-0.42882E+02	0.30557E+01	0.43639E+01					
178	-0.44400E-04	-0.37978E-03	0.81432E-03	-0.78289E-09	-0.35134E-03	0.47792E-03	-0.17307E-01	0.53322E+01
	0.22343E-01	0.11238E+00	0.12221E+00	0.11343E-07	-0.32534E-01	-0.77411E-01	-0.26098E+01	
	-0.42882E+02	0.22514E+01	0.48282E+01					
179	-0.49386E-04	-0.25862E-03	0.96529E-03	0.41327E-09	-0.41648E-03	0.32545E-03	-0.19251E-01	0.53317E+01
	0.22293E-01	0.11212E+00	0.12317E+00	0.15476E-07	-0.32951E-01	-0.77085E-01	-0.29668E+01	
	-0.42882E+02	0.13798E+01	0.51459E+01					
180	-0.53934E-04	-0.95106E-04	0.10872E-02	0.17935E-09	-0.46908E-03	0.11968E-03	-0.21022E-01	0.53319E+01
	0.22239E-01	0.11203E+00	0.12426E+00	0.17269E-07	-0.33420E-01	-0.76965E-01	-0.31490E+01	
	-0.42882E+02	0.44431E+00	0.53071E+01					

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

SUMMARY PAGE
COMPONENT 1
IN REGION 1

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

COMPONENT NUMBER = 1

PANEL ID = BODY

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA / BETA	C D / L/D	C L / C M	C A / C LL	C Y / C LN	C N
6.00	0.03511	0.12125	0.02224	0.11203	0.12426
0.0	3.45390	-0.03342	0.00000	-0.07697	

DATA SAVED ON UNIT 9, SET NUMBER = 1

```
INORM,ISUPF,IX,IY,IZ =    4    0    4    6    5
U1,UL,V1,VL,CP,CT,CHX =        -48.00000        -52.00000        0.0        3.14159        -52.00000        0.0        0.0
N2 =    60      N3 =    63
XO =        0.0        YO =        0.0        ZO =        0.0
TRANSFORMATION MATRIX,  APIJ,  I=1,3,  J=1,3
            1.000000        0.0        0.0
            0.0        1.000000        0.0
            0.0        0.0        1.000000
THE NUMBER OF ELEMENTS IN COMPONENT =    342
THE NUMBER OF ELEMENTS INTERPOLATED =    162
```

DIRECTORY
INFORMATION
FOR REGION 2
SURFACE SPLINE

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

FIRST PAGE OF FORCE DATA FOR COMPONENT 2 IN REGION 2

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

AIR

ELEMENT DATA MACH= 4.500 ALT = 0. S REF = 102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
YCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 6.00 YAW ANGLE = 0.0 K = 0.0 FTAC = 0.0 DELTA E = 0.0
IDERIV = 0 0 = 0.0 R = 0.0 P = 0.0

L	DEL CA / CA / XCENT	DEL CY / CY / YCENT	DEL CN / CN / ZCENT	DEL CLL / CLL	DEL CLM / CLM	DEL CLN / CLN	CP / DELTA	AREA
181	0.77151E-04	0.22857E-03	0.26130E-02	0.58062E-08	-0.12738E-02	-0.32499E-03	0.48476E-01	0.55530E+01
	0.77151E-04	0.22857E-03	0.26130E-02	0.58062E-08	-0.12738E-02	-0.32499E-03	0.76618E+01	
	-0.49586E+02	0.49317E+00	-0.55307E+01					
182	0.73205E-04	0.64426E-03	0.24044E-02	-0.27212E-08	-0.11721E-02	-0.91604E-03	0.46004E-01	0.55529E+01
	0.15036E-03	0.97203E-03	0.50174E-02	0.30850E-09	-0.24459E-02	-0.12410E-02	0.74788E+01	
	-0.49586E+02	0.14360E+01	-0.53626E+01					
183	0.67758E-04	0.97377E-03	0.20880E-02	-0.17710E-07	-0.10179E-02	-0.13846E-02	0.42577E-01	0.55533E+01
	0.21811E-03	0.18466E-02	0.71054E-02	-0.14625E-07	-0.34638E-02	-0.26256E-02	0.71189E+01	
	-0.49586E+02	0.23463E+01	-0.50316E+01					
184	0.59675E-04	0.11637E-02	0.16620E-02	0.10856E-07	-0.81019E-03	-0.16545E-02	0.37495E-01	0.55532E+01
	0.27779E-03	0.30103E-02	0.87673E-02	-0.37689E-09	-0.42740E-02	-0.42801E-02	0.65938E+01	
	-0.48586E+02	0.31344E+01	-0.45477E+01					
185	0.49918E-04	0.12001E-02	0.12001E-02	0.0	-0.58505E-03	-0.17064E-02	0.31370E-01	0.55524E+01
	0.32771E-03	0.42104E-02	0.99674E-02	-0.37689E-09	-0.48590E-02	-0.59865E-02	0.59188E+01	
	-0.48586E+02	0.39257E+01	-0.39257E+01					
186	0.39358E-04	0.10961E-02	0.76749E-03	-0.71450E-09	-0.37414E-03	-0.15585E-02	0.24729E-01	0.55532E+01
	0.35706E-03	0.53065E-02	0.10735E-01	-0.10714E-07	-0.52332E-02	-0.75451E-02	0.51156E+01	
	-0.48586E+02	0.45477E+01	-0.31844E+01					
187	0.28353E-04	0.39912E-03	0.41465E-03	0.75524E-09	-0.20214E-03	-0.12642E-02	0.18130E-01	0.55533E+01
	0.39592E-03	0.61956E-02	0.11150E-01	-0.33615E-08	-0.54353E-02	-0.88092E-02	0.42091E+01	
	-0.48586E+02	0.50316E+01	-0.23463E+01					
188	0.19229E-04	0.63152E-03	0.16022E-03	0.72760E-09	-0.82492E-04	-0.89792E-03	0.12083E-01	0.55529E+01
	0.41515E-03	0.68271E-02	0.11319E-01	-0.26339E-09	-0.55178E-02	-0.97072E-02	0.32262E+01	
	-0.48586E+02	0.51626E+01	-0.14369E+01					
189	0.11021E-04	0.37327E-03	0.32652E-04	-0.82946E-09	-0.15917E-04	-0.53073E-03	0.69249E-02	0.55530E+01
	0.42617E-03	0.72004E-02	0.11351E-01	-0.34634E-09	-0.55337E-02	-0.10238E-01	0.21975E+01	
	-0.48586E+02	0.55337E+01	-0.48387E+00					

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

LAST PAGE OF FORCE DATA FOR COMPONENT 2 IN REGION 2

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

AIR

ELEMENT DATA MACH= 4.500 ALT = 0. S REF = 102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
XCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 6.00 YAW ANGLE = 0.0 K = 0.0 ETAC = 0.0 DELTA E = 0.0
IDERIV = 0 Q = 0.0 P = 0.0 P = 0.0

L	DEL CA CA XCENT	DEL CY CY YCENT	DEL CN CN ZCENT	DEL CLL CLL	DEL CLM CLM	DEL CLN CLN	CP DELTA	AREA
334	0.18139E-03 0.97254E-02 -0.97012E+02	-0.12960E-02 -0.76491E-01 0.29099E+01	0.11351E-03 0.11265E+00 0.25460E+00	0.10186E-07 -0.10113E-07	-0.10966E-03 -0.73824E-01	0.36517E-02 0.18178E+00	-0.45978E-01 -0.84166E+01	0.29307E+01
335	0.19095E-03 0.99154E-02 -0.97012E+02	-0.13166E-02 -0.77908E-01 0.23215E+01	0.35274E-03 0.11300E+00 0.75604E+00	-0.45839E-08 -0.14697E-07	-0.34077E-03 -0.74165E-01	0.37098E-02 0.18549E+00	-0.48178E-01 -0.94466E+01	0.29304E+01
336	0.19336E-03 0.10109E-01 -0.97012E+02	-0.13602E-02 -0.79058E-01 0.26474E+01	0.58754E-03 0.11359E+00 0.12345E+01	-0.86875E-09 -0.23384E-07	-0.56759E-03 -0.74732E-01	0.35509E-02 0.18904E+00	-0.49145E-01 -0.10433E+02	0.29305E+01
337	0.19495E-03 0.10304E-01 -0.97012E+02	-0.11453E-02 -0.82213E-01 0.23929E+01	0.80195E-03 0.11439E+00 0.16755E+01	-0.12369E-08 -0.24621E-07	-0.77472E-03 -0.75507E-01	0.32271E-02 0.19227E+00	-0.49415E-01 -0.11345E+02	0.29306E+01
338	0.19453E-03 0.10499E-01 -0.97012E+02	-0.98644E-03 -0.81230E-01 0.27655E+01	0.98644E-03 0.11538E+00 0.20455E+01	-0.29104E-10 -0.24650E-07	-0.95295E-03 -0.76460E-01	0.27794E-02 0.19505E+00	-0.49304E-01 -0.12155E+02	0.29306E+01
339	0.19430E-03 0.10693E-01 -0.97012E+02	-0.79927E-03 -0.81099E-01 0.16755E+01	0.11415E-02 0.11652E+00 0.23928E+01	0.12078E-08 -0.23443E-07	-0.11027E-02 -0.77562E-01	0.22521E-02 0.19730E+00	-0.49250E-01 -0.12835E+02	0.29306E+01
340	0.19407E-03 0.10837E-01 -0.97012E+02	-0.59816E-03 -0.83597E-01 0.13345E+01	0.12616E-02 0.11778E+00 0.26474E+01	0.86075E-08 -0.14755E-07	-0.12187E-02 -0.78781E-01	0.16572E-02 0.19896E+00	-0.49197E-01 -0.13366E+02	0.29305E+01
341	0.19364E-03 0.11091E-01 -0.97012E+02	-0.35942E-03 -0.82046E-01 0.75604E+00	0.13415E-02 0.11913E+00 0.28215E+01	0.46857E-09 -0.10069E-07	-0.12960E-02 -0.80077E-01	0.10127E-02 0.19997E+00	-0.49089E-01 -0.13729E+02	0.29304E+01
342	0.19403E-03 0.11275E-01 -0.97012E+02	-0.12145E-03 -0.83058E-01 0.25460E+00	0.13967E-02 0.12051E+00 0.29099E+01	-0.10893E-07 -0.20962E-07	-0.13396E-02 -0.81417E-01	0.34220E-03 0.20031E+00	-0.49194E-01 -0.13914E+02	0.29307E+01

```
SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0
CASE     **                                                          PAGE    43
CASE 101A  SAMPLE PROBLEM USING INPUT PRESSURES
COMPONENT NUMBER =  2
PANEL ID =  BODY
  MACH=    4.500  VEL=    5024.0 FT/SEC  RE/FT =  0.31945E+08
  ALT =      0.
    *AIR*
  S REF =    102.58  SPAN =    34.29  MAC  =  100.00
  X CG  =      0.0   Y CG =     0.0  .Z CG =     0.0
FORCE DATA
  ALPHA     C D        C L        C A        C Y        C N
  BETA      L/D        C M        C LL       C LN

   6.00   0.02381   0.11867   0.01127  -0.08307   0.12051
   0.0    4.98417  -0.08142  -0.00000   0.20031
          DATA SAVED ON UNIT 9.  SET NUMBER =   2
```

SUMMARY PAGE
FOR COMPONENT 2
IN REGION 2

```
INORM,ISURF,IX,IY,IZ =    1    1    1    2    3
U1,UL,V1,VL,CR,CT,CHX =        -47.62939        -76.60291        4.94940        12.19350        -21.33571        0.0        -6.57892
N2 =    15        N3 =    18
X0 =        0.0        Y0 =        0.0        Z0 =        0.0
TRANSFORMATION MATRIX,  APIJ, I=1,3, J=1,3
            1.000000        0.0        0.0
            0.0        1.000000        0.0
            0.0        0.0        1.000000
THE NUMBER OF ELEMENTS IN COMPONENT =    78
THE NUMBER OF ELEMENTS INTERPOLATED =    35
```

DIRECTORY
INFORMATION
FOR REGION 3
SURFACE SPLINE

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE ** PAGE 44

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

AIR

FIRST PAGE OF FORCE DATA FOR COMPONENT 3 IN REGION 3

ELEMENT DATA MACH= 4.500 ALT = 0. S REF = 102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
XCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 6.00 YAW ANGLE = 0.0 K = 0.0 ETAC = 0.0 DELTA E = 0.0
IDERIV = 0 Q = 0.0 R = 0.0 P = 0.0

L	DEL CA CA XCENT	DEL CY CY YCENT	DEL CN CN ZCENT	DEL CLL CLL	DEL CLM CLM	DEL CLN CLN	CP DELTA	AREA
1	-0.18856E-03 -0.19956E-03 -0.52585E+02	-0.47139E-03 -0.47139E-03 0.67546E+01	0.18856E-02 0.19956E-02 0.19474E+00	0.36890E-03 0.36890E-03	-0.99188E-03 -0.99188E-03	0.76014E-03 0.76014E-03	-0.23078E-01 -0.28091E+00	0.86799E+01
2	-0.20340E-03 -0.39196E-03 -0.56489E+02	-0.50850E-03 -0.97999E-03 0.67566E+01	0.20340E-02 0.39196E-02 0.58423E+00	0.39217E-03 0.76107E-03	-0.11500E-02 -0.21419E-02	0.87775E-03 0.16379E-02	-0.24895E-01 -0.28091E+00	0.86799E+01
3	-0.20960E-03 -0.60156E-03 -0.60374E+02	-0.52401E-03 -0.15039E-02 0.67566E+01	0.20960E-02 0.60156E-02 0.97371E+00	0.39818E-03 0.11593E-02	-0.12675E-02 -0.34094E-02	0.96404E-03 0.26019E-02	-0.25654E-01 -0.28091E+00	0.86799E+01
4	-0.20851E-03 -0.81007E-03 -0.64269E+02	-0.52127E-03 -0.20252E-02 0.67566E+01	0.20851E-02 0.81007E-02 0.13632E+01	0.39017E-03 0.15494E-02	-0.13429E-02 -0.47523E-02	0.10182E-02 0.36201E-02	-0.25520E-01 -0.28091E+00	0.86799E+01
5	-0.16213E-03 -0.97220E-03 -0.68164E+02	-0.40533E-03 -0.24305E-02 0.67566E+01	0.16213E-02 0.97220E-02 0.17527E+01	0.29879E-03 0.18482E-02	-0.11080E-02 -0.58603E-02	0.83780E-03 0.44579E-02	-0.19844E-01 -0.28091E+00	0.86799E+01
12	-0.52951E-04 -0.10652E-02 -0.57055E+02	-0.23240E-03 -0.26620E-02 0.86895E+01	0.92960E-03 0.10652E-01 0.15349E+00	0.23453E-03 0.20827E-02	-0.53052E-03 -0.63908E-02	0.41029E-03 0.48682E-02	-0.17888E-01 -0.28140E+00	0.55208E+01
13	-0.10290E-03 -0.11679E-02 -0.60225E+02	-0.25700E-03 -0.29190E-02 0.86895E+01	0.10279E-02 0.11679E-01 0.47549E+00	0.25694E-03 0.23397E-02	-0.61951E-03 -0.70103E-02	0.47749E-03 0.53457E-02	-0.19779E-01 -0.28042E+00	0.55208E+01
14	-0.11075E-03 -0.12787E-02 -0.63395E+02	-0.27690E-03 -0.31958E-02 0.86895E+01	0.11076E-02 0.12787E-01 0.79249E+00	0.27431E-03 0.26140E-02	-0.70303E-03 -0.77133E-02	0.54006E-03 0.58858E-02	-0.21313E-01 -0.28140E+00	0.55208E+01
15	-0.11580E-03 -0.13946E-02 -0.66565E+02	-0.29975E-03 -0.34955E-02 0.85895E+01	0.11580E-02 0.13946E-01 0.11095E+01	0.28433E-03 0.28983E-02	-0.77258E-03 -0.84860E-02	0.59191E-03 0.64777E-02	-0.22300E-01 -0.28042E+00	0.55208E+01

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

AIR

LAST PAGE OF FORCE DATA FOR COMPONENT 3 IN REGION 3

ELEMENT DATA MACH= 4.500 ALT = 0. S REF = 102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
XCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 4.00 YAW ANGLE = 0.0 K = 0.0 ETAC = 0.0 DELTA E = 0.0
INDEPIV = 0 Q = 0.0 R = 0.0 P = 0.0

L	DEL CA / CA / XCENT	DEL CY / CY / YCENT	DEL CN / CN / ZCENT	DEL CLL / CLL	DEL CLM / CLM	DEL CLN / CLN	CP / DELTA	AREA
58	0.15449E-06 -0.20443E-02 -0.73578E+02	0.39675E-06 -0.51108E-02 0.15463E+02	-0.15470E-05 0.20443E-01 0.15747E+00	-0.69595E-04 0.49047E-02	0.11447E-05 -0.12895E-01	-0.90426E-06 0.98928E-02	0.13330E-03 -0.28157E+00	0.12329E+01
59	0.55855E-05 -0.20387E-02 -0.74608E+02	0.13966E-04 -0.50966E-02 0.15463E+02	-0.55863E-04 0.20387E-01 0.22046E+00	-0.25105E-04 0.48796E-02	0.41691E-04 -0.12853E-01	-0.32910E-04 0.98599E-02	0.48134E-02 -0.28162E+00	0.12329E+01
60	0.63139E-05 -0.20324E-02 -0.75239E+02	0.15787E-04 -0.50810E-02 0.15463E+02	-0.63145E-04 0.20324E-01 0.28345E+00	-0.28348E-04 0.48512E-02	0.47527E-04 -0.12805E-01	-0.37491E-04 0.99224E-02	0.54410E-02 -0.28146E+00	0.12329E+01
69	-0.12410E-04 -0.20448E-02 -0.49739E+02	-0.32175E-04 -0.51132E-02 0.56538E+01	0.12363E-03 0.20448E-01 0.18633E+00	0.20213E-04 0.48714E-02	-0.61517E-04 -0.12867E-01	0.48724E-04 0.98711E-02	-0.25775E-01 -0.25955E+00	0.51083E+00
69	-0.80466E-05 -0.20537E-02 -0.54494E+02	-0.20489E-04 -0.51337E-02 0.56746E+01	0.90240E-04 0.20538E-01 0.65425E+00	0.14545E-04 0.48860E-02	-0.49225E-04 -0.12916E-01	0.34039E-04 0.99051E-02	-0.27268E-01 -0.32982E+00	0.34974E+00
70	-0.15729E-04 -0.20694E-02 -0.58767E+02	-0.32902E-04 -0.51645E-02 0.56417E+01	0.15659E-03 0.20695E-01 0.10887E+01	0.24725E-04 0.49107E-02	-0.92194E-04 -0.13008E-01	0.58811E-04 0.99639E-02	-0.27692E-01 -0.25856E+00	0.59553E+00
71	-0.35583E-04 -0.21050E-02 -0.62964E+02	-0.86085E-04 -0.52526E-02 0.55535E+01	0.35229E-03 0.21047E-01 0.15312E+01	0.53269E-04 0.49640E-02	-0.22236E-03 -0.13231E-01	0.16386E-03 0.10128E-01	-0.26384E-01 -0.22578E+00	0.14168E+01
72	-0.52407E-04 -0.21574E-02 -0.67311E+02	-0.13158E-03 -0.53841E-02 0.54128E+01	0.52586E-03 0.21573E-01 0.20034E+01	0.75331E-04 0.50393E-02	-0.35502E-03 -0.13586E-01	0.26660E-03 0.10394E-01	-0.21237E-01 -0.22965E+00	0.26307E+01

```
SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0
CASE     **                                                          PAGE    48
CASE 101A  SAMPLE PROBLEM USING INPUT PRESSURES
COMPONENT NUMBER =  3
PANEL ID =  WNGU    WBU
  MACH=    4.500  VEL=    5024.0 FT/SEC  RE/FT =  0.31945E+08
  ALT =      0.
    *AIR*
  S REF =    102.59  SPAN =   34.29  MAC  =  100.00
  X CG  =      0.0   Y CG =     0.0   Z CG =    0.0
FORCE DATA
  ALPHA     C D        C L        C A        C Y        C N
  BETA      L/D        C M        C LL       C LN
   6.00   0.00011   0.02168  -0.00216  -0.00538   0.02157
   0.0  198.27116  -0.01359   0.00504   0.01039
          DATA SAVED ON UNIT 9.  SET NUMBER =   3
```

SUMMARY PAGE
FOR COMPONENT 3
IN REGION 3

```
INORM,ISURF,IX,IY,IZ =    1    1    1    2    3
U1,UL,V1,VL,CP,CT,CHX =        -68.96510       -76.60291        3.37250        13.77040       -24.85080        0.0            -6.7853
N2 =    15      N3 =    18
XO =          0.0          YO =          0.0          ZO =          0.0
TRANSFORMATION MATRIX,  APIJ, I=1,3, J=1,3
              1.000000              0.0                  0.0
              0.0                   1.000000             0.0
              0.0                   0.0                  1.000000
THE NUMBER OF ELEMENTS IN COMPONENT =     78
THE NUMBER OF ELEMENTS INTERPOLATED =     41
```

DIRECTORY
INFORMATION
FOR REGION 4
SURFACE SPLINE

UPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

ASE **

FIRST PAGE OF FORCE DATA FOR COMPONENT 4 IN REGION 4

ASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

AIR

LEMENT DATA MACH= 4.500 ALT = 0. S REF = 102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
XCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 6.00 YAW ANGLE = 0.0 K = 0.0 ETAC = 0.0 DELTA E = 0.0
IDEPIV = 0 Q = 0.0 R = 0.0 P = 0.0

L	DEL CA / CA / XCENT	DEL CY / CY / YCENT	DEL CN / CN / ZCENT	DEL CLL / CLL	DEL CLM / CLM	DEL CLN / CLN	CP / DELTA	AREA
6	0.49456E-03	-0.60570E-03	0.48456E-02	0.92395E-03	-0.34832E-02	0.11775E-02	-0.59307E-01	0.84881E+01
	0.49456E-03	-0.60570E-03	0.48456E-02	0.92395E-03	-0.34832E-02	0.11775E-02	-0.11620E+02	
	-0.72059E+02	0.67566E+01	0.17527E+01					
7	0.49080E-03	-0.61350E-03	0.49090E-02	0.94282E-03	-0.37211E-02	0.12624E-02	-0.60071E-01	0.84881E+01
	0.97537E-03	-0.12192E-02	0.97536E-02	0.18668E-02	-0.72043E-02	0.24399E-02	-0.11620E+02	
	-0.75954E+02	0.67566E+01	0.13632E+01					
8	0.49628E-03	-0.62035E-03	0.49628E-02	0.96039E-03	-0.39579E-02	0.13470E-02	-0.60742E-01	0.84881E+01
	0.14716E-02	-0.19326E-02	0.14716E-01	0.28272E-02	-0.11162E-01	0.37869E-02	-0.11620E+02	
	-0.79869E+02	0.67566E+01	0.97371E+00					
9	0.50110E-03	-0.62637E-03	0.50110E-02	0.97683E-03	-0.41934E-02	0.14312E-02	-0.61331E-01	0.84881E+01
	0.19727E-02	-0.24659E-02	0.19727E-01	0.38040E-02	-0.15356E-01	0.52180E-02	-0.11620E+02	
	-0.83743E+02	0.67566E+01	0.58423E+00					
10	0.50531E-03	-0.63163E-03	0.50531E-02	0.99220E-03	-0.44274E-02	0.15150E-02	-0.61846E-01	0.84881E+01
	0.24781E-02	-0.30976E-02	0.24781E-01	0.47962E-02	-0.19783E-01	0.67330E-02	-0.11620E+02	
	-0.87638E+02	0.67566E+01	0.19474E+00					
11	0.0	0.0	0.0	0.0	0.0	0.0	-0.62043E-01	0.0
	0.24781E-02	-0.30976E-02	0.24781E-01	0.47962E-02	-0.19783E-01	0.67330E-02	0.0	
	-0.89539E+02	0.67941E+01	0.0					
17	0.30253E-03	-0.37826E-03	0.30256E-02	0.75108E-03	-0.22015E-02	0.72765E-03	-0.58222E-01	0.53988E+01
	0.27806E-02	-0.34758E-02	0.27806E-01	0.55473E-02	-0.21985E-01	0.74606E-02	-0.11619E+02	
	-0.72905E+02	0.86895E+01	0.14265E+01					
18	0.30623E-03	-0.33230E-03	0.30620E-02	0.76366E-03	-0.23260E-02	0.77176E-03	-0.58922E-01	0.53988E+01
	0.30868E-02	-0.38596E-02	0.30869E-01	0.63109E-02	-0.24311E-01	0.82324E-02	-0.11620E+02	
	-0.76075E+02	0.86895E+01	0.11095E+01					
19	0.30940E-03	-0.38682E-03	0.30943E-02	0.77528E-03	-0.24495E-02	0.81565E-03	-0.59543E-01	0.53988E+01
	0.33962E-02	-0.42454E-02	0.33962E-01	0.70862E-02	-0.26760E-01	0.90480E-02	-0.11619E+02	
	-0.79245E+02	0.86895E+01	0.79249E+00					

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

LAST PAGE OF FORCE DATA FOR COMPONENT 4 IN REGION 4

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

AIR

ELEMENT DATA MACH= 4.500 ALT = 0. S REF = 102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
XCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 6.00 YAW ANGLE = 0.0 K = 0.0 ETAC = 0.0 DELTA E = 0.0
IDERIV = 0 O = 0.0 R = 0.0 P = 0.0

L	DEL CA / CA / XCENT	DEL CY / CY / YCENT	DEL CN / CN / ZCENT	DEL CLL / CLL	DEL CLM / CLM	DEL CLN / CLN	CP / DELTA	AREA
74	0.24105E-03 0.74445E-02 -0.75942E+02	-0.23372E-03 -0.93891E-02 0.52514E+01	0.24152E-02 0.74452E-01 0.15509E+01	0.35664E-03 0.19722E-01	-0.18304E-02 -0.58577E-01	0.61366E-03 0.19345E-01	-0.60999E-01 -0.11614E+02	0.41116E+01
75	0.30392E-03 0.77484E-02 -0.80451E+02	-0.33102E-03 -0.95691E-02 0.51381E+01	0.30556E-02 0.77508E-01 0.11157E+01	0.44552E-03 0.20167E-01	-0.24549E-02 -0.61032E-01	0.84850E-03 0.20194E-01	-0.61761E-01 -0.11590E+02	0.51394E+01
76	0.41533E-03 0.81637E-02 -0.85125E+02	-0.52662E-03 -0.10196E-01 0.49479E+01	0.41534E-02 0.81661E-01 0.67316E+00	0.58905E-03 0.20756E-01	-0.35330E-02 -0.64565E-01	0.12476E-02 0.21441E-01	-0.62508E-01 -0.11617E+02	0.69046E+01
77	0.59314E-03 0.87469E-02 -0.90008E+02	-0.73305E-03 -0.10929E-01 0.46656E+01	0.58093E-02 0.87470E-01 0.21989E+00	0.78592E-03 0.21542E-01	-0.52276E-02 -0.69793E-01	0.19451E-02 0.23286E-01	-0.63258E-01 -0.11640E+02	0.95424E+01
78	0.0 0.87469E-02 -0.92352E+02	0.0 -0.10929E-01 0.45434E+01	0.0 0.87470E-01 0.0	0.0 0.21542E-01	0.0 -0.69793E-01	0.0 0.23286E-01	-0.63584E-01 0.0	0.0

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

COMPONENT NUMBER = 4

PANEL ID = WNGU WBU

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.25 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA / BETA	C D / L/D	C L / C M	C A / C LL	C Y / C LN	C N
6.00	0.01784	0.08608	0.00975	-0.01093	0.08747
0.0	4.82437	-0.06979	0.02154	0.02329	

DATA SAVED ON UNIT 9. SET NUMBER = 4

SUMMARY PAGE
FOR COMPONENT 4
IN REGION 4

```
INORM,ISURF,IX,IY,IZ =    1   1   1   2   3
U1,UL,V1,VL,CR,CT,CHX =        -47.62939      -76.60291       4.94940       12.19350      -21.33571       0.0          -6.5789
N2 =    15      N3 =    18
XO =          0.0          YO =          0.0          ZO =          0.0
TRANSFORMATION MATRIX,  APIJ, I=1,3, J=1,3
              1.000000              0.0                  0.0
              0.0                   1.000000             0.0
              0.0                   0.0                  1.000000
THE NUMBER OF ELEMENTS IN COMPONENT =     78
THE NUMBER OF ELEMENTS INTERPOLATED =     35
```

DIRECTORY
INFORMATION
FOR REGION 5
SURFACE SPLINE

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

AIR

FIRST PAGE OF FORCE DATA FOR COMPONENT 5 IN REGION 5

ELEMENT DATA MACH= 4.500 ALT = 0. S REF = 102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
XCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 6.00 YAW ANGLE = 0.0 K = 0.0 ETAC = 0.0 DELTA E = 0.0
IDERIV = 0 Q = 0.0 R = 0.0 P = 0.0

L	DEL CA / CA / XCENT	DEL CY / CY / YCENT	DEL CN / CN / ZCENT	DEL CLL / CLL	DEL CLM / CLM	DEL CLN / CLN	CP / DELTA	AREA
1	0.15216E-03	0.33042E-03	0.15219E-02	0.68600E-03	-0.11067E-02	-0.87547E-03	0.13113E+00	0.12329E+01
	0.15216E-03	0.33042E-03	0.15219E-02	0.68600E-03	-0.11067E-02	-0.87547E-03	0.11359E+02	
	-0.72719E+02	0.15463E+02	-0.31494E-01					
2	0.14970E-03	0.37428E-03	0.14972E-02	0.67423E-03	-0.10983E-02	-0.86823E-03	0.12901E+00	0.12329E+01
	0.30197E-03	0.75470E-03	0.30190E-02	0.13602E-02	-0.22050E-02	-0.17437E-02	0.11359E+02	
	-0.73249E+02	0.15463E+02	-0.94481E-01					
3	0.14803E-03	0.37011E-03	0.14805E-02	0.66600E-03	-0.10955E-02	-0.86535E-03	0.12757E+00	0.12329E+01
	0.44990E-03	0.11248E-02	0.44995E-02	0.20262E-02	-0.33005E-02	-0.26091E-02	0.11359E+02	
	-0.73978E+02	0.15452E+02	-0.15747E+00					
4	0.14633E-03	0.36601E-03	0.14640E-02	0.65794E-03	-0.10926E-02	-0.86249E-03	0.12615E+00	0.12330E+01
	0.59629E-03	0.14938E-02	0.59635E-02	0.26842E-02	-0.43931E-02	-0.34715E-02	0.11359E+02	
	-0.74608E+02	0.15463E+02	-0.22046E+00					
5	0.14420E-03	0.36057E-03	0.14422E-02	0.64746E-03	-0.10855E-02	-0.85628E-03	0.12427E+00	0.12329E+01
	0.74049E-03	0.18514E-02	0.74057E-02	0.33316E-02	-0.54786E-02	-0.43278E-02	0.11359E+02	
	-0.75238E+02	0.15463E+02	-0.28345E+00					
12	0.37971E-03	0.94929E-03	0.37971E-02	0.14802E-02	-0.25788E-02	-0.20294E-02	0.12669E+00	0.31842E+01
	0.11202E-02	0.29037E-02	0.11203E-01	0.48118E-02	-0.80574E-02	-0.63562E-02	0.11360E+02	
	-0.67507E+02	0.13393E+02	-0.70505E-01					
13	0.37300E-03	0.93251E-03	0.37300E-02	0.14502E-02	-0.25863E-02	-0.20309E-02	0.12445E+00	0.31842E+01
	0.14932E-02	0.37332E-02	0.14933E-01	0.67620E-02	-0.10644E-01	-0.83871E-02	0.11360E+02	
	-0.69317E+02	0.13393E+02	-0.21152E+00					
14	0.36662E-03	0.91655E-03	0.36662E-02	0.14216E-02	-0.25943E-02	-0.20338E-02	0.12232E+00	0.31842E+01
	0.18599E-02	0.46497E-02	0.18599E-01	0.76836E-02	-0.13238E-01	-0.10421E-01	0.11360E+02	
	-0.70727E+02	0.13393E+02	-0.35253E+00					
15	0.36060E-03	0.90150E-03	0.36060E-02	0.13945E-02	-0.26030E-02	-0.20375E-02	0.12031E+00	0.31842E+01
	0.22204E-02	0.55512E-02	0.22205E-01	0.90781E-02	-0.15841E-01	-0.12450E-01	0.11360E+02	
	-0.72138E+02	0.13393E+02	-0.49354E+00					

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

LAST PAGE OF FORCE DATA FOR COMPONENT 5 IN REGION 5

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

AIR

ELEMENT DATA MACH= 4.500 ALT = 0. S REF = 102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
XCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 6.00 YAW ANGLE = 0.0 K = 0.0 FTAC = 0.0 DELTA E = 0.0
IDERIV = 0 Q = 0.0 R = 0.0 P = 0.0

L	DEL CA CA XCENT	DEL CY CY YCENT	DEL CN CN ZCENT	DEL CLL CLL	DEL CLM CLM	DEL CLN CLN	CP DELTA	AREA
58	0.84619E-03 0.11961E-01 -0.60374E+02	0.21155E-02 0.23653E-01 0.67566E+01	0.84620E-02 0.11961E+00 -0.57371E+00	0.16075E-02 0.34355E-01	-0.51171E-02 -0.76538E-01	-0.38920E-02 -0.59243E-01	0.10357E+00 0.11360E+02	0.86799E+01
59	0.79737E-03 0.12658E-01 -0.64769E+02	0.19934E-02 0.31646E-01 0.67566E+01	0.79737E-02 0.12659E+00 -0.13632E+01	0.14921E-02 0.35847E-01	-0.51355E-02 -0.81674E-01	-0.38939E-02 -0.63137E-01	0.97593E-01 0.11360E+02	0.86799E+01
60	0.75967E-03 0.13418E-01 -0.68164E+02	0.18992E-02 0.33545E-01 0.67566E+01	0.75967E-02 0.13418E+00 -0.17527E+01	0.14000E-02 0.37247E-01	-0.51915E-02 -0.86865E-01	-0.39255E-02 -0.67063E-01	0.92979E-01 0.11360E+02	0.86799E+01
68	0.57104E-04 0.13475E-01 -0.49739E+02	0.14805E-03 0.33693E-01 0.54538E+01	0.56889E-03 0.13475E+00 -0.18633E+00	0.93008E-04 0.37340E-01	-0.28307E-03 -0.87149E-01	-0.22420E-03 -0.67287E-01	0.11860E+00 0.11352E+02	0.51083E+00
69	0.35665E-04 0.13510E-01 -0.54484E+02	0.81676E-04 0.33774E-01 0.54746E+01	0.35973E-03 0.13511E+00 -0.65425E+00	0.57981E-04 0.37398E-01	-0.19623E-03 -0.87345E-01	-0.13570E-03 -0.67423E-01	0.10870E+00 0.11371E+02	0.34974E+00
70	0.57112E-04 0.13567E-01 -0.59767E+02	0.11910E-03 0.33894E-01 0.55417E+01	0.56358E-03 0.13568E+00 -0.10887E+01	0.89778E-04 0.37488E-01	-0.33476E-03 -0.87679E-01	-0.21354E-03 -0.67636E-01	0.10055E+00 0.11486E+02	0.59552E+00
71	0.12676E-03 0.13694E-01 -0.62954E+02	0.32665E-03 0.34200E-01 0.55585E+01	0.12549E-02 0.13693E+00 -0.15312E+01	0.18976E-03 0.37677E-01	-0.79208E-03 -0.88471E-01	-0.58370E-03 -0.68220E-01	0.93986E-01 0.11430E+02	0.14168E+01
72	0.21862E-03 0.13913E-01 -0.67311E+02	0.54836E-03 0.34749E-01 0.54128E+01	0.21935E-02 0.13913E+00 -0.20034E+01	0.31423E-03 0.37992E-01	-0.14809E-02 -0.89952E-01	-0.11121E-02 -0.69332E-01	0.88584E-01 0.11340E+02	0.26307E+01

```
SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0
CASE     **                                                                    PAGE   59
CASE 101A  SAMPLE PROBLEM USING INPUT PRESSURES
COMPONENT NUMBER =  5
PANEL ID =  WNGL    WBL
  MACH=    4.500  VEL=    5024.0 FT/SEC  RE/FT =  0.31945E+08
  ALT =       0.
    *AIR*
  S REF =    102.58  SPAN =    34.29  MAC   =  100.00
  X CG  =      0.0   Y CG =     0.0   Z CG =     0.0
FORCE DATA
  ALPHA     C D        C L        C A        C Y        C N
  BETA      L/D        C M        C LL       C LN

   6.00   0.02838    0.13591    0.01391    0.03475    0.13913
   0.0    4.82429   -0.08795    0.03799   -0.06933

          DATA SAVED ON UNIT 9.  SET NUMBER =   5
```

SUMMARY PAGE
FOR COMPONENT 5
IN REGION 5

```
INORM,ISURF,IX,IY,I2 =    1    1    1    2    3
U1,UL,V1,VL,CR,CT,CHX =        -68.96510      -76.60291      3.37250      13.77040      -24.85080      0.0          -6.7853
N2 =    15      N3 =    18
XO =          0.0          YO =          0.0          ZO =          0.0
TRANSFORMATION MATRIX,  APIJ, I=1,3, J=1,3
            1.000000            0.0                 0.0
            0.0                 1.000000            0.0
            0.0                 0.0                 1.000000
THE NUMBER OF ELEMENTS IN COMPONENT =    78
THE NUMBER OF ELEMENTS INTERPOLATED =    41
```

DIRECTORY
INFORMATION
FOR REGION 6
SURFACE SPLINE

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

FIRST PAGE OF FORCE DATA FOR COMPONENT 6 IN REGION 6

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

AIR

ELEMENT DATA MACH= 4.500 ALT = 0. S REF = 102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
XCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 6.00 YAW ANGLE = 0.0 K = 0.0 ETAC = 0.0 DELTA E = 0.0
IDERIV = 0 Q = 0.0 R = 0.0 P = 0.0

L	DEL CA CA XCENT	DEL CY CY YCENT	DEL CN CN ZCENT	DEL CLL CLL	DEL CLM CLM	DEL CLN CLN	CP DELTA	AREA
6	-0.16574E-05	0.20717E-05	0.16576E-04	0.74587E-05	-0.12571E-04	-0.38367E-05	0.14252E-02	0.12057E+01
	-0.16574E-05	0.20717E-05	0.16576E-04	0.74587E-05	-0.12571E-04	-0.38367E-05	0.28796E+00	
	-0.75848E+02	0.15463E+02	-0.28345E+00					
7	-0.35122E-06	0.43977E-05	0.25126E-05	0.15814E-05	-0.26863E-05	-0.82104E-06	0.30266E-03	0.12057E+01
	-0.20096E-05	0.25106E-05	0.20089E-04	0.90400E-05	-0.15257E-04	-0.46578E-05	0.28785E+00	
	-0.76498E+02	0.15463E+02	-0.22046E+00					
8	0.95446E-06	-0.11929E-05	-0.95459E-05	-0.42998E-05	0.73611E-05	0.22530E-05	-0.82251E-03	0.12057E+01
	-0.10541E-05	0.13178E-05	0.10542E-04	0.47402E-05	-0.78962E-05	-0.24048E-05	0.28796E+00	
	-0.77128E+02	0.15463E+02	-0.15747E+00					
9	0.22900E-05	-0.28403E-05	-0.22802E-04	-0.10276E-04	0.17728E-04	0.54337E-05	-0.19648E-02	0.12057E+01
	0.12259E-05	-0.15315E-05	-0.12260E-04	-0.55359E-05	0.98322E-05	0.30289E-05	0.28774E+00	
	-0.77758E+02	0.15463E+02	-0.94484E-01					
10	0.39022E-05	-0.47512E-05	-0.39026E-04	-0.17146E-04	0.29807E-04	0.91479E-05	-0.32765E-02	0.12057E+01
	0.50281E-05	-0.42937E-05	-0.50286E-04	-0.22682E-04	0.39639E-04	0.12177E-04	0.28785E+00	
	-0.78388E+02	0.15463E+02	-0.31496E-01					
17	0.13518E-04	-0.15898E-04	-0.13518E-03	-0.52452E-04	0.10124E-03	0.31666E-04	-0.45102E-02	0.31138E+01
	0.18544E-04	-0.23180E-04	-0.18547E-03	-0.75134E-04	0.14088E-03	0.43843E-04	0.28720E+00	
	-0.74958E+02	0.13393E+02	-0.63455E+00					
18	0.17439E-04	-0.21739E-04	-0.17439E-03	-0.67754E-04	0.13309E-03	0.41747E-04	-0.58182E-02	0.31139E+01
	0.35985E-04	-0.44779E-04	-0.35986E-03	-0.14289E-03	0.27397E-03	0.85590E-04	0.28725E+00	
	-0.76369E+02	0.13393E+02	-0.49354E+00					
19	0.21170E-04	-0.25463E-04	-0.21170E-03	-0.82361E-04	0.16458E-03	0.51768E-04	-0.70631E-02	0.31138E+01
	0.57155E-04	-0.71442E-04	-0.57156E-03	-0.22525E-03	0.43856E-03	0.13736E-03	0.28720E+00	
	-0.77778E+02	0.13393E+02	-0.35253E+00					
20	0.24713E-04	-0.31801E-04	-0.24713E-03	-0.96269E-04	0.19564E-03	0.61702E-04	-0.82451E-02	0.31138E+01
	0.81868E-04	-0.11233E-03	-0.81869E-03	-0.32152E-03	0.63420E-03	0.19906E-03	0.28714E+00	
	-0.79188E+02	0.13383E+02	-0.21152E+00					

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

LAST PAGE OF FORCE DATA FOR COMPONENT 6 IN REGION 6

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

AIR

ELEMENT DATA MACH= 4.500 ALT = 0. S REF = 102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
XCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 6.00 YAW ANGLE = 0.0 K = 0.0 ETAC = 0.0 DELTA E = 0.0
IDEFIV = 0 O = 0.0 R = 0.0 P = 0.0

L	DEL CA CA XCENT	DEL CY CY YCENT	DEL CN CN ZCENT	DEL CLL CLL	DEL CLM CLM	DEL CLN CLN	CP DELTA	AREA
74	0.81590E-04 0.20711E-02 -0.75042E+02	-0.99417E-04 -0.25933E-02 0.52514E+01	-0.81749E-03 -0.20714E-01 -0.15509E+01	-0.12071E-03 -0.49914E-02	0.61955E-03 0.16449E-01	0.20771E-03 0.54855E-02	-0.20647E-01 0.29829E+00	0.41116E+01
75	0.11245E-03 0.21836E-02 -0.80451E+02	-0.13039E-03 -0.27243E-02 0.51341E+01	-0.11304E-02 -0.21844E-01 -0.11157E+01	-0.16484E-03 -0.51562E-02	0.90829E-03 0.17358E-01	0.31394E-03 0.57994E-02	-0.22851E-01 0.31741E+00	0.51394E+01
76	0.17093E-03 0.23545E-02 -0.85129E+02	-0.21673E-03 -0.29410E-02 0.49479E+01	-0.17093E-02 -0.23554E-01 -0.67317E+00	-0.24242E-03 -0.53986E-02	0.14540E-02 0.18812E-01	0.51345E-03 0.63129E-02	-0.25725E-01 0.28725E+00	0.69046E+01
77	0.26702E-03 0.26215E-02 -0.90008E+02	-0.33555E-03 -0.32757E-02 0.46656E+01	-0.26600E-02 -0.26214E-01 -0.21989E+00	-0.35992E-03 -0.57584E-02	0.23937E-02 0.21205E-01	0.84484E-03 0.71577E-02	-0.28965E-01 0.26556E+00	0.95424E+01
78	0.0 0.26215E-02 -0.92352E+02	0.0 -0.32767E-02 0.45434E+01	0.0 -0.26214E-01 0.0	0.0 -0.57584E-02	0.0 0.21205E-01	0.0 0.71577E-02	-0.30282E-01 0.0	0.0

```
SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0
CASE     **                                                                    PAGE   65
CASE 101A  SAMPLE PROBLEM USING INPUT PRESSURES
COMPONENT NUMBER =  6
PANEL ID =  WNGL   WBL
  MACH=   4.500  VEL=   5024.0 FT/SEC  RE/FT =  0.31945E+08
  ALT =      0.
    *AIR*
  S REF =    102.58  SPAN =    34.29  MAC  =  100.00
  X CG  =      0.0   Y CG =     0.0   Z CG =     0.0
FORCE DATA
  ALPHA     C D        C L        C A        C Y        C N
  BETA      L/D        C M        C LL       C LN
   6.00  -0.00013  -0.02634   0.00262  -0.00328  -0.02621
   0.0  199.23750   0.02121  -0.00576   0.00716
          DATA SAVED ON UNIT 9.  SET NUMBER =   6
 *** AERODYNAMIC ANALYSIS IS COMPLETE
```

SUMMARY PAGE
FOR COMPONENT 6
IN REGION 6

```
***** AERODYNAMIC PROGRAM *****
 CASE 101A, WING AFT (NASA TN D-6441)
     RUN ID =   0      CORRELATION ID =    0
                1 INVISCID PRESSURES  (OPTION 3)
```

2ND AERO OPTION

*** PRESSURE CALCULATION PROGRAM

INVISCID PRESSURES OPTION

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE **

FORCE DATA FOR COMPONENT 7 IN REGION 2

CASE 101A SAMPLE PROBLEM USING INPUT PRESSURES

AIR

ELEMENT DATA MACH= 4.500 ALT = 0. S REF = -102.6 SPAN = 34.3 IMPACT = 0 IMPACI = 0
XCG = 0.0 YCG = 0.0 ZCG = 0.0 MAC = 100.0 ISHAD = 0 ISHADI = 0
ANGLE OF ATTACK = 4.00 YAW ANGLE = 0.0 K = 0.0 ETAC = 0.0 DELTA F = 0.0
IDERIV = 0 Q = 0.0 P = 0.0 P = 0.0

L	DEL CA CA XCENT	DEL CY CY YCENT	DEL CN CN ZCENT	DEL CLL CLL	DEL CLM CLM	DEL CLN CLN	CP DELTA	AREA
1	-0.15939E-05 -0.15939E-05 -0.61853E+02	0.34911E-03 0.34911E-03 0.53193E+01	0.77390E-04 0.77390E-04 -0.76116E+00	0.42563E-05 0.42563E-05	-0.47856E-04 -0.47856E-04	-0.62957E-03 -0.62957E-03	-0.14702E-02 0.10422E+01	0.24951E+02
2	-0.14075E-04 -0.15669E-04 -0.61853E+02	0.33927E-02 0.34318E-02 0.53193E+01	-0.68336E-03 -0.60597E-03 0.76116E+00	-0.37584E-04 -0.33328E-04	0.42257E-03 0.37472E-03	-0.55592E-02 -0.61888E-02	-0.12982E-01 -0.15504E+01	0.24951E+02
3	-0.47375E-03 -0.49942E-03 -0.77249E+02	0.55517E-02 0.90335E-02 0.45925E+01	0.12528E-02 0.64685E-03 -0.76116E+00	0.42344E-04 0.90167E-05	-0.96410E-03 -0.58947E-03	-0.12670E-01 -0.18859E-01	-0.20434E-01 -0.33579E+01	0.29159E+02
4	-0.73035E-03 -0.12108E-02 -0.77249E+02	0.87128E-02 0.17796E-01 0.45925E+01	-0.19314E-02 -0.12846E-02 0.76115E+00	-0.65280E-04 -0.56263E-04	0.14964E-02 0.89697E-03	-0.19533E-01 -0.38392E-01	-0.31501E-01 -0.59504E+01	0.29159E+02

```
SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0
CASE     **                                                                    PAGE    67
CASE 101A  SAMPLE PROBLEM USING INPUT PRESSURES
COMPONENT NUMBER =  1
PANEL ID =      C
  MACH=    4.500  VEL=    5024.0 FT/SEC  RE/FT =  0.31945E+08
  ALT =      0.
    *AIR*
  S REF =  -102.58  SPAN =    34.29  MAC  =  100.00
  X CG  =     0.0   Y CG =     0.0   Z CG =     0.0
FORCE DATA
  ALPHA     C D        C L        C A        C Y        C N
  BETA      L/D        C M        C LL       C LN

   6.00  -0.00135  -0.00115  -0.00122   0.01780  -0.00128
   0.0    0.85354   0.00030  -0.00006  -0.03839
          DATA SAVED ON UNIT 9.  SET NUMBER =   7
*** AERODYNAMIC ANALYSIS IS COMPLETE
```

SUMMARY PAGE
FOR COMPONENT 7
IN REGION 2

(NOTE NEGATIVE
S REF)

```
***** AERODYNAMIC PROGRAM *****
 CASE 101A, WING AFT (NASA TN D-6441)
     RUN ID =   0      CORRELATION ID =    0
                1 SPECIAL ROUTINES     (OPTION 5)
```

3RD AERO OPTION
SUMMATIONS OF
COMPONENT FORCES

NOTE: THE PRINTED OUTPUT
FOR THIS OPTION
IS GIVEN IN ITS
ENTIRETY.

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

COMPONENT DATA SET NUMBER 1

MACH= 4.500 VEL = 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA BETA	C D L/D	C L C M	C A C LL	C Y C LN	C N
6.00 0.0	0.03511 3.45390	0.12125 -0.03342	0.02224 0.00000	0.11203 -0.07697	0.12426

COMPONENT DATA SET NUMBER 2

MACH= 4.500 VEL = 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA BETA	C D L/D	C L C M	C A C LL	C Y C LN	C N
6.00 0.0	0.02381 4.98417	0.11967 -0.08142	0.01127 -0.00000	-0.08307 0.20031	0.12051

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

SUMMATION DATA SET NUMBER 1

```
        1   0   0   0   0   0   0   0   0   0
        1   0   0   0   0   0   0   0   0   0

  MACH=    4.500  VEL=    5024.0 FT/SEC  RE/FT =  0.31945E+08
  ALT =      0.
    *AIR*

  S REF =    102.58  SPAN =    34.29  MAC  =  100.00
  X CG  =      0.0   Y CG =     0.0   Z CG =     0.0

FORCE DATA
  ALPHA      C D        C L        C A        C Y        C N
  BETA       L/D        C M        C IL       C LN

   6.00    0.11783    0.47285    0.06703    0.0        0.48954
   0.0     4.07234   -0.22947    0.0        0.0

          DATA SAVED ON UNIT 9.  SUMMATION SET NUMBER =   1
```

COMPONENT DATA SET NUMBER 7

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.59 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA BETA	C D L/D	C L C M	C A C LL	C Y C LN	C N
6.00 0.0	-0.00135 0.85354	-0.00115 0.00390	-0.00122 -0.00006	0.01780 -0.03839	-0.00128

```
SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0
CASE      0                                                                                    PAGE   72

SUMMATION DATA SET NUMBER    2
        5    0    0    0    0    0    0    0    0    0

  MACH=    4.500  VEL=    5024.0 FT/SEC  RE/FT =  0.31945E+08
  ALT =       0.
    *AIR*

  S REF =    102.58  SPAN =    34.29  MAC  =  100.00
  X CG  =      0.0   Y CG =     0.0   Z CG =     0.0

FORCE DATA
  ALPHA     C D        C L        C A        C Y        C N
  BETA      L/D        C M        C LL       C LN

   6.00   -0.00269   -0.00230   -0.00244    0.0       -0.00257
   0.0     0.85354    0.00179    0.0        0.0

          DATA SAVED ON UNIT 9.  SUMMATION SET NUMBER =    2
```

PREVIOUS SUMMATION NUMBER 1

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
ATR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA / BETA	C D / I/D	C L / C M	C A / C LL	C Y / C LN	C N
6.00	0.11783	0.47395	0.06703	0.0	0.48954
0.0	4.07234	-0.22967	0.0	0.0	

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

PREVIOUS SUMMATION NUMBER 2

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA BETA	C D L/D	C L C M	C A C LL	C Y C LN	C N
6.00 0.0	-0.00269 0.85354	-0.00230 0.00179	-0.00244 0.0	0.0 0.0	-0.00257

SUMMATION DATA SET NUMBER 3

1	2	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0

MACH= 4.500 VFL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA BETA	C D L/D	C L C M	C A C LL	C Y C LN	C N
6.00 0.0	0.11514 4.14767	0.47755 -0.22789	0.06459 0.0	0.0 0.0	0.48697

DATA SAVED ON UNIT 9. SUMMATION SET NUMBER = 3

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

COMPONENT DATA SET NUMBER 3

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA BETA	C D L/D	C L C M	C A C LL	C Y C LN	C N
6.00 0.0	0.00011 198.77116	0.02168 -0.01359	-0.00216 0.00504	-0.00538 0.01039	0.02157

COMPONENT DATA SET NUMBER 4

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA BETA	C D L/D	C L C M	C A C LL	C Y C LN	C N
6.00 0.0	0.01784 4.82437	0.08608 -0.06979	0.00875 0.02154	-0.01093 0.02329	0.08747

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

SUMMATION DATA SET NUMBER 4

```
2   4   0   0   0   0   0   0   0   0
2   4   0   0   0   0   0   0   0   0
```

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA / BETA	C D / L/D	C L / C M	C A / C LL	C Y / C LN	C N
6.00	0.03550	0.21551	0.01318	0.0	0.21809
0.0	6.00269	-0.16676	0.0	0.0	

DATA SAVED ON UNIT 9. SUMMATION SET NUMBER = 4

COMPONENT DATA SET NUMBER 5

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA BETA	C D L/D	C L C M	C A C LL	C Y C LN	C N
6.00 0.0	0.02838 4.82429	0.13691 -0.08955	0.01391 0.03799	0.03475 -0.06933	0.13913

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

COMPONENT DATA SET NUMBER 6

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA BETA	C D L/D	C L C M	C A C LL	C Y C LN	C N
6.00 0.0	-0.00013 198.23750	-0.02634 0.02121	0.00262 -0.00576	-0.00328 0.00716	-0.02621

```
SUMMATION DATA SET NUMBER    5
        3   5   0   0   0   0   0   0   0   0
        3   5   0   0   0   0   0   0   0   0

  MACH=    4.500  VEL=    5024.0 FT/SEC  RE/FT =  0.31945E+08
  ALT =      0.
    *AIR*

  S REF =    102.58  SPAN =    34.29  MAC  =  100.00
  X CG  =      0.0   Y CG =     0.0   Z CG =     0.0

FORCE DATA
  ALPHA     C D        C L        C A        C Y        C N
  BETA      L/D        C M        C LL       C LN

   6.00   0.05649   0.22113   0.03307   0.0       0.22593
   0.0    3.91434  -0.13749   0.0       0.0

          DATA SAVED ON UNIT 9.  SUMMATION SET NUMBER =   5
```

PREVIOUS SUMMATION NUMBER 4

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.59 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA BETA	C D L/D	C L C M	C A C LL	C Y C LN	C N
6.00 0.0	0.03590 6.00269	0.21551 -0.16676	0.01318 0.0	0.0 0.0	0.21809

PREVIOUS SUMMATION NUMBER 5

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA BETA	C D L/D	C L C M	C A C LL	C Y C LN	C N
6.00 0.0	0.05649 3.91434	0.22113 -0.13749	0.03307 0.0	0.0 0.0	0.22583

```
SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0
CASE     0                                                                    PAGE   84

SUMMATION DATA SET NUMBER   6
        3   4   0   0   0   0   0   0   0   0
        5   6   0   0   0   0   0   0   0   0

  MACH=    4.500   VEL =    5024.0 FT/SEC   RE/FT =   0.31945E+08
  ALT =      0.
    *AIR*

  S REF =    102.58   SPAN =    34.29   MAC   =   100.00
  X CG  =      0.0    Y CG =     0.0    Z CG  =     0.0

FORCE DATA
  ALPHA     C D        C L        C A        C Y        C N
  BETA      L/D        C M        C LL       C LN

   6.00   0.09240   0.43665   0.04625   0.0        0.44391
   0.0    4.72583  -0.30425   0.0       0.0

          DATA SAVED ON UNIT 9.  SUMMATION SET NUMBER =   6
```

SUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, MARK IVS MOD 0

CASE 0

PREVIOUS SUMMATION NUMBER 3

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.25 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA / BETA	C D / L/D	C L / C M	C A / C LL	C Y / C LN	C N
6.00	0.11514	0.47755	0.06459	0.0	0.48697
0.0	4.14767	-0.22788	0.0	0.0	

PREVIOUS SUMMATION NUMBER 6

MACH= 4.500 VEL= 5024.0 FT/SEC RE/FT = 0.31945E+08
ALT = 0.
AIR

S REF = 102.58 SPAN = 34.29 MAC = 100.00
X CG = 0.0 Y CG = 0.0 Z CG = 0.0

FORCE DATA

ALPHA BETA	C D L/D	C L C M	C A C LL	C Y C LN	C N
6.00 0.0	0.09240 4.72583	0.43665 -0.30425	0.04625 0.0	0.0 0.0	0.44391

```
SUMMATION DATA SET NUMBER    7
        1   2   0   0   0   0   0   0   0   0
        4   5   6   0   0   0   0   0   0   0

  MACH=    4.500   VEL=    5024.0 FT/SEC   RE/FT =  0.31945E+08
  ALT =      0.
    *ATR*

  S REF =    102.58   SPAN =    34.29   MAC   =  100.00
  X CG  =      0.0    Y CG =     0.0    Z CG  =    0.0

FORCE DATA
  ALPHA     C D        C L        C A        C Y        C N
  BETA      L/D        C M        C LL       C LN

   6.00   0.20753   0.91520   0.11084   0.0        0.93088
   0.0    4.40507  -0.53213   0.0       0.0

          DATA SAVED ON UNIT 9.  SUMMATION SET NUMBER =   7

 *** AERODYNAMIC ANALYSIS IS COMPLETE
```

```
******** MAIN PROGRAM NOW HAS CONTROL OF SYSTEM ********

***** PROGRAM HAS REACHED NORMAL TERMINATION ******
```

APPENDIX C

OVERLAY STRUCTURE AND GO-DECK SETUP

I. OVERLAY STRUCTURE

The original MARK IV Program used the FORTRAN overlay features for normal operation. The present stripped version, MARK IVS, follows the original overlay structure very closely. The subroutines and links no longer required have been removed, and the new routines (INPUT, SHEAR, and HNGFT) have been added.

CDC Overlay

```
OVERLAY (MARK4,0,0)
     MAIN, HEADER, MONITR, ARCOS, ARSIN
OVERLAY (MARK4,1,0)
     GEØM, PUNCH
OVERLAY (MARK4,1,1)
     IELE
OVERLAY (MARK4,1,2)
     ELLIP
OVERLAY (MARK4,1,3)
     CUBIC
OVERLAY (MARK4,1,4)
     AIRCFT, SURF, NACEL, FUSE, STATS3
OVERLAY (MARK4,2,0)
     AERØ, SOLVIT, GETT, SAVE, ROWFM1, ROWFM2, MIS1
OVERLAY (MARK4,2,1)
     FLØW, FFSURF
OVERLAY (MARK4,2,2)
     INPUT
OVERLAY (MARK4,2,3)
     PRES, FORCE, CPINPT, VALU2, SFNTR2
OVERLAY (MARK4,2,5)
     SPEC, SUM
OVERLAY (MARK4,2,7)
     SHEAR, HNGFT
```

IBM Overlay

```
ENTRY MAIN
INSERT MAIN,HEADER,EROR,MONITR
INSERT EXEC,TAPE,ASSOCV,TAGS
     OVERLAY ALPHA
        INSERT GEOM
          INSERT GFLAG
           OVERLAY BETA
             INSERT IELE,ELLIP,CUBIC,PUNCH
           OVERLAY BETA
             INSERT AIRCFT,SURF,FUSE,NACEL,STATS3
               INSERT ACFT,AREOIL,FUSD,NACELD
```

```
OVERLAY ALPHA
  INSERT AERO
    INSERT FLIGHT,GASD,FSBS,REF,ABDATA,PLOTCP,INDATA
    INSERT SAVSPL,GDATA,FARRAY,MDATA
     OVERLAY BETA
       INSERT FLOW,PRESURE
         INSERT FFIELD
     OVERLAY BETA
       INSERT INPUT
     OVERLAY BETA
       INSERT DRES,CPINPT,FORCE,SENTRO,VALU2
       INSERT POWFM2,MISL,SOLVIT,POWFM1,SAVE,GETT,SUREN2
     OVERLAY BETA
       INSERT SPEC,SUM
     OVERLAY BETA
       INSERT SHEAR,ENGET
```

II. IBM GØ-DECK SETUP

An example set of JCL cards for operation of the MARK IVS Program in IBM 360 and 370 computers is given below.

```
// EXEC FORTGO,REGION=300K
//STEPLIB DD DSNAME=TSOT3BW.K7HC.MKIVS.LOAD,DISP=SHR
//GO.FT01F001  DD DSN=&FT01,UNIT=SYSDA,SPACE=(TRK,(10,5)),
//               DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//GO.FT02F001 DD DSN=&FT02,UNIT=SYSDA,SPACE=(TRK,(10,5)),
//     DCB=(BUFNO=1,BLKSIZE=6447,RECFM=VBS)
//GO.FT03F001  DD DUMMY
//GO.FT04F001  DD DSN=DAC.F0H5.DNS.A101.GEOMD,
//               SPACE=(100,(2000)),
//               UNIT=3330-1,
//               DCB=(BUFNO=1,DSORG=DA),
//               DISP=(SHR,PASS),VOL=SER=CSLB30
//GO.FT07F001  DD DUMMY
//GO.FT08F001  DD DSN=&FT08,UNIT=SYSDA,SPACE=(TRK,(25,5)),
//               DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//GO.FT09F001  DD DSN=DAC.F0H5.DNS.A101.FORCED,
//               SPACE=(164,(500)),
//               UNIT=3330-1,
//               DCB=(BUFNO=1,DSORG=DA),
//               DISP=(SHR,PASS),VOL=SER=CSLB30
//GO.FT10F001  DD DSN=DAC.F0H5.DNS.A101.FFIELD,
//               SPACE=(100,(2000)),
//               UNIT=3330-1,
//               DCB=(BUFNO=1,DSORG=DA),
//               DISP=(SHR,PASS),VOL=SER=CSLB30
//GO.FT11F001  DD DSN=&FT11,UNIT=SYSDA,SPACE=(TRK,(10,5)),
//               DCB=(BUFNO=1,BLKSIZE=6447,RECFM=VBS)
//GO.FT12F001  DD DSN=&FT12,UNIT=SYSDA,SPACE=(TRK,(10,5)),
//               DCB=(BUFNO=1,BLKSIZE=6447,RECFM=VBS)
//GO.FT13F001  DD DSN=&FT13,UNIT=SYSDA,SPACE=(TRK,(10,5)),
//               DCB=(BUFNO=1,BLKSIZE=6447,RECFM=VBS)
//GO.FT14F001  DD DSN=&FT14,UNIT=SYSDA,SPACE=(TRK,(10,5)),
//               DCB=(BUFNO=1,BLKSIZE=6447,RECFM=VBS)
//GO.FT15F001  DD DSN=&FT15,
//               SPACE=(400,(2000)),
//               DCB=(BUFNO=1,DSORG=DA),
//               UNIT=SYSDA
//GO.FT16F001  DD DSN=&FT16,
//               SPACE=(80,(5000)),
//               DCB=(BUFNO=1,DSORG=DA),
//               UNIT=SYSDA
//GO.FT17F001  DD DUMMY
//GO.FT18F001  DD DUMMY
//GO.FT19F001  DD DUMMY
```

```
//GO.FT20F001  DD DSN=&FT20,
//             SPACE=(432,(115)),
//             DCB=(BUFNO=1,DSORG=DA),
//             UNIT=SYSDA
//GO.FT21F001  DD DSN=&FT21,
//             SPACE=(144,(1034)),
//             DCB=(BUFNO=1,DSORG=DA),
//             UNIT=SYSDA
//GO.FT22F001  DD DUMMY
//GO.FT88F001  DD DSN=&FT88,UNIT=SYSDA,SPACE=(TRK,(5,1)),
//             DCB=(BUFNO=1,BLKSIZE=6447,RECFM=VBS)
//GO.SYSIN     DD *
/*
//
```

APPENDIX D

PROGRAM LISTINGS

This part of the report contains the source card listings for the MARK IVS Mod 0 Version of the Supersonic-Hypersonic Arbitrary-Body Program. The listing as presented here is for the CDC version of the program and is written in FORTRAN for the CDC FTN compiler.

In this listing all cards that are peculiar to the CDC version of FORTRAN are identified by a C in card column 80. All cards that are peculiar to the IBM FORTRAN IV compiler are identified by an I in card column 80 and a C in card column 1. In other words, the code for both CDC and IBM machines is in the deck but the IBM cards are made inactive by converting them to comment statements (C in card column 1). Since all of the machine dependent cards are identified by an I or C in card column 80 it is a simple matter to convert the deck from one version to the other with a small conversion program. When converting from CDC to IBM code this conversion program reads and copies each card to a storage unit. If a card has a C in card column 80, then a C is written into card column 1 to make the CDC peculiar card inactive. If a card has an I in card column 80, then the C is removed from card column 1 and the card image written to the storage unit as an active FORTRAN statement. The conversion from IBM back to CDC is made in a similar manner.

The conversion program to convert the deck from CDC to IBM FORTRAN is listed below (for use on an IBM machine).

```
      DIMENSION DATA(22)
      DATA CB,CC,CI/1H ,1HC,1HI/
      REWIND 19
      DO 100 I=1, 20000
        READ (5,20,END=300) DATA
 20        FORMAT (1A1,19A4,1A2,1A1)
      IF (DATA(22) .EQ. CI) DATA(1) = CB
      IF (DATA(22) .EQ. CC) DATA(1) = CC
      WRITE (19,20) DATA
100   CONTINUE
300   STOP
      END
```

This program places the new deck with IBM cards made active, and CDC cards inactive, on to Unit 19. The references to Unit 19 above can be changed to Unit 7 to punch the deck out.

An index for the subroutine listings is provided on the next page. Except for the Executive Main which is listed first, the subroutines are in alphabetical order.

INDEX TO SUBROUTINE LISTINGS

ROUTINE NAME	DESCRIPTION	DECK ID	PAGE
MAIN	MARK IVS Program Executive Routine	MAIN	226
AERO	Aerodynamic Analysis Executive Routine	AERO	230
AIRCFT	Aircraft Geometry Executive Routine	AIRC	244
ARCØS	Arc-Cosine Conversion Routine (CDC)	ARCO	254
ARSIN	Arc-Sine Conversion Routine (CDC)	ARSI	255
CPINPT	Pre-Processor for Surface Data	CPIN	256
CUBIC	Parametric Cubic Routine	CUBI	270
ELLIP	Elliptical Cross-Section Routine	ELLI	277
ERØR	IBM ABEND Dump Routine	EROR	284
FFSURF	Surface Data Transfer Routine	FFSU	285
FLOW	Flow Field Analysis Executive Routine	FLOW	293
FORCE	Force Calculation Routine	FORC	300
FUSE	Fuselage Geometry Generation Routine	FUSE	309
GEØM	Geometry Data Preparation and Control	GEOM	313
GETT	Data Retrieval Routine	GETT	326
HEADER	Page Header Subroutine	HEAD	327
HNGFT	Calculates Hinge-Moment Factors	HNGF	328
IELE	Element Data Transfer Routine	IELE	334
INPUT	Automated Aero Input Routine	INPT	336
MIS1	Matrix Solution Routine	MIS1	345
MONITR	Input Data Read and Print Routine	MONI	349
NACEL	Nacelle Geometry Generation Routine	NACE	350
PRES	Pressure Calculation Executive Routine	PRES	353
PUNCH	Element Data Write Routine	PUNC	363
ROWFM1	Matrix Formation Routine for SOLVIT	ROW1	365
ROWFM2	Matrix Formation Routine for MIS1	ROW2	367
SAVE	Data Save Routine	SAVE	369
SFNTR2	Surface Spline Definition Routine	SFN2	370
SHEAR	Shear Calculation Routine	SHEA	381
SOLVIT	Matrix Solution Routine	SOLV	396
SPEC	Special Function Executive Routine	SPEC	406
STATS3	Dummy Zero-Area Element Routine	STAT	407
SUM	Force Data Summation Routine	SUM	408
SURF	General Airfoil Surface Routine	SURF	415
VALU2	Spline Interpolation Routine	VAL2	423

```
      OVERLAY(MARK4,0,0)
      PROGRAM MAIN (INPUT,OUTPUT, PUNCH=129,TAPE5=INPUT,TAPE6=OUTPUT,
     1  TAPE7=PUNCH,TAPE1,TAPE2=129,TAPE3=129,TAPE4=2000,TAPE8=129,
     2  TAPE9,TAPE10,TAPE11,TAPE12,TAPE13,
     3  TAPE14,TAPE15,TAPE16,TAPE17,TAPE18,
     4  TAPE19,TAPE20,TAPE21,TAPE22=129,TAPE88)
C*********************************************************************
C*****                    STRIPPED  VERSION                    *****
C** SUPERSONIC-HYPERSONIC ARBITRARY-BODY AERODYNAMIC COMPUTER PROGRAM **
C*****                    CDC/IBM MODEL                        *****
C*********************************************************************
C
C     THIS IS THE MARK IVS MOD 0 VERSION OF THE SUPERSONIC-HYPERSONIC
C     ARBITRARY-BODY FORCE ANALYSIS PROGRAM.  THE PROGRAM WAS DEVELOPED
C     UNDER SPONSORSHIP OF THE NASA AMES RESEARCH CENTER (CONTRACT NAS2-
C     7585).  THE ORIGINAL MARK IV PROGRAM HAS BEEN MODIFIED SUCH THAT
C     ONLY THE INPUT PRESSURE OPTION IS AVAILABLE (IPIN=1).  ALL THE
C     OTHER PRESSURE METHOD OPTIONS AND VISCOUS CALCULATIONS HAVE BEEN
C     STRIPPED FROM THE PROGRAM.
C
C********** THIS PROGRAM WRITTEN BY D. N. SMYTH AND H. C. LOO **********
C*****                RELEASE DATE    APRIL  1979                *****
C*********************************************************************
C
C
C
C***** EXECUTIVE MAIN PROGRAM *****
C
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR,INPUTD
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      COMMON/SUBIDX/SIDX04(1001),SIDX16(1001),SIDX21(1001)
      COMMON/ASSOCV/IA15,IA16
      COMMON /MASTER/ MIDX04(21),INDEX9(501),INDX10(2001),INDX15(2001),
     1  MIDX16(94),INDX20(2421),MIDX21(73)
      DIMENSION  IPG(20),TITLE(15)
      INTEGER SIDX04,SIDX16,SIDX21
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      INTEGER      PAGE,ERROR
C     EXTERNAL EROR
C     DEFINE FILE 4( 3000,25,U,ITAG4)
C     DEFINE FILE 4(20000,25,U,ITAG4)
C     DEFINE FILE 9  (500,41,U,ITAG9)
C     DEFINE FILE 10(2000,25,U,ITAG10)
```

```
C       DEFINE FILE  15(2000,100,U,IA15)                                MAIN  046I
C       DEFINE FILE 16(92720,20,U,IA16)                                 MAIN  047I
C       DEFINE FILE 16(5000,20,U,IA16)                                  MAIN  048I
C       DEFINE FILE 20(2420,108,U,IA20)                                 MAIN  049I
C       DEFINE FILE 20(115,108,U,IA20)                                  MAIN  050I
C       DEFINE FILE 21(71440,36,U,IA21)                                 MAIN  051I
C       DEFINE FILE 21(1034,36,U,IA21)                                  MAIN  052I
        CALL OPENMS ( 4,MIDX04,21,0)                                    MAIN  053C
        CALL OPENMS (9,INDEX9,501,0)                                    MAIN  054C
        CALL OPENMS (10,INDX10,2001,0)                                  MAIN  055C
        CALL OPENMS (15,INDX15,2001,0)                                  MAIN  056C
        CALL OPENMS (16,MIDX16,  94,0)                                  MAIN  057C
        CALL OPENMS (20,INDX20,2421,0)                                  MAIN  058C
        CALL OPENMS (21,MIDX21,  73,0)                                  MAIN  059C
        ERROR = 0                                                       MAIN  060
        PAGE = 1                                                        MAIN  061
        TAPEIN = 5                                                      MAIN  062
        TAPEOT = 6                                                      MAIN  063
        TAPEA = 3                                                       MAIN  064
        TAPEB = 2                                                       MAIN  065
        TAPEC = 11                                                      MAIN  066
        TAPED = 14                                                      MAIN  067
        TAPEE = 12                                                      MAIN  068
        TAPEF = 13                                                      MAIN  069
        TAPEG = 7                                                       MAIN  070
        REWIND TAPEA                                                    MAIN  071
        REWIND TAPEB                                                    MAIN  072
        REWIND TAPEC                                                    MAIN  073
        REWIND TAPEE                                                    MAIN  074
        TAPEH = 17                                                      MAIN  075
        TAPEI = 18                                                      MAIN  076
        TAPEJ = 19                                                      MAIN  077
        REWIND TAPEF                                                    MAIN  078
        REWIND 8                                                        MAIN  079
        REWIND  22                                                      MAIN  080
        DO 5 I=1,1001                                                   MAIN  081C
        SIDX04(I)=0                                                     MAIN  082C
        SIDX16(I)=0                                                     MAIN  083C
      5 SIDX21(I)=0                                                     MAIN  084C
C                                                                       MAIN  085
C ***********************************************************************MAIN  086
C ***** READ EXECUTIVE FLAG CARD                                        MAIN  087
        READ (TAPEIN,10) IEROR,INMONT,INPUTD                            MAIN  088
   10   FORMAT (3I1)                                                    MAIN  089
C       IF (IEROR .EQ. 0) CALL ERRSET (207,256,256,2,EROR,301)          MAIN  090I
```

```
C     CALL ERRSET (208,256,-1,1,1)                                      MAIN  091I
      IF((INMONT .EQ. 0).AND.(INPUTD .EQ. 1))  CALL MONITR              MAIN  092
C                                                                       MAIN  093
C ***** READ SYSTEM CONTROL CARD TO SELECT PROGRAMS TO BE USED AND ORDER MAIN  094
C ***** OF USE. A MAXIMUM OF 20 OPTIONS MAY BE USED.  THIS VERSION OF   MAIN  095
C ***** THE PROGRAM WILL ALLOW ONLY OPTIONS  1 & 2                      MAIN  096
C*********************************************************************** MAIN  097
C                                                                       MAIN  098
      READ (TAPEIN,20) IPG,CASE                                         MAIN  099
   20 FORMAT (20I1,41X,A2)                                              MAIN  100
      WRITE (6,30)                                                      MAIN  101
   30 FORMAT (1H1,////,/1H ,39H***SUPERSONIC-HYPERSONIC ARBITRARY-BODY  MAIN  102
     1 50H AERODYNAMIC COMPUTER PROGRAM SYSTEM ************ ,///,1H ,   MAIN  103
     2 10X,46HPROGRAM OPTIONS ARE IN THE FOLLOWING ORDER....   ,/1H  ) MAIN  104
      IF (IPG(1) .EQ. 0) GO TO 220                                      MAIN  105
C                                                                       MAIN  106
C                                                                       MAIN  107
      DO 80 I=1,20                                                      MAIN  108
      IF (IPG(I) .EQ. 0) GO TO 90                                       MAIN  109
      IF (IPG(I) .GT. 4) GO TO 240                                      MAIN  110
      IF (IPG(I) .EQ. 1) WRITE (TAPEOT,40) I                            MAIN  111
   40  FORMAT (1H0,15X,I2,30H  GEOMETRY PROGRAM  (OPTION 1) )           MAIN  112
      IF (IPG(I) .EQ. 2) WRITE (TAPEOT,50) I                            MAIN  113
   50  FORMAT (1H0,15X,I2,33H  AERODYNAMIC PROGRAM  (OPTION 2)   )      MAIN  114
   80 CONTINUE                                                          MAIN  115
   90 I = 0                                                             MAIN  116
C                                                                       MAIN  117
  100 I = I + 1                                                         MAIN  118
      IF (IPG(I) .EQ. 0) GO TO 260                                      MAIN  119
C                                                                       MAIN  120
      IPROG = IPG(I)                                                    MAIN  121
      GO TO (110,120),IPROG                                             MAIN  122
C 110 CALL GEOM                                                         MAIN  123I
  110 CALL OVERLAY (5HMARK4,1,0,6HRECALL)                               MAIN  124C
      GO TO 150                                                         MAIN  125
C                                                                       MAIN  126
C 120 CALL AERO                                                         MAIN  127I
  120 CALL OVERLAY (5HMARK4,2,0,6HRECALL)                               MAIN  128C
      GO TO 150                                                         MAIN  129
C                                                                       MAIN  130
C                                                                       MAIN  131
  150 IF (ERROR .NE. 0) GO TO 200                                       MAIN  132
      WRITE (TAPEOT,160)                                                MAIN  133
  160 FORMAT (1H1,//////,1H0,40H******** MAIN PROGRAM NOW HAS CONTROL OF MAIN  134
     1 16H SYSTEM ******** )                                            MAIN  135
```

```
      IF (ERROR .EQ. 0) GO TO 100                                       MAIN  136
      IF (ERROR .EQ. 2) GO TO 200                                       MAIN  137
C                                                                       MAIN  138
C                                                                       MAIN  139
      WRITE (TAPEOT,170) I, ERROR                                       MAIN  140
  170 FORMAT (1H0,36H*****AN ERROR HAS OCCURRED IN PHASE ,I3,           MAIN  141
     1 61H AND WAS DETECTED AFTER RETURN TO THE MAIN EXECUTIVE ROUTINE.,MAIN  142
     2 9H  ERROR =,I2)                                                  MAIN  143
C                                                                       MAIN  144
  200 WRITE (TAPEOT,210)                                                MAIN  145
      WRITE(22,210)                                                     MAIN  146
  210 FORMAT (1H ,//,1H ,40H******** FATAL ERROR *** PROGRAM STOPPED )  MAIN  147
      GO TO 280                                                         MAIN  148
C                                                                       MAIN  149
C                                                                       MAIN  150
C                                                                       MAIN  151
  220 WRITE (TAPEOT,230)                                                MAIN  152
  230 FORMAT (1H ,//,1H ,38H***** FIRST PHASE OPTION IS ZERO *****      MAIN  153
     1 21H**** FATAL ERROR **** )                                       MAIN  154
      GO TO 280                                                         MAIN  155
C                                                                       MAIN  156
  240 WRITE (TAPEOT,250)                                                MAIN  157
  250 FORMAT (1H ,//,1H ,42H***** PROGRAM OPTION IS GREATER THAN 4 ***  MAIN  158
     1 22H***** FATAL ERROR ****  )                                     MAIN  159
      GO TO 280                                                         MAIN  160
C                                                                       MAIN  161
  260 WRITE (TAPEOT,270)                                                MAIN  162
      WRITE(22,270)                                                     MAIN  163
  270 FORMAT (1H ,//,1H ,44H***** PROGRAM HAS REACHED NORMAL TERMINATION MAIN  164
     1  7H ******  )                                                    MAIN  165
  280 STOP                                                              MAIN  166
      END                                                               MAIN  167
```

```
      OVERLAY (MARK4,2,0)                                               AERO  0010
      PROGRAM AERO                                                      AERO  0020
C     SUBROUTINE AERO                                                   AERO  0030
C                                                                       AERO  0040
C**** THIS ROUTINE CONTROLS THE AERODYNAMIC PROGRAM                     AERO  0050
C                                                                       AERO  0060
C                                                                       AERO  0070
      COMMON /FLIGHT/MACH,ALT,PSTAG,TSTAG,V,RENO,PES,TES,AES,RHOES,VIS  AERO  0080
      COMMON /GASD/GAM,GASCP,PRAN,IGAS,AV1,AV2,AV3,GTYPE(2)             AERO  0090
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR,INPUTD                         AERO  0100
      COMMON /ESBS/ES,BS                                                AERO  0110
      COMMON /TAGS/ITAG4,ITAG9,ITAG10                                   AERO  0120
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   AERO  0130
     1            TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                         AERO  0140
      COMMON /REF/SREF,MAC,SPAN,XCG,YCG,ZCG                             AERO  0150
      COMMON /ABDATA/NAB,ALPHA(20),BETA(20),POL(20),CDELTA(20),QI(20),  AERO  0160
     1               RI(20),PI(20)                                      AERO  0170
      COMMON/PLOTCP/IRCP,IRCT,NEXTCP                                    AERO  0180
      COMMON/ASSOCV/IA15,IA16                                           AERO  0190
      COMMON/INDATA/IRUNID,ICOPID,IPLTCP,IBETA,BETAIN,IWRIT,            AERO  0200
     1     ICKPRT,ISVFMS,ISVSPL                                         AERO  0210
      COMMON/SAVSPL/NEXPAN,NUMPAN,NUMCOR,NEXSPL,IPDP(72),IPSP(72),      AERO  0220
     1     IPAN(72),IPANL(10)                                           AERO  0230
      COMMON/SUBIDX/SIDX04(1001),SIDX16(1001),SIDX21(1001)              AERO  0240
      COMMON /MASTER/ MIDX04(21),INDEX9(501),INDX10(2001),INDX15(2001), AERO  0250
     1  MIDX16(94),INDX20(2421),MIDX21(73)                              AERO  0260
      DIMENSION IDUMS(36),ICID(36),ICPP(36),IBTA(36),IPSP1(36),IPSP2(36) AERO  0270
      DIMENSION  TITLE(15),IPG(20),ES(8),BS(8),EO(25)                   AERO  0280
      DIMENSION IDUMR(100),IDUM(8),ADUM(8),IRCP(200),IRCT(200),         AERO  0290
     1     ADUMR(100),TITLL5(15)                                        AERO  0300
      DIMENSION MDUM(21),BUF(100),NDUM(4),JDUM(100)                     AERO  0310
      EQUIVALENCE (IDUMR(1),IRCP(1)), (ADUMR(1),IRCT(1))                AERO  0320
      EQUIVALENCE (IPDP(1),IDUMS(1)), (IPSP(1),IPSP1(1)),               AERO  0330
     1            (IPSP(37),IPSP2(1))                                   AERO  0340
      EQUIVALENCE (NEXPAN,NDUM(1))                                      AERO  0350
      EQUIVALENCE (BUF(1),IDUM(1)), (BUF(9),ADUM(1))                    AERO  0360
      INTEGER  ERROR,PAGE                                               AERO  0370
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,       AERO  0380
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                            AERO  0390
      INTEGER SIDX04,SIDX16,SIDX21                                      AERO  0400
      REAL  MACH,MAC                                                    AERO  0410
C                                                                       AERO  0420
      DATA GAS1,GAS2,GAS3,GAS4/4H  *A,4HIR* ,4H*HEL,4HIUM*/             AERO  0430
      DATA JDUM /100*0/                                                 AERO  0440
C                                                                       AERO  0450
```

```
C   SPECIAL INPUT PROCEDURE FOR AERO LOADS STUDY
    5 CONTINUE
      ERROR = 0
      IF (INPUTD .NE. 0)  GO TO 12
C     READ(5,10,END=280) TITLE,ISVFMS,ISVSPL,
C    1      IRUNID,ICORID,ICKPRT,IPLTCP,IBETA,IWRIT,BETAIN
      READ(5,10) TITLE,ISVFMS,ISVSPL,
     1      IRUNID,ICORID,ICKPRT,IPLTCP,IBETA,IWRIT,BETAIN
      IF (EOF(5)) 280, 281
  281 CONTINUE
   10 FORMAT( 10A4, 5A1, 2I1, 3XI3, 2XI4, 4I1, F9.4)
      TAPEIN = 1
      REWIND TAPEIN
C     CALL INPUT
      CALL OVERLAY (5HMARK4,2,2,6HRECALL)
      IF (ERROR .NE. 0)  GO TO 5
      CALL MONITR
C
C   END OF SPECIAL INPUT PROCEDURE
C
C
C   READ AERO SYSTEM CONTROL CARD
C   12 READ(TAPEIN,10,END=280) TITLE,ISVFMS,ISVSPL,
C    1      IRUNID,ICORID,ICKPRT,IPLTCP,IBETA,IWRIT,BETAIN
   12 READ(TAPEIN,10) TITLE,ISVFMS,ISVSPL,
     1      IRUNID,ICORID,ICKPRT,IPLTCP,IBETA,IWRIT,BETAIN
      IF (EOF(TAPEIN)) 280, 282
  282 CONTINUE
      WRITE (TAPEOT,20) TITLE
      WRITE(22,20)  TITLE
   20  FORMAT (1HL,31H***** AERODYNAMIC PROGRAM *****,/1H0,15A4)
      WRITE(TAPEOT,15)  IRUNID,ICORID
      WRITE(22,15)  IRUNID,ICORID
   15 FORMAT(1H0, 5X,9HRUN ID = , I3, 5X17HCORRELATION ID = , I4)
C
C    SPECIAL OPTION TO SAVE CP ROW DATA FOR PLOTTING
C
C       IPLTCP = 0,  DO NOT SAVE
C
C              = 1,  SAVE NEW DATA ON EXISTING DATA SET
C
C              = 3,  GENERATE A NEW DATA SET
C
      IF (IPLTCP.EQ. 0)  GO TO 29
C
```

```
C   SPECIAL CP PLOT OPTION SPECIFIED,                                    AERO  091
C   DETERMINE RUN AND SEQUENCE NUMBERS.                                  AERO  092
      IA15 = 1                                                           AERO  093
C     READ (15'IA15) MDUM                                                AERO  094I
      CALL READMS (15,MDUM,21,IA15)                                      AERO  095C
      IF (IPLTCP.EQ.1) GO TO 28                                          AERO  096
      MDUM(18) = 0                                                       AERO  097
      MDUM(19) = 31                                                      AERO  098
      MDUM(20) = 131                                                     AERO  099
      MDUM(21) = 1                                                       AERO  100
      IA15 = 1                                                           AERO  101
C     WRITE (15'IA15) MDUM                                               AERO  102I
      CALL WRITMS (15,MDUM,21,IA15,1)                                    AERO  103C
      DO 27 I=21,1330                                                    AERO  104
C     WRITE (15'I) JDUM                                                  AERO  105I
      CALL WRITMS (15,JDUM,100,I,1)                                      AERO  106C
   27 CONTINUE                                                           AERO  107
   28 CONTINUE                                                           AERO  108
      NRUN=MDUM(18)                                                      AERO  109
      NEXTRN=MDUM(19)                                                    AERO  110
      NEXTRW=MDUM(20)                                                    AERO  111
      NEXTCP=MDUM(21)                                                    AERO  112
C                                                                        AERO  113
C   READ IN RUN ID,S                                                     AERO  114
      IA15 = 21                                                          AERO  115
C     READ(15'IA15)  IDUMR                                               AERO  116I
      CALL READMS (15,IDUMR,100,IA15)                                    AERO  117C
C                                                                        AERO  118
C   SEARCH RUN ID'S                                                      AERO  119
      DO 21 I = 1,100                                                    AERO  120
      IRUN = I                                                           AERO  121
      IF (IRUNID .EQ. IDUMR(I))  GO TO 22                                AERO  122
      IF (IDUMR(I) .EQ. 0)  GO TO 19                                     AERO  123
   21 CONTINUE                                                           AERO  124
C                                                                        AERO  125
C   NEW RUN                                                              AERO  126
   19 NRUN = NRUN + 1                                                    AERO  127
      ICORP = NEXTRN                                                     AERO  128
      NEXTRN = NEXTRN + 1                                                AERO  129
      GO TO 23                                                           AERO  130
C                                                                        AERO  131
C   RUN EXISTS,  READ POINTER TO CORRELATION DIRECTORY                   AERO  132
   22 IA15 = 23                                                          AERO  133
C     READ(15'IA15)  IDUMR                                               AERO  134I
      CALL READMS (15,IDUMR,100,IA15)                                    AERO  135C
```

```
      ICORP = IDUMR(IRUN)
C
C   READ CORRELATION DIRECTORY TABLE
   23 CONTINUE
C     READ (15'ICORP) BUF
      CALL READMS (15,BUF,100,ICORP)
C
C   DETERMINE CORRELATION SEQUENCE NUMBER
      DO 24 I = 1,4
      ICOR = I
      IF (IDUM(I) .EQ. 0)  GO TO 25
   24 CONTINUE
   25 CONTINUE
C
      WRITE(TAPEOT,16)  IRUN, ICOR
      WRITE(22,16)  IRUN,ICOR
   16 FORMAT(1H0,33HSPECIAL CP PLOT OPTION REQUESTED./
     1 1H0, 43HCORRESPONDING SEQUENCE NUMBERS ARE  IRUN = ,I3,
     2       5X7HICOR = , I4)
C
C   READ IN ROW POINTERS AND COUNTERS
      IRWP = IDUM(ICOR+4)
      IF (IRWP .NE. 0)  GO TO 26
      IRWP = NEXTRW
      NEXTRW = NEXTRW + 4
C
C   READ IN ROW DIRECTORY TABLE
   26 CONTINUE
C     READ (15'IRWP) IRCP
      CALL READMS (15,IRCP,100,IRWP)
      CALL READMS (15,IRCP(101),100,IRWP+1)
C     READ(15'IRWP+2)     IRCT
      CALL READMS (15,IRCT,100,IRWP+2)
      CALL READMS (15,IRCT(101),100,IRWP+3)
C
   29 CONTINUE
C   SPECIAL OPTION TO SAVE SPLINE DATA
C     ISVSPL = 0,  DO NOT SAVE
C
C            = 1,  SAVE NEW DATA ON EXISTING DATA SET
C
C            = 2,  REPLACE OLD DATA ON EXISTING DATA SET
C
C            = 3,  GENERATE A NEW DATA SET
C
```

```
      IF (ISVSPL .EQ. 0)  GO TO 500                                     AERO  181
      IF (ISVSPL .NE. 3)  GO TO 320                                     AERO  182
C                                                                       AERO  183
C  GENERATE NEW DATA SET                                                AERO  184
      DO 300 I = 1,72                                                   AERO  185
      IPAN(I) = 0                                                       AERO  186
  300 IPDP(I) = 0                                                       AERO  187
C                                                                       AERO  188
C  ZERO OUT CORRELATION DIRECTORY                                       AERO  189
      NSUB21=1                                                          AERO  190C
      CALL STINDX (21,MIDX21,73)                                        AERO  191C
      CALL READMS (21,SIDX21,1001,NSUB21)                               AERO  192C
      CALL STINDX (21,SIDX21,1001)                                      AERO  193C
      DO 310 I = 150,254                                                AERO  194
      IG21 = I                                                          AERO  195
C     WRITE(21'IG21)  IDUMS                                             AERO  196I
      NSRC21=IG21                                                       AERO  197C
      CALL WRITMS (21,IDUMS,36,NSRC21,0)                                AERO  198C
  310 CONTINUE                                                          AERO  199
C                                                                       AERO  200
      NEXPAN  = 6                                                       AERO  201
      NUMPAN  = 0                                                       AERO  202
      NUMCOR  = 0                                                       AERO  203
      NEXSPL  = 255                                                     AERO  204
      NSRC21=1                                                          AERO  205C
      CALL WRITMS (21,NDUM,4,NSRC21,0)                                  AERO  206C
      NSRC21=2                                                          AERO  207C
      CALL WRITMS (21,IPAN,72,NSRC21,0)                                 AERO  208C
      NSRC21=4                                                          AERO  209C
      CALL WRITMS (21,IPDP,72,NSRC21,0)                                 AERO  210C
      GO TO 330                                                         AERO  211
C                                                                       AERO  212
C  USE EXISTING DATA SET                                                AERO  213
  320 IG21 = 1                                                          AERO  214
      NSUB21=1                                                          AERO  215C
      NSRC21=1                                                          AERO  216C
C     READ (21'IG21) NDUM                                               AERO  217I
C     READ(21'IG21+1)  IPAN                                             AERO  218I
C     READ(21'IG21+3)  IPDP                                             AERO  219I
      CALL STINDX (21,MIDX21,73)                                        AERO  220C
      CALL READMS (21,SIDX21,1001,NSUB21)                               AERO  221C
      CALL STINDX (21,SIDX21,1001)                                      AERO  222C
      CALL READMS (21,NDUM,4,NSRC21)                                    AERO  223C
      CALL READMS (21,IPAN,72,NSRC21+1)                                 AERO  224C
      CALL READMS (21,IPDP,72,NSRC21+3)                                 AERO  225C
```

```
C
  330 IF (ISVSPL .EQ. 2)  GO TO 350
      NUMCOR = NUMCOR + 1
      IF (NUMCOR .GT. 1260)  GO TO 380
      ICOREC = 150 + (NUMCOR-1)/36
      ICOSEQ = NUMCOR - (ICOREC-150)*36
C     READ(21'ICOREC)  ICID
C     READ(21'ICOREC+35)  ICPP
C     READ(21'ICOREC+70)  IBTA
      CALL READMS (21,ICID,36,ICOREC)
      CALL READMS (21,ICPP,36,ICOREC+35)
      CALL READMS (21,IBTA,36,ICOREC+70)
      GO TO 390
C
C  REPLACE EXISTING DATA
  350 CONTINUE
      DO 370 I = 150,184
      ICOREC = I
      NSRC21=I
C     READ(21'ICOREC)  ICID
      CALL READMS (21,ICID,36,NSRC21)
      DO 360 J = 1,36
      ICOSEQ = J
      IF (ICID(J) .EQ. ICORID)  GO TO 400
      IF (ICID(J) .EQ. 0)  GO TO 390
  360 CONTINUE
  370 CONTINUE
C
C  CORRELATION ID COULD NOT BE FOUND AND ARRAY IS FULL.
C  TURN OFF SAVE FLAG
  380 ISVSPL = 0
      GO TO 500
C
C  CORRELATION ID COULD NOT BE FOUND.  TREAT AS NEW
  390 ICID(ICOSEQ) = ICORID
      ISVSPL = 1
      ICPPS = NEXSPL
      NEXSPL = NEXSPL + 1
      ICPP(ICOSEQ) = ICPPS
      DO 395 I = 1,36
      IPSP1(I) = 0
  395 IPSP2(I) = 0
      NSUB=(ICPPS-1)/1000+1
      NSRC21=ICPPS-(NSUB-1)*1000
      IF (NSUB.EQ.NSUB21) GO TO 396
```

```
      CALL STINDX (21,MIDX21,73)                                        AERO  271C
      CALL WRITMS (21,SIDX21,1001,NSUB21,1)                             AERO  272C
      CALL READMS (21,SIDX21,1001,NSUB)                                 AERO  273C
      CALL STINDX (21,SIDX21,1001)                                      AERO  274C
      NSUB21=NSUB                                                       AERO  275C
  396 CALL WRITMS (21,IPSP1,36,NSRC21,0)                                AERO  276C
      IF (IBETA.EQ.0) GO TO 450                                         AERO  277
      NEXSPL = NEXSPL + 1                                               AERO  278
      IBTA(ICOSEQ) = 1                                                  AERO  279
      NSUB=ICPPS/1000+1                                                 AERO  280C
      NSRC21=ICPPS+1-(NSUB-1)*1000                                      AERO  281C
      IF (NSUB.EQ.NSUB21) GO TO 397                                     AERO  282C
      CALL STINDX (21,MIDX21,73)                                        AERO  283C
      CALL WRITMS (21,SIDX21,1001,NSUB21,1)                             AERO  284C
      CALL READMS (21,SIDX21,1001,NSUB)                                 AERO  285C
      CALL STINDX (21,SIDX21,1001)                                      AERO  286C
      NSUB21=NSUB                                                       AERO  287C
  397 CALL WRITMS (21,IPSP2,36,NSRC21,0)                                AERO  288C
      GO TO 450                                                         AERO  289
C                                                                       AERO  290
C                                                                       AERO  291
  400 IG21 = ICOREC + 35                                                AERO  292
C     READ(21'IG21)  ICPP                                               AERO  293I
      NSRC21=IG21                                                       AERO  294C
      CALL READMS (21,ICPP,36,NSRC21)                                   AERO  295C
      ICPPS = ICPP(ICOSEQ)                                              AERO  296
C     READ(21'ICPPS)  IPSP1                                             AERO  297I
      NSUB=(ICPPS-1)/1000+1                                             AERO  298C
      NSRC21=ICPPS-(NSUB-1)*1000                                        AERO  299C
      IF (NSUB.EQ.NSUB21) GO TO 3                                       AERO  300C
      CALL STINDX (21,MIDX21,73)                                        AERO  301C
      CALL READMS (21,SIDX21,1001,NSUB)                                 AERO  302C
      CALL STINDX (21,SIDX21,1001)                                      AERO  303C
      NSUB21=NSUB                                                       AERO  304C
    3 CALL READMS (21,IPSP1,36,NSRC21)                                  AERO  305C
      IF (IBETA .EQ. 0)  GO TO 500                                      AERO  306
C     READ(21'ICPPS+1)  IPSP2                                           AERO  307I
      NSUB=ICPPS/1000+1                                                 AERO  308C
      NSRC21=ICPPS+1-(NSUB-1)*1000                                      AERO  309C
      IF (NSUB.EQ.NSUB21) GO TO 4                                       AERO  310C
      CALL STINDX (21,MIDX21,73)                                        AERO  311C
      CALL READMS (21,SIDX21,1001,NSUB)                                 AERO  312C
      CALL STINDX (21,SIDX21,1001)                                      AERO  313C
      NSUB21=NSUB                                                       AERO  314C
    4 CALL READMS (21,IPSP2,36,NSRC21)                                  AERO  315C
```

```
C
  450 CONTINUE
      CALL STINDX (21,MIDX21,73)
      CALL WRITMS (21,SIDX21,1001,NSUB21,1)
C
  500 CONTINUE
C
C
C  READ AERO FLAG CARD
C     READ (TAPEIN,30) IPG
C   30  FORMAT (20I1)
      READ (TAPEIN,30) IPG, NTAPE, IREP
   30 FORMAT ( 20I1, 5X, I2, I1)
      IF (IREP.NE.0) GO TO 31
      NSUB=1
      CALL STINDX (4,MIDX04,21)
      CALL READMS (4,SIDX04,1001,NSUB)
      CALL STINDX (4,SIDX04,1001)
      NSUB4=NSUB
      IG4=3
C     READ (4'IG4) EO
      NSREC4=3
      CALL READMS (4,EO,25,NSREC4)
      NEXT=EO(1)
      IG4=2
C     READ (4'IG4) EO
      NSREC4=2
      CALL READMS (4,EO,25,NSREC4)
      EO(1)=NEXT
C     WRITE (4'IG4) EO
      CALL WRITMS (4,EO,25,NSREC4,1)
      CALL STINDX (4,MIDX04,21)
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)
   31 CONTINUE
C
      DO 80 I=1,20
       IF (IPG(I) .EQ. 0) GO TO 90
       IF (IPG(I) .EQ. 1) WRITE (TAPEOT,40) I
   40 FORMAT (1H0,15X,I2,31H FLOW FIELD ANALYSIS (OPTION 1))
       IF (IPG(I) .EQ. 3) WRITE (TAPEOT,60) I
   60   FORMAT (1H0,15X,I2,31H INVISCID PRESSURES  (OPTION 3) )
      IF (IPG(I) .EQ. 5) WRITE (TAPEOT,71) I
   71   FORMAT (1H0,15X,I2,31H SPECIAL ROUTINES    (OPTION 5) )
      IF (IPG(I) .EQ. 7)   WRITE(TAPEOT,73)  I
   73 FORMAT(1H0,15X,I2,31H SHEARS AND MOMENTS  (OPTION 7) )
```

```
   80 CONTINUE
C
C  READ FLIGHT CONDITION CARD
   85 CONTINUE
   90 READ(TAPEIN,100) MACH,ALT,PSTAG,TSTAG,IGAS,NAB,AMBP
  100 FORMAT(4F10.0, I1, I2, 7XF10.0)
C  READ REFERENCE DIMENSION CARD
      READ (TAPEIN,110) SREF,MAC,SPAN,XCG,YCG,ZCG
  110  FORMAT (6F10.0)
      DO 130 I=1,NAB
      READ (TAPEIN,120) ALPHA(I),BETA(I),RCL(I),CDELTA(I),QI(I),RI(I),
     1                  PI(I)
  120  FORMAT (7F10.0)
  130 CONTINUE
C
C  SET GAS  CONSTANTS
      IF (IGAS .EQ. 1) GO TO 140
C****AIR****
      GAM = 1.40
      GASCP = 6.00793E+3
      PRAN = 0.71
      AV1 = 225.0
      AV2 = 8.0382436E-10
      AV3 = 1.0
      GTYPE(1) = GAS1
      GTYPE(2) = GAS2
      GO TO 150
C
C****HELIUM****
  140 GAM = 5./3.
      GASCP = 3.1250E+4
      PRAN = 2./3.
      AV1 = 1.0E+5
      AV2 = 7.17884E-9
      AV3 = 0.647
      GTYPE(1) = GAS3
      GTYPE(2) = GAS4
C
  150  CONTINUE
      RHOFS=0.002377
      PFS = 2116.231
      TFS =  518.67
      AFS = 1116.447
      IF(TFS.GE.AV1) VIS = 2.27E-8*TFS**1.5/(TFS + 198.6)
      IF(TFS.LT.AV1) VIS = AV2*TFS**AV3
```

```
      V = MACH * AFS
      RENC = RHOFS * V / VIS
C   INITIALIZE FREE STREAM DATA ARRAY
      FS(1) = RHOFS
      FS(2) = PFS
      FS(3) = TFS
      FS(4) = AFS
      FS(5) = VIS
      FS(6) = MACH
      FS(7) = MACH * FS(4)
      FS(8) = FS(1)*FS(7)/FS(5)
C
      J = 0
  200 J = J + 1
      IF (IPG(J) .EQ. 0) GO TO 260
      IPROG = IPG(J)
C
      GO TO (210,220,230,240,241,243,245), IPROG
C
C 210 CALL FLOW
  210 CALL OVERLAY (5HMARK4,2,1,6HRECALL)
       GO TO 250
  220 WRITE (TAPEOT,410)
      WRITE (22,410)
  410 FORMAT (1H0, 51H*** SHIELD IS NOT AVAILABLE IN THE STRIPPED VERSIO
     1N)
      ERROR=2
       GO TO 250
C 230 CALL PRES
  230 CALL OVERLAY (5HMARK4,2,3,6HRECALL)
C
C
C
      IF (ISVSPL .EQ. 0)  GO TO 235
C
C   RESET MASTER DIRECTORY
      IG21 = 1
C     WRITE (21'IG21) NDUM
      NSUB=1
      NSRC21=1
      CALL STINDX (21,MIDX21,73)
      CALL READMS (21,SIDX21,1001,NSUB)
      CALL STINDX (21,SIDX21,1001)
      NSUB21 = NSUB
    6 CALL WRITMS (21,NDUM,4,NSRC21,1)
```

```
C                                                                       AERO  451
C   RESET PANEL DIRECTORY                                               AERO  452
C      WRITE(21'IG21+1)  IPAN                                           AERO  453I
C      WRITE(21'IG21+3)  IPDP                                           AERO  454I
       CALL WRITMS (21,IPAN,72,NSRC21+1,1)                              AERO  455C
       CALL WRITMS (21,IPDP,72,NSRC21+3,1)                              AERO  456C
C                                                                       AERO  457
C   RESET CORRELATION DIRECTORY                                         AERO  458
C      WRITE(21'ICOREC)  ICID                                           AERO  459I
       CALL WRITMS (21,ICID,36,ICOREC,1)                                AERO  460C
C      WRITE(21'ICOREC+35)  ICPP                                        AERO  461I
       CALL WRITMS (21,ICPP,36,ICOREC+35,1)                             AERO  462C
C      WRITE(21'ICOREC+70)  IBTA                                        AERO  463I
       CALL WRITMS (21,IBTA,36,ICOREC+70,1)                             AERO  464C
C                                                                       AERO  465
C   SET CORRELATION PANEL DIRECTORY                                     AERO  466
C      WRITE(21'ICPPS)  IPSP1                                           AERO  467I
       NSUB=(ICPPS-1)/1000+1                                            AERO  468C
       NSRC21=ICPPS-(NSUB-1)*1000                                       AERO  469C
       IF (NSUB.EQ.NSUB21) GO TO 11                                     AERO  470C
       CALL STINDX (21,MIDX21,73)                                       AERO  471C
       CALL WRITMS (21,SIDX21,1001,NSUB21,1)                            AERO  472C
       CALL READMS (21,SIDX21,1001,NSUB)                                AERO  473C
       CALL STINDX (21,SIDX21,1001)                                     AERO  474C
       NSUB21=NSUB                                                      AERO  475C
    11 CALL WRITMS (21,IPSP1,36,NSRC21,1)                               AERO  476C
       IF (IBETA .EQ. 0)  GO TO 233                                     AERO  477
C      WRITE(21'ICPPS+1)  IPSP2                                         AERO  478I
       NSUB=ICPPS/1000+1                                                AERO  479C
       NSRC21=ICPPS+1-(NSUB-1)*1000                                     AERO  480C
       IF (NSUB.EQ.NSUB21) GO TO 13                                     AERO  481C
       CALL STINDX (21,MIDX21,73)                                       AERO  482C
       CALL WRITMS (21,SIDX21,1001,NSUB21,1)                            AERO  483C
       CALL READMS (21,SIDX21,1001,NSUB)                                AERO  484C
       CALL STINDX (21,SIDX21,1001)                                     AERO  485C
       NSUB21=NSUB                                                      AERO  486C
    13 CALL WRITMS (21,IPSP2,36,NSRC21,1)                               AERO  487C
C                                                                       AERO  488
   233 CONTINUE                                                         AERO  489
       CALL STINDX (21,MIDX21,73)                                       AERO  490C
       CALL WRITMS (21,SIDX21,1001,NSUB21,1)                            AERO  491C
       NSUB21=0                                                         AERO  492C
       WRITE (TAPEOT,234)  IRUNID, ICORID                               AERO  493
       WRITE(22,234) IRUNID,ICORID                                      AERO  494
   234 FORMAT(1H0,38HDIRECTORIES ON UNIT 21 HAVE BEEN RESET/            AERO  495
```

```
     1        1H0,25HSPLINE DATA FOR RUN ID = ,I3,17H, CORRELATION ID ,
     2        I4, 16H HAS BEEN SAVED.)
C
C
  235 CONTINUE
      IF (IPLTCP.EQ. 0)  GO TC 250
C
C   SAVE ROW POINTERS AND COUNTERS
C     WRITE(15'IRWP)      IRCP
      CALL WRITMS (15,IRCP,100,IRWP,1)
      CALL WRITMS (15,IRCP(101),100,IRWP+1,1)
C     WRITE(15'IRWP+2)    IRCT
      CALL WRITMS (15,IRCT,100,IRWP+2,1)
      CALL WRITMS (15,IRCT(101),100,IRWP+3,1)
C
C   RESET CORRELATION DIRECTORY TABLE
      IDUM(ICOR)        =  ICORID
      IDUM(ICOR + 4)    =  IRWP
      ADUM(ICOR)        =  ALPHA(1)
      ADUM(ICOR + 4)    =  BETA(1)
C     WRITE (15'ICORP) BUF
      CALL WRITMS (15,BUF,100,ICORP,1)
C
C   RESET RUN DIRECTORY TABLE
C   RUN ID
      IG15 = 21
C     READ(15'IG15) IDUMR
      CALL READMS (15,IDUMR,100,IG15)
      IDUMR(IRUN) = IRUNID
C     WRITE(15'IG15)  IDUMR
      CALL WRITMS (15,IDUMR,100,IG15,1)
C
C   NUMBER OF CORRELATIONS
C     READ(15'IG15+1)  IDUMR
      CALL READMS (15,IDUMR,100,IG15+1)
      IDUMR(IRUN) = ICOR
C     WRITE(15'IG15 +1) IDUMR
      CALL WRITMS (15,IDUMR,100,IG15+1,1)
C
C   POINTER TO CORRELATION DIRECTORY
C     READ(15'IG15+2)  IDUMR
      CALL READMS (15,IDUMR,100,IG15+2)
      IDUMR(IRUN) = ICORP
C     WRITE(15'IG15+2) IDUMR
      CALL WRITMS (15,IDUMR,100,IG15+2,1)
```

```
C                                                                       AERO  541
C   RUN MACH NUMBER                                                     AERO  542
C      READ(15'IG15+3)  ADUMR                                           AERO  543I
       CALL READMS (15,ADUMR,100,IG15+3)                                AERO  544C
       ADUMR(IRUN) = MACH                                               AERO  545
C      WRITE(15'IG15+3) ADUMR                                           AERO  546I
       CALL WRITMS (15,ADUMR,100,IG15+3,1)                              AERO  547C
C                                                                       AERO  548
C   MASS FLOW                                                           AERO  549
C      READ(15'IG15+4)  ADUMR                                           AERO  550I
       CALL READMS (15,ADUMR,100,IG15+4)                                AERO  551C
       ADUMR(IRUN) = AMBP                                               AERO  552
C      WRITE(15'IG15+4) ADUMR                                           AERO  553I
       CALL WRITMS (15,ADUMR,100,IG15+4,1)                              AERO  554C
C                                                                       AERO  555
C   RESET MASTER CONTROLS                                               AERO  556
       MDUM(18)=NRUN                                                    AERO  557
       MDUM(19)=NEXTRN                                                  AERO  558
       MDUM(20)=NEXTRW                                                  AERO  559
       MDUM(21)=NEXTCP                                                  AERO  560
       IG15 = 1                                                         AERO  561
C      WRITE (15'IG15) MDUM                                             AERO  562I
       CALL WRITMS (15,MDUM,21,IG15,1)                                  AERO  563C
C                                                                       AERO  564
C                                                                       AERO  565
       WRITE(TAPEOT,275)  IRUN, ICOR                                    AERO  566
       WRITE(22,275) IRUN,ICOR                                          AERO  567
  275 FORMAT(1H0,38HDIRECTORIES ON UNIT 15 HAVE BEEN RESET/             AERO  568
     1        1H0, 4HRUN ,I3,  14H, CORRELATION , I2, 13H IS COMPLETE.) AERO  569
C                                                                       AERO  570
        GO TO 250                                                       AERO  571
  240 WRITE (TAPEOT,420)                                                AERO  572
       WRITE (22,420)                                                   AERO  573
  420 FORMAT (1H0, 51H*** VISCUS IS NOT AVAILABLE IN THE STRIPPED VERSIO AERO  574
     1N)                                                                AERO  575
       ERROR=2                                                          AERO  576
       GO TO 250                                                        AERO  577
C 241 CALL SPEC                                                         AERO  578I
  241 CALL OVERLAY (5HMARK4,2,5,6HRECALL)                               AERO  579C
       GO TO 250                                                        AERO  580
C                                                                       AERO  581
  243 WRITE (TAPEOT,430)                                                AERO  582
       WRITE (22,430)                                                   AERO  583
  430 FORMAT (1H0, 51H*** STREAM IS NOT AVAILABLE IN THE STRIPPED VERSIO AERO  584
     1N)                                                                AERO  585
```

```
      ERROR=2                                                           AERO  586
      GO TO 250                                                         AERO  587
C 245 CALL SHEAR                                                        AERO  588I
  245 CALL OVERLAY (5HMARK4,2,7,6HRECALL)                               AERO  589C
C                                                                       AERO  590
  250 IF (ERROR .EQ. 0) GO TO 200                                       AERO  591
      IF (ERROR.EQ.2) GO TO 280                                         AERO  592
      GO TO 5                                                           AERO  593
  260 CONTINUE                                                          AERO  594
C                                                                       AERO  595
      WRITE (TAPEOT,270)                                                AERO  596
      WRITE(22,270)                                                     AERO  597
  270  FORMAT (1H0,37H *** AERODYNAMIC ANALYSIS IS COMPLETE   )         AERO  598
      IF (INPUTD .EQ. 0)  GO TO 5                                       AERO  599
C                                                                       AERO  600
C                                                                       AERO  601
C 280 RETURN                                                            AERO  602I
  280 CONTINUE                                                          AERO  603C
      END                                                               AERO  604
```

```
      OVERLAY (MARK4,1,4)                                               AIRC  001C
      PROGRAM AIRCFT                                                    AIRC  002C
C     SUBROUTINE AIRCFT                                                 AIRC  003I
C                                                                       AIRC  004
C*****************************************************************      AIRC  005
C*******MAIN PROGRAM FOR AIRCRAFT TYPE VEHICLE GEOMETRY GENERATION      AIRC  006
C*******OPTION.  THIS OPTION MAY ALSO BE USED TO CONVERT GEOMETRY       AIRC  007
C*******DATA FOR THE NASA (HARRIS) WAVE DRAG PROGRAM TO ELEMENT         AIRC  008
C*******DATA FORMAT FOR USE BY THE HYPERSONIC ARBITRARY-BODY            AIRC  009
C*******PROGRAM OR BY THE ARBITRARY-BODY GRAPHICS PROGRAMS.             AIRC  010
C*****************************************************************      AIRC  011
C                                                                       AIRC  012
C        *** THIS PROGRAM WRITTEN BY A.E. GENTRY AND D.N. SMYTH ***     AIRC  013
C            DOUGLAS AERO RESEARCH GROUP, LONG BEACH                    AIRC  014
C                                                                       AIRC  015
C    NOTE- THE INPUT DATA MUST BE EXACTLY AS REQUIRED FOR THE           AIRC  016
C          CONFIGURATION DATA FOR THE HARRIS PROGRAM (INCLUDING         AIRC  017
C          THE IDENTIFICATION CARD).  MULTIPLE CONFIGURATIONS           AIRC  018
C          CAN BE STACKED ONE BEHIND THE OTHER.                         AIRC  019
C                                                                       AIRC  020
C                                                                       AIRC  021
      DIMENSION                                                         AIRC  022
     1     XAF(30),WAFORG(20,4),TZORD(20,30),WAFORD(20,30),             AIRC  023
     2     NRADX(4),NFORX(4),XFUS(4,30),ZFUS(4,30),FUSARD(4,30),        AIRC  024
     3          YFUSY(4,30,30),ZFUSZ(4,30,30),                          AIRC  025
     4     PODORG(9,6),XPOD(9,30),PODR(9,30),IOR(9),IEL(9),NEL(9),      AIRC  026
     5     FINORG(6,2,4),XFIN(6,10),FINORD(6,10),                       AIRC  027
     6     CANORG(2,2,4),XCAN(2,10),CANORD(2,10),CANOR1(2,10),          AIRC  028
     7     XOC(30),AFORG(20,7),AFCAM(20,30),AFORD(20,30),               AIRC  029
     8     CARD(21),SURF2(6),SURF3(2)                                   AIRC  030
C                                                                       AIRC  031
      COMMON /ACFT/IPRINT,ITAPE,ITYPE,PI,DX,CASE                        AIRC  032
C                                                                       AIRC  033
      COMMON /AREOIL/ NAF,NAFORD,NCAM,NACA,ITIP,IROOT,                  AIRC  034
     1                XOC,AFORG,AFCAM,AFORD,SURFID                      AIRC  035
     2               ,ISURF,IFLAP,NHNGE                                 AIRC  036
C                                                                       AIRC  037
      COMMON /FUSD/J2,J6,NFUS,NRADX,NFORX,XFUS,ZFUS,YFUSY,ZFUSZ,FUSARD  AIRC  038
C                                                                       AIRC  039
      COMMON /NACELD/NP,NPODOR,PODORG,XPOD,PODR,IOR,IEL,NEL             AIRC  040
C                                                                       AIRC  041
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   AIRC  042
     1          TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                           AIRC  043
      COMMON /GFLAG/IOUT,ISTAT,IORIEN,COMPIN                            AIRC  044
C                                                                       AIRC  045
```

```
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      INTEGER CASE
C
      DATA TYCH/2H99/
      DATA SURF1/4HWING/,
     1     SURF2(1),SURF2(2),SURF2(3),SURF2(4),SURF2(5),SURF2(6)/
     2     4HFIN1,4HFIN2,4HFIN3,4HFIN4,4HFIN5,4HFIN6/,
     3     SURF3(1),SURF3(2)/4HCAN1,4HCAN2/
      IPRINT = 1
C
      ITAPE = IOUT
      ITYPE  = 3
      PI = 3.1415927
C
C
C**PROGRAM OPERATION STARTS HERE----
C
C   READ IDENTIFICATION CARD, CONTROL CARD, AND REFERENCE AREA
C   CARD (IF REQUIRED)
   10 READ (TAPEIN,20) (CARD(I),I=1,17),ISTAT3,IHARIS, CASE,
     1                 (CARD(I), I=19,21)
   20 FORMAT (9A3,8A4,I1,2XI1,1XI2,4XA2,2A4)
      IF (CARD(19) .EQ. TYCH) GO TO 780
C   CHECK FOR NASA-HARRIS INPUT COORDINATES.
      DX = 1.0
      IF (IHARIS .EQ. 1) DX = -1.0
C
      WRITE (TAPEOT,30) (CARD(I),I=1,17), CASE
   30 FORMAT (1H1,48H*** OUTPUT GEOMETRY DATA FROM AIRCRAFT GEOMETRY ,
     111HOPTION ****,/1H0,5X,13HCONFIGURATION,5X,9A3,8A4,6X,5HCASE ,I2,
     2 //,6X,1HX,9X,1HY,9X,1HZ,4X,1HS,5X,1HX,9X,1HY,9X,1HZ,4X,1HS,/1H0)
      READ (TAPEIN,40) J0,J1,J2,J3,J4,J5,J6,J7,NWAF,NWAFOR,NFUS,
     1     (NRADX(I),NFORX(I),I=1,4),NP,NPODOR,NF,NFINOR,NCAN,NCANOR
   40 FORMAT (7I3,I1,I2,16I3,8X)
      IF (J0 .EQ. 1) READ (TAPEIN,50) REFA
   50 FORMAT (F7.2,73X)
C
C***CONTROL FLAG CARD DATA CHECK
      IF(IABS(J1) .NE. 1) GO TO 60
      I = 9
      IF (NWAF .LT. 2 .OR. NWAF .GT. 20)  GO TO 750
      I = 10
      IF (NWAFOR .LT. 3 .OR. NWAFOR .GT. 30)  GO TO 750
   60 IF(IABS(J2) .NE. 1) GO TO 80
```

```
      I = 11
      IF (NFUS .LT. 1 .OR. NFUS .GT. 4) GO TO 750
      DO 70 J = 1,NFUS
      I = I + 1
      IF (NRADX(J) .LT. 3 .OR. NRADX(J) .GT. 30) GO TO 750
      I = I + 1
      IF (NFORX(J) .LT. 2 .OR. NFORX(J) .GT. 30) GO TO 750
   70 CONTINUE
   80 IF (J3 .NE. 1) GO TO 90
      I = 20
      IF (NP .LT. 1 .OR. NP .GT. 9) GO TO 750
      I = 21
      IF (NPODOR .LT. 2 .OR. NPODOR .GT. 30) GO TO 750
   90 IF (J4 .NE. 1) GO TO 100
      I = 22
      IF (NF .LT. 1 .OR. NF .GT. 6) GO TO 750
      I = 23
      IF (NFINOR .LT. 3 .OR. NFINOR .GT. 10) GO TO 750
  100 IF (J5 .NE. 1) GO TO 110
      I = 24
      IF (NCAN .LT. 1 .OR. NCAN .GT. 2) GO TO 750
      I = 25
      N = IABS(NCANOR)
      IF (N .LT. 3 .OR. N .GT. 10) GO TO 750
  110 CONTINUE
C
C***END OF CONTROL FLAG DATA CHECK
C
C
C
C**WING**
C********
      IF (J1.EQ.0 .OR. J1.EQ.2) GO TO 260
C     READ PERCENT-CHORD LOCATIONS
      READ (TAPEIN,120) (XAF(I),I=1,NWAFOR)
  120 FORMAT (10F7.0,10X)
C    READ LOCATION OF CHORD LEADING EDGES AND LENGTH
      DO 140 I=1,NWAF
      READ (TAPEIN,130) (WAFORG(I,J),J=1,4)
  130 FORMAT (4F7.0,52X)
  140 CONTINUE
C    READ IN CAMBER LINE DATA
      NCAM = 0
      IF (J1 .EQ.-1) GO TO 170
      NCAM = 1
```

```
      DO 160 J=1,NWAF
      READ (TAPEIN,150) (TZORD(J,I),I=1,NWAFOR)
  150 FORMAT (10F7.0,10X)
  160 CONTINUE
      GO TO 200
C     CAMBER IS NOT INPUT. ZERO OUT VALUES
  170 DO 190 J=1,NWAF
      DO 180 I=1,NWAFOR
       TZORD(J,I) = 0.0
  180  CONTINUE
  190  CONTINUE
C
C
C   READ IN AIRFOIL ORDINATE DATA
  200 DO 210 J=1,NWAF
      READ (TAPEIN,150) (WAFORD(J,I),I=1,NWAFOR)
  210 CONTINUE
C
C  INITIALIZE DATA FOR WING SURFACE GENERATION
      NAF = NWAF
      NAFORD = NWAFOR
      SURFID = SURF1
      DO 240 I = 1,NAF
      DO 220 J = 1,3
  220 AFORG(I,J) = WAFORG(I,J)
      AFORG(I,4) = AFORG(I,1) - WAFORG(I,4)*DX
      AFORG(I,5) = WAFORG(I,2)
      AFORG(I,6) = WAFORG(I,3)
      AFORG(I,7) = 0.0
C
      DO 230 K = 1,NAFORD
      AFCAM(I,K) = TZORD(I,K)*100./WAFORG(I,4)
  230 AFORD(I,K) = WAFORD(I,K)
  240 CONTINUE
      DO 250 K = 1,NAFORD
  250 XOC(K) = XAF(K)
C
      NACA = 0
      ITIP = 0
      IROOT = 0
      ISURF = 0
      IFLAP = 0
C
      CALL SURF
C
```

```
C                                                                       AIRC  181
C**FUSELAGE**                                                           AIRC  182
C************                                                           AIRC  183
  260 IF (J2.EQ.0 .OR. J2.EQ.2) GO TO 360                               AIRC  184
C                                                                       AIRC  185
C    START CYCLE ON FUSELAGE SEGMENTS (NFUS TIMES)                      AIRC  186
      DO 350 I=1,NFUS                                                   AIRC  187
C                                                                       AIRC  188
      NFOX = NFORX(I)                                                   AIRC  189
      NRAX = NRADX(I)                                                   AIRC  190
C    READ X-VALUES OF FUSELAGE STATIONS FOR A SEGMENT                   AIRC  191
      READ (TAPEIN,270) (XFUS(I,J),J=1,NFOX)                            AIRC  192
  270 FORMAT (10F7.0,10X)                                               AIRC  193
C                                                                       AIRC  194
C    CHECK CIRCULAR AND CAMBER FLAGS                                    AIRC  195
      IF (J2 .EQ. 1 .AND. J6 .NE. 2) GO TO 300                          AIRC  196
C                                                                       AIRC  197
C    ZERO OUT CAMBER WHEN NOT REQUIRED                                  AIRC  198
  280 DO 290 J=1,NFOX                                                   AIRC  199
  290 ZFUS(I,J) = 0.0                                                   AIRC  200
      IF (J2 .EQ. 1) GO TO 310                                          AIRC  201
      IF (J2.EQ.-1 .AND. J6.NE.0) GO TO 330                             AIRC  202
      IF (J2.EQ.1  .AND. J6.EQ.1) GO TO 330                             AIRC  203
C                                                                       AIRC  204
C    READ FUSELAGE CAMBER DATA                                          AIRC  205
  300 READ (TAPEIN,270) (ZFUS(I,J),J=1,NFOX)                            AIRC  206
      IF (J2 .EQ. -1) GO TO 330                                         AIRC  207
C                                                                       AIRC  208
C    READ IN FUSELAGE CROSS-SECTION COORDINATES                         AIRC  209
  310 DO 320 J=1,NFOX                                                   AIRC  210
      READ (TAPEIN,270) (YFUSY(I,J,N),N=1,NRAX)                         AIRC  211
      READ (TAPEIN,270) (ZFUSZ(I,J,N),N=1,NRAX)                         AIRC  212
  320 CONTINUE                                                          AIRC  213
      GO TO 340                                                         AIRC  214
C                                                                       AIRC  215
C    READ IN FUSELAGE AREAS WHEN CIRCULAR                               AIRC  216
  330 READ (TAPEIN,270) (FUSARD(I,J),J=1,NFOX)                          AIRC  217
C                                                                       AIRC  218
C                                                                       AIRC  219
  340 CONTINUE                                                          AIRC  220
C  END OF FUSELAGE SEGMENT DO-LOOP                                      AIRC  221
  350 CONTINUE                                                          AIRC  222
C                                                                       AIRC  223
C  CALL FUSELAGE SUBROUTINE                                             AIRC  224
      CALL FUSE                                                         AIRC  225
```

```
C
      IFLAP = 0
C
C**POD**
C*******
  360 IF (J3.EQ.0 .OR. J3.EQ.2) GO TO 400
C
      DO 390 I=1,NP
C     READ POD ORIGIN DATA
      READ (TAPEIN,370) (PODORG(I,J),J=1,3)
     1      ,IOR(I),(PODORG(I,J),J=4,6),IEL(I),NEL(I)
  370 FORMAT(3F7.0,3XI1,3X3F7.0,6XI1,5XI2)
C     READ X-ORDINATES FOR POD RADII
      READ (TAPEIN,380) (XPOD(I,J),J=1,NPODOR)
  380 FORMAT (10F7.0,10X)
C     READ POD-RADII DISTRIBUTION
      READ (TAPEIN,380) (PODR(I,J),J=1,NPODOR)
  390 CONTINUE
C
C  CALL POD SUBROUTINE
      CALL NACEL
C
      IFLAP = 0
C
C**FIN**
C*******
  400 IF (J4.EQ.0 .OR. J4.EQ.2) GO TO 490
C
      NAF = 2
      NAFORD = NFINOR
      NCAM = 0
      NACA = 0
      ITIP = 0
      IROOT = 0
      ISURF = 0
      IFLAP = 0
C     READ FIN DATA (NF FINS)
      DO 430 NN=1,NF
C
C     READ ORDINATE DATA AND LENGTH
      READ (TAPEIN,410) ((FINORG(NN,I,J),J=1,4),I=1,2)
  410 FORMAT (8F7.0,24X)
C     READ PERCENT-CHORD LOCATIONS
      READ (TAPEIN,420) (XFIN(NN,I),I=1,NFINOR)
  420 FORMAT (10F7.0,10X)
```

```
C     READ FIN AIRFOIL ORDINATES
      READ (TAPEIN,420) (FINORD(NN,I),I=1,NFINOR)
C
  430 CONTINUE
C
C  INITIALIZE DATA FOR FIN SURFACE GENERATION.
      DO 480 NN = 1,NF
      SURFID = SURF2(NN)
      DO 460  I = 1,NAF
      DO 440 J = 1,3
  440 AFORG(I,J) = FINORG(NN,I,J)
      AFORG(I,4) = AFORG(I,1) - FINORG(NN,I,4)*DX
      AFORG(I,5) = AFORG(I,2)
      AFORG(I,6) = AFORG(I,3)
      AFORG(I,7) = 90.0
C
      DO 450 K = 1,NAFORD
      AFCAM(I,K) = 0.0
  450 AFORD(I,K) = FINORD(NN,K)
  460 CONTINUE
      DO 470 K = 1,NAFORD
  470 XOC(K) = XFIN(NN,K)
C
C
      CALL SURF
  480 CONTINUE
C    END OF FIN DO LOOP
C
C
C**CANARD OR HORIZONTAL TAIL
C*******
  490 IF (J5.EQ.0 .OR. J5.EQ.2) GO TO 600
C
      N = IABS(NCANOR)
      NAF = 2
      NAFORD = N
      NACA = 0
      ITIP = 0
      IROOT = 0
      ISURF = 0
      IFLAP = 0
C    READ IN ALL CANARD DATA (NCAN CANARDS)
      DO 540 NN = 1,NCAN
C     READ ORDINATE DATA AND LENGTH
      READ (TAPEIN,500) ((CANORG(NN,I,J),J=1,4),I=1,2)
```

```
  500 FORMAT (8F7.C,24X)
C     READ PERCENT-CHORD LOCATIONS
      READ (TAPEIN,510) (XCAN(NN,J),J=1,N)
  510 FORMAT (10F7.C,10X)
C    READ FIN AIRFOIL ORDINATES
      READ (TAPEIN,510) (CANORD(NN,J),J=1,N)
C
C     IF AIRFOIL IS NOT SYMMETRICAL READ LOWER ORDINATES
      NCAM = 0
      IF (NCANOR .GT. C) GO TO 520
      NCAM = 1
      READ (TAPEIN,510) (CANOR1(NN,J),J=1,N)
      GO TO 540
C
C    SET LOWER SURFACE EQUAL TO UPPER FOR SYMMETRICAL
  520 DO 530 J=1,N
  530 CANOR1(NN,J) = CANORD(NN,J)
C
  540 CONTINUE
C
C  INITIALIZE DATA FOR CANARD SURFACE GENERATION.
      DO 590 NN = 1,NCAN
      SURFID = SURF3(NN)
      DO 570 I = 1,NAF
      DO 550 J = 1,3
  550 AFORG(I,J) = CANORG(NN,I,J)
      AFORG(I,4) = AFORG(I,1) - CANORG(NN,I,4)*DX
      AFORG(I,5) = AFORG(I,2)
      AFORG(I,6) = AFORG(I,3)
      AFORG(I,7) = C.0
C
      DO 560 K = 1,NAFORD
      AFORD(I,K) = (CANORD(NN,K) + CANOR1(NN,K))*0.5
  560 AFCAM(I,K) = CANORD(NN,K) - AFORD(I,K)
  570 CONTINUE
      DO 580 K = 1,NAFORD
  580 XOC(K) = XCAN(NN,K)
C
C
      CALL SURF
  590 CONTINUE
C    END OF CANARD DO LOOP.
C
C
C**ARBITRARY AIRFOIL SURFACE
```

```
C********
  600 IF (J7 .NE. 1) GO TO 740
C
C READ CONTROL CARD
  610 READ (TAPEIN,620) ISURF,NAF,NAFORD,NCAM,NACA,ITIP,IROOT,ISIMC,
     1                  ISIMT,MORE,SURFID
C
  620 FORMAT (22XI3, 2(3XI2), 7(4XI1),2XA4,4X)
C
C***  CHECK SURFACE TYPE, ISURF =  +1(UPPER ONLY), -1(LOWER ONLY),
C
C                                  ABS .GT. 1,(WITH CONTROL SURFACE).
      IFLAP = 0
      NHNGE =IABS(ISURF)
      IF (NHNGE .LE. 1)  GO TO 630
C
C***  CONTROL SURFACE INPUT INDICATED.  CHECK IF CONSTRAINTS MET.
C     NAFORD MUST BE ODD AND NHNGE EVEN.  IF THESE CONDITIONS ARE
C     NOT SATISFIED, FLAP IS NEGLECTED.
C
      ISURF = ISURF/NHNGE
      IF ((NAFORD/2 + NAFORD/2) .NE. (NAFORD - 1)) GO TO 630
      IF ((NHNGE/2  + NHNGE/2) .NE. NHNGE)   GO TO 630
C
C***  CONSTRAINTS SATISFIED.
      IFLAP = 1
  630 CONTINUE
C READ CHORD LOCATIONS
      READ (TAPEIN,640) (XOC(J),J=1,NAFORD)
  640 FORMAT(10F7.0,10X)
C
C READ AIRFOIL SECTION ORIENTATION DATA
      DO 650 I = 1,NAF
      READ (TAPEIN,640) (AFORG(I,J),J=1,7)
  650 CONTINUE
C
C CHECK FOR CAMBER DATA
      IF (NCAM .NE. 1) GO TO 680
      IN = NAF
      IF (ISIMC .EQ. 1) IN = 1
      DO 660 I = 1,IN
  660 READ (TAPEIN,640) (AFCAM(I,J),J=1,NAFORD)
      IF (IN .EQ. NAF) GO TO 700
      DO 670 I = 2,NAF
      DO 670 J = 1,NAFORD
```

```
  670 AFCAM(I,J) = AFCAM(1,J)                                          AIRC  406
      GO TO 700                                                        AIRC  407
C NO CAMBER DATA INPUT, SET TO ZERO.                                   AIRC  408
  680 DO 690 I = 1,NAF                                                 AIRC  409
      DO 690 J = 1,NAFORD                                              AIRC  410
  690 AFCAM(I,J) = 0.0                                                 AIRC  411
C                                                                      AIRC  412
  700 CONTINUE                                                         AIRC  413
C                                                                      AIRC  414
C READ IN THICKNESS DISTRIBUTION                                       AIRC  415
      IN = NAF                                                         AIRC  416
      IF (ISIMT .EQ. 1) IN = 1                                         AIRC  417
      DO 710 I = 1,IN                                                  AIRC  418
  710 READ (TAPEIN,640) (AFORD(I,J),J=1,NAFORD)                        AIRC  419
      IF (IN .EQ. NAF) GO TO 730                                       AIRC  420
      DO 720 I = 2,NAF                                                 AIRC  421
      DO 720 J = 1,NAFORD                                              AIRC  422
  720 AFORD(I,J) = AFORD(1,J)                                          AIRC  423
C                                                                      AIRC  424
  730 CALL SURF                                                        AIRC  425
C                                                                      AIRC  426
C CHECK FOR STACKED DATA                                               AIRC  427
      IF (MORE .EQ. 1) GO TO 610                                       AIRC  428
C                                                                      AIRC  429
C CHECK IF A DUMMY STATLS 3 ELEMENT WANTED.                            AIRC  430
  740 IF (IFLAP .EQ. 1)  GO TO 10                                      AIRC  431
      IF (ISTAT3 .NE. 1) CALL STATS3                                   AIRC  432
C                                                                      AIRC  433
C***END OF ALL CONFIGURATION COMPONENTS (GO TO STARTING STATEMENT)     AIRC  434
      GO TO 10                                                         AIRC  435
C                                                                      AIRC  436
  750 WRITE (TAPECT,760) I                                             AIRC  437
  760 FORMAT(1H ,//,1H ,37H**** FATAL ERROR *** PROGRAM STOPPED.,      AIRC  438
     1        ///5X,34HCONTROL FLAG CARD PARAMETER NUMBER,I3,4H IS ,   AIRC  439
     2        28HOUTSIDE THE ALLOWABLE RANGE.)                         AIRC  440
C                                                                      AIRC  441
C***AFTER END OF FILE ON INPUT - STOP                                  AIRC  442
  770 STOP                                                             AIRC  443
C 780 RETURN                                                           AIRC  444I
  780 CONTINUE                                                         AIRC  445C
C                                                                      AIRC  446
      END                                                              AIRC  447
```

```
      FUNCTION ARCOS(X)                                                 ARCO  001C
C  THIS ROUTINE IS REQUIRED BECAUSE OF DIFFERENCES BETWEEN CDC AND IBM  ARCO  002
C  FORTRAN.                                                            ARCO  003
      ARCOS = ACOS(X)                                                   ARCO  004C
      RETURN                                                            ARCO  005C
      END                                                               ARCO  006C
```

```
      FUNCTION ARSIN(X)                                                 ARSI  001C
C  THIS ROUTINE IS REQUIRED BECAUSE OF DIFFERENCES BETWEEN CDC AND IBM  ARSI  002
C  FORTRAN.                                                            ARSI  003
      ARSIN = ASIN(X)                                                   ARSI  004C
      RETURN                                                            ARSI  005C
      END                                                               ARSI  006C
```

```
      SUBROUTINE CPINPT(INP)
C
C
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR
      COMMON /GDATA/LTOT,J,SYMFCT,IORN,IGTYPE,L,LTOTAB(20)
      COMMON /ABDATA/NAB,ALPHA(20),BETA(20),ROL(20),CDELTA(20),QI(20),
     1               RI(20),PI(20)
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      COMMON /SURFN2/INORM,ISURF,IX,IY,IZ,U1,UL,V1,VL,CR,CT,CHX,N2,N3,
     1    XO,YO,ZO,AP11,AP12,AP13,AP21,AP22,AP23,AP31,AP32,AP33,
     2  FLOWC(7),XI(500),YI(500),B(503,7),XKF(503),AA(5999),
     A  XB(4), YB(4), ZB(4),
     3  NDSET,IABSET,IR,LORG(20),ISR(20),NS,ISFR,IVIS, IFTYP,IFLOW,
     4    BX(2,20), BY(2,20), NB
      COMMON/PLOTCP/IRCP,IRCT,NEXTCP
      COMMON/ASSOCV/IA15,IA16
      COMMON/INDATA/IRUNID,ICORID,IPLTCP,IBETA,BETAIN,IWRIT,
     1    IPRINT,ISVFMS,ISVSPL
      COMMON/SUBIDX/SIDX04(1001),SIDX16(1001),SIDX21(1001)
      COMMON /MASTER/ MIDX04(21),INDEX9(501),INDX10(2001),INDX15(2001),
     1  MIDX16(94),INDX20(2421),MIDX21(73)
C
      DIMENSION  IRCP(200),IRCT(200),X(100),Y(100),Z(100),IO(100),
     1    IPP(100),IDUM(200),CP(100),IRW(100)
      DIMENSION                  ELEM(25), LCOND(3), DINFL(6)
      DIMENSION TITLE(15),INP(20)
      DIMENSION TITLEM(10),IMTAB(9),LOAB(20),TITLES(10),TITLEA(10),
     1  TITLER(10),E4(12),E2(13),DAT(6),                  IDTYP(5),
     3   E3(17), ISF(5), FDATA(34), XP(4,6)
      DIMENSION XA(500)
C
      INTEGER ERROR,PAGE,CASE,SYMFCT
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      INTEGER SIDX04,SIDX16,SIDX21
C
      EQUIVALENCE  (FDATA(1), LCOND(1)), (FDATA(4), ELEM(1)),
     1  (FDATA(29), DINFL(1))
      EQUIVALENCE  (AA(1),X(1)), (AA(101),Y(1)), (AA(201),Z(1)),
     1    (AA(301),IO(1)), (AA(401),IPP(1)), (AA(501),IDUM(1)),
     2    (AA(701),CP(1)), (AA(801),IRW(1))
      DATA RC/0.1745329E-1/
C  FOR CHECKOUT SET IPRINT = 1
C
```

```
C
      IF (IPRINT .EQ. 0)  GO TO 6
      WRITE(TAPEOT,5)
    5 FORMAT(1H1,45HPREVIOUSLY SAVED SURFACE DATA WILL BE USED TO,
     1              17H CALCULATE FORCES)
C
C  BEGIN ALPHA-BETA CYCLE
    6 REWIND TAPEF
      DO 2000 IAB = 1,NAB
      ICT = 0
      ISYM = 1
      IF (INP(IAB)   .EQ. 0)  GO TO 2000
C
C  INPUT REGION ID INFO AND NORMALIZATION FLAGS
   10 READ (TAPEIN,20) LASTR,NDSET,IABSET,IR,INORM,ISURF,IPF,
     1  (ISR(I),I=1,10), ISF,NN
   20  FORMAT (I1,3I2,3I1,   10I2,5I1,4X,I4)
C
      IF (IPRINT .EQ. 0)  GO TO 34
      WRITE (TAPEOT,32) LASTR,NDSET,IABSET,IR,INORM,ISURF,
     1     IPF, (ISR(I),I=1,10), ISF
   32  FORMAT (1H0,5X,6HLASTR=,I2,4X,6HNDSET=,I2,4X,7HIABSET=,I2,4X,
     1 4HIR =,I2,                 4X,6HINORM=,I2,4X,6HISURF=,I2,4X,
     2     /1H , 4HIPF=, I2, 4X4HISR=, 10I3, 4X4HISF=, 5I2/)
C
C  READ BOUNDARY PCINTS FCR NORMALIZATION
   34 CONTINUE
      DO 38 I=1,4
   38 READ (TAPEIN,50) XB(I),YB(I),ZB(I)
   50  FORMAT (3F10.0)
C
C  READ IN MASTER DIRECTORY
      ITAG10 = 1
      CALL READMS (10,TITLEM,10,ITAG10)
C     READ(10'ITAG10) TITLEM
      ITAG10 = 2
      CALL READMS (10,IMTAB,9,ITAG10)
C     READ(10'ITAG10) IMTAB
C
C  CHECK DATA SET NUMBER
      IF (IMTAB(1) .GE. NDSET)  GO TO 40
      WRITE (TAPEOT,39) NDSET,IMTAB(1)
   39  FORMAT (1H0,9H**NDSET =,I2,31H IS GREATER THAN THE NUMBER OF ,
     1 44HDATA SETS ACTUALLY ON UNIT 10. PROGRAM HALT.  )
      STOP
```

```
C                                                                       CPIN  091
C                                                                       CPIN  092
C  READ IN FLOW DATA SET DIRECTORY                                      CPIN  093
   40 ITAG10 = IMTAB(NDSET + 4)                                         CPIN  094
      CALL READMS (10,E4,12,ITAG10)                                     CPIN  095C
C     READ (10'ITAG10) E4                                               CPIN  096I
      DO 41 I=1,10                                                      CPIN  097
   41 TITLES(I) = E4(I)                                                 CPIN  098
      MACH = E4(11)                                                     CPIN  099
      NABS = E4(12)                                                     CPIN  100
      ITAG10 = ITAG10 + 1                                               CPIN  101
      CALL READMS (10,LOAB,20,ITAG10)                                   CPIN  102C
C     READ(10'ITAG10)  LOAB                                             CPIN  103I
C                                                                       CPIN  104
C  READ IN FLOW REGION DIRECTORY FOR REQUESTED ALPHA-BETA SET           CPIN  105
      ITAG10 = LOAB(IABSET)                                             CPIN  106
      CALL READMS (10,E2,13,ITAG10)                                     CPIN  107C
C     READ (10'ITAG10) E2                                               CPIN  108I
      DO 42 I=1,10                                                      CPIN  109
   42 TITLEA(I) = E2(I)                                                 CPIN  110
      ALPHAS = E2(11)                                                   CPIN  111
      BETAS = E2(12)                                                    CPIN  112
      NREG = E2(13)                                                     CPIN  113
      ITAG10 = ITAG10 + 1                                               CPIN  114
      CALL READMS (10,LORG,20,ITAG10)                                   CPIN  115C
C     READ(10'ITAG10)  LORG                                             CPIN  116I
C                                                                       CPIN  117
C                                                                       CPIN  118
C                                                                       CPIN  119
C                                                                       CPIN  120
C  READ IN REQUESTED FLOW REGIONS                                       CPIN  121
C                                                                       CPIN  122
C                                                                       CPIN  123
C                                                                       CPIN  124
      ITAG10 = LORG(IR)                                                 CPIN  125
      CALL READMS (10,E3,17,ITAG10)                                     CPIN  126C
C     READ (10'ITAG10) E3                                               CPIN  127I
      DO 91 I=1,10                                                      CPIN  128
   91 TITLER(I) = E3(I)                                                 CPIN  129
      DO 92 I=1,5                                                       CPIN  130
      IDTYP(I) = E3(I+10)                                               CPIN  131
   92 CONTINUE                                                          CPIN  132
      IFTYP = IDTYP(1)                                                  CPIN  133
      IFLOW = IDTYP(2)                                                  CPIN  134
      IVIS = 0                                                          CPIN  135
```

```
      IF (IFTYP .EQ. 4)  IVIS = 1                                       CPIN  136
   94 NSREG = E3(16)                                                    CPIN  137
      ITAG10 = ITAG10 + 1                                               CPIN  138
      CALL READMS (10,DAT, 6,ITAG10)                                    CPIN  139C
C     READ (10'ITAG10) DAT                                              CPIN  140I
      XO      = DAT( 1)                                                 CPIN  141
      YO      = DAT( 2)                                                 CPIN  142
      ZO      = DAT( 3)                                                 CPIN  143
      PSIO      = DAT( 4)                                               CPIN  144
      THETO      = DAT( 5)                                              CPIN  145
      PHIO      = DAT( 6)                                               CPIN  146
C                                                                       CPIN  147
      SINT = SIN(THETO*RC)                                              CPIN  148
      COST = COS(THETO*RC)                                              CPIN  149
      COSPS = COS(PSIO*RC)                                              CPIN  150
      SINPS = SIN(PSIO*RC)                                              CPIN  151
      SINP = SIN(PHIO*RC)                                               CPIN  152
      COSP = COS(PHIO*RC)                                               CPIN  153
      AP11 = COST*COSPS                                                 CPIN  154
      AP12 = COST*SINPS                                                 CPIN  155
      AP13 =-SINT                                                       CPIN  156
      AP21 =-COSP*SINPS + SINP*SINT*COSPS                               CPIN  157
      AP22 = COSP*COSPS + SINP*SINT*SINPS                               CPIN  158
      AP23 = SINP*COST                                                  CPIN  159
      AP31 = SINP*SINPS + COSP*SINT*COSPS                               CPIN  160
      AP32 =-SINP*COSPS + COSP*SINT*SINPS                               CPIN  161
      AP33 = COSP*COST                                                  CPIN  162
C                                                                       CPIN  163
C  SET INDICES FOR NORMALIZING DATA                                     CPIN  164
      IF(INORM .LE. 0)  GO TO 100                                       CPIN  165
      GO TO (110,130,140,150),INORM                                     CPIN  166
C                                                                       CPIN  167
C  NORMALIZE W.R.T.  A,R                                                CPIN  168
  100 IX = 4                                                            CPIN  169
      IY = 5                                                            CPIN  170
      IZ = 6                                                            CPIN  171
      GO TO 190                                                         CPIN  172
C                                                                       CPIN  173
C  NORMALIZE W.R.T.  X,Y                                                CPIN  174
  110 IX = 1                                                            CPIN  175
      IY = 2                                                            CPIN  176
      IZ = 3                                                            CPIN  177
      GO TO 190                                                         CPIN  178
C                                                                       CPIN  179
C  NORMALIZE W.R.T.  X,Z                                                CPIN  180
```

```
  130 IX = 1                                                            CPIN  181
      IY = 3                                                            CPIN  182
      IZ = 2                                                            CPIN  183
      GO TO 190                                                         CPIN  184
C                                                                       CPIN  185
C  NORMALIZE W.R.T.  Y,Z                                                CPIN  186
  140 IX = 2                                                            CPIN  187
      IY = 3                                                            CPIN  188
      IZ = 1                                                            CPIN  189
      GO TO 190                                                         CPIN  190
C                                                                       CPIN  191
C  NORMALIZE W.R.T.  A,PHI                                              CPIN  192
  150 IX = 4                                                            CPIN  193
      IY = 6                                                            CPIN  194
      IZ = 5                                                            CPIN  195
      GO TO 190                                                         CPIN  196
C                                                                       CPIN  197
C  CALCULATE NORMALIZING LENGTHS                                        CPIN  198
  190 CONTINUE                                                          CPIN  199
      DO 95 I = 1,4                                                     CPIN  200
      XX =XB(I) - XO                                                    CPIN  201
      YY =YB(I) - YO                                                    CPIN  202
      ZZ =ZB(I) - ZO                                                    CPIN  203
      XP(I,1) = XX*AP11 + YY*AP12 + ZZ*AP13                             CPIN  204
      XP(I,2) = XX*AP21 + YY*AP22 + ZZ*AP23                             CPIN  205
      XP(I,3) = XX*AP31 + YY*AP32 + ZZ*AP33                             CPIN  206
      XP(I,4)  =  XP(I,1)                                               CPIN  207
      XP(I,5)  =  SQRT(XP(I,2)**2 + XP(I,3)**2)                         CPIN  208
      IF (XP(I,2) .EQ. 0.0)  GO TO 194                                  CPIN  209
      XP(I,6)  =  ATAN2(XP(I,2), -XP(I,3))                              CPIN  210
      IF(XP(I,6).LT.0.0) XP(I,6) = 6.283185308 + XP(I,6)                CPIN  211
      GO TO 95                                                          CPIN  212
  194 XP(I,6)  =  0.0                                                   CPIN  213
      IF (XP(I,3) .GT. 0.0)  XP(I,6) = 3.141592654                      CPIN  214
   95 CONTINUE                                                          CPIN  215
C                                                                       CPIN  216
C                                                                       CPIN  217
      U1 = XP(1,IX)                                                     CPIN  218
      UL = XP(2,IX) - U1                                                CPIN  219
      CR = UL                                                           CPIN  220
      IF (ISURF  .EQ. 1)  GO TO 96                                      CPIN  221
      V1 = XP(3,IY)                                                     CPIN  222
      VL = XP(4,IY) - V1                                                CPIN  223
      GO TO 97                                                          CPIN  224
C                                                                       CPIN  225
```

```
   96 UL = XP(3,IX)                                                     CPIN  226
      CT = XP(4,IX) - XP(3,IX)                                          CPIN  227
      V1 = XP(1,IY)                                                     CPIN  228
      VL = XP(3,IY) - V1                                                CPIN  229
   97 CONTINUE                                                          CPIN  230
C                                                                       CPIN  231
C                                                                       CPIN  232
C                                                                       CPIN  233
C  ESTABLISH SUB-REGION COUNTERS                                        CPIN  234
      IF (ISR(1) .GT. 0)  GO TO 210                                     CPIN  235
      NS = NSREG                                                        CPIN  236
      DO 200 I = 1,NS                                                   CPIN  237
  200 ISR(I) = I                                                        CPIN  238
      GO TO 230                                                         CPIN  239
  210 NS = 0                                                            CPIN  240
      DO 220 I = 1,10                                                   CPIN  241
      IF (ISR(I) .LE. 0)  GO TO 230                                     CPIN  242
  220 NS = NS + 1                                                       CPIN  243
C                                                                       CPIN  244
  230 IF (NS .LE. NSREG)  GO TO 250                                     CPIN  245
      WRITE(TAPEOT,240)  NS,NSREG                                       CPIN  246
  240 FORMAT(1H0,28HSURFACE INTERPOLATION.  THE ,I2,13H SUB-REGIONS ,   CPIN  247
     1 36HREQUESTED IS GREATER THAN AVAILABLE./1H0, 12HCOUNTER SET ,    CPIN  248
     2 11HTO NSREG = , I2, 20H AND CASE CONTINUES.)                     CPIN  249
C                                                                       CPIN  250
      NS = NSREG                                                        CPIN  251
  250 CONTINUE                                                          CPIN  252
C                                                                       CPIN  253
C                                                                       CPIN  254
      REWIND TAPEE                                                      CPIN  255
      INTAPE = TAPEC                                                    CPIN  256
      ITAPES = TAPED                                                    CPIN  257
C                                                                       CPIN  258
      I = 0                                                             CPIN  259
      ISFR = 0                                                          CPIN  260
      ICYCLE = 0                                                        CPIN  261
      NTOT = 0                                                          CPIN  262
      IF (IPF .NE. 0)  GO TO 1520                                       CPIN  263
 1100 CONTINUE                                                          CPIN  264
      ERROR = 0                                                         CPIN  265
      CALL SFNTR2                                                       CPIN  266
      IF (ERROR .EQ. 1)  GO TO 1520                                     CPIN  267
C                                                                       CPIN  268
C  SPECIAL OUTPUT OF FITTED CP SURFACE                                  CPIN  269
C                                                                       CPIN  270
```

```
      IF (NN .EQ. 0)  GO TO 1110                                        CPIN  271
      IF (NN - 1) 1101, 1120, 1102                                      CPIN  272
C                                                                       CPIN  273
C  READ IN PANEL NUMBER AND NUMBER OF ROWS                              CPIN  274
 1120 CONTINUE                                                          CPIN  275
 1125 READ(TAPEIN,1130)  NPAN, NR,MOREP                                 CPIN  276
 1130 FORMAT(3I5)                                                       CPIN  277
C                                                                       CPIN  278
C  READ IN ROW SEQUENCES NUMBERS                                        CPIN  279
      READ(TAPEIN,1140)  (IRW(I),I=1,NR)                                CPIN  280
 1140 FORMAT(14I5)                                                      CPIN  281
C                                                                       CPIN  282
C  BEGIN CYCLE ON ROWS                                                  CPIN  283
      DO 1190 I = 1,NR                                                  CPIN  284
      ISQ = IRW(I)                                                      CPIN  285
C                                                                       CPIN  286
C  READ IN COORDINATE DATA                                              CPIN  287
      IG15 = 7                                                          CPIN  288
C     READ(15'IG15) IDUM                                                CPIN  289I
      CALL READMS (15,IDUM,100,IG15)                                    CPIN  290C
      CALL READMS (15,IDUM(101),100,IG15+1)                             CPIN  291C
      NPT = IDUM(ISQ)                                                   CPIN  292
C     READ(15'IG15+4)  IDUM                                             CPIN  293I
      CALL READMS (15,IDUM,100,IG15+4)                                  CPIN  294C
      CALL READMS (15,IDUM(101),100,IG15+5)                             CPIN  295C
      IG15 = IDUM(ISQ)                                                  CPIN  296
      K=NPT*5                                                           CPIN  297
C     READ (15'IG15) (XA(J),J=1,K)                                      CPIN  298I
      CALL READMS (15,XA,K,IG15)                                        CPIN  299C
      REWIND 88                                                         CPIN  300
      WRITE (88) (XA(J),J=1,K)                                          CPIN  301
      REWIND 88                                                         CPIN  302
      READ (88) (X(J),Y(J),Z(J),IO(J),IPP(J),J=1,NPT)                   CPIN  303
C                                                                       CPIN  304
C  CHECK CP POINTER                                                     CPIN  305
      NEWCP = 0                                                         CPIN  306
      IF (IRCP(ISQ) .NE. 0)  GO TO 1160                                 CPIN  307
      NEWCP = 1                                                         CPIN  308
      ICPP = NEXTCP                                                     CPIN  309
      DO 1150 J = 1,NPT                                                 CPIN  310
 1150 CP(J) = 0.0                                                       CPIN  311
      NCP = 0                                                           CPIN  312
      GO TO 1170                                                        CPIN  313
C                                                                       CPIN  314
C  READ IN EXISTING CP DATA                                             CPIN  315
```

```
 1160 ICPP = IRCP(ISQ)                                                  CPIN  316
      NCP = IRCT(ISQ)                                                   CPIN  317
C     READ(16'ICPP)  (CP(J), J=1,NPT)                                   CPIN  318I
      NSUB=(ICPP-1)/1000+1                                              CPIN  319C
      NSRC16=ICPP-(NSUB-1)*1000                                         CPIN  320C
      CALL STINDX (16,MIDX16,94)                                        CPIN  321C
      CALL READMS (16,SIDX16,1001,NSUB)                                 CPIN  322C
      CALL STINDX (16,SIDX16,1001)                                      CPIN  323C
      CALL READMS (16,CP,NPT,NSRC16)                                    CPIN  324C
C                                                                       CPIN  325
C  CALCULATE CP'S                                                       CPIN  326
 1170 IC = 0                                                            CPIN  327
      IT = 1                                                            CPIN  328
      ISVE = 0                                                          CPIN  329
C                                                                       CPIN  330
      DO 1180 J = 1,NPT                                                 CPIN  331
C                                                                       CPIN  332
C  CHECK PANEL NUMBER                                                   CPIN  333
      IF (NPAN .NE. IPP(J))  GO TO 1180                                 CPIN  334
      XX = X(J)                                                         CPIN  335
      YY = Y(J)                                                         CPIN  336
      ZZ = Z(J)                                                         CPIN  337
C                                                                       CPIN  338
      CALL VALU2(IT,XX,YY,ZZ,XN,YN,ZN,IC,IT,INR,                        CPIN  339
     1  DINFL(1),DINFL(2),DINFL(3),DINFL(4),DINFL(5),DINFL(6))          CPIN  340
C                                                                       CPIN  341
C  CHECK IF POINT WITHIN BOUNDARIES                                     CPIN  342
      IF (INR .NE. 0)  GO TO 1175                                       CPIN  343
      DINFL(5) = 999.999                                                CPIN  344
      IO(J) = -1                                                        CPIN  345
      ISVE = 1                                                          CPIN  346
C                                                                       CPIN  347
 1175 NCP = NCP + 1                                                     CPIN  348
      CP(J) = DINFL(5)                                                  CPIN  349
      IF (IPRINT .EQ. 0)  GO TO 1180                                    CPIN  350
      WRITE(TAPEOT,5010)  XX,YY,ZZ,CP(J),IO(J),ISQ,J,IPP(J)             CPIN  351
 1180 CONTINUE                                                          CPIN  352
C                                                                       CPIN  353
C  SAVE CP DATA ON UNIT 16                                              CPIN  354
      IRCP(ISQ) = ICPP                                                  CPIN  355
      IRCT(ISQ) = NCP                                                   CPIN  356
      IA16 = ICPP                                                       CPIN  357
C     WRITE(16'IA16)  (CP(J), J=1,NPT)                                  CPIN  358I
      NSUB=(IA16-1)/1000+1                                              CPIN  359C
      NSRC16=IA16-(NSUB-1)*1000                                         CPIN  360C
```

```
      CALL STINDX (16,MIDX16,94)                                        CPIN  361C
      CALL READMS (16,SIDX16,1001,NSUB)                                 CPIN  362C
      CALL STINDX (16,SIDX16,1001)                                      CPIN  363C
      CALL WRITMS (16,CP,NPT,NSRC16,0)                                  CPIN  364C
      CALL STINDX (16,MIDX16,94)                                        CPIN  365C
      CALL WRITMS (16,SIDX16,1001,NSUB,1)                               CPIN  366C
C     IF (NEWCP.EQ.1) NEXTCP = IA16                                     CPIN  367I
      IF (NEWCP.EQ.1) NEXTCP=IA16+1                                     CPIN  368C
C                                                                       CPIN  369
C  CHECK IF ANY IO FLAGS HAVE BEEN CHANGED                              CPIN  370
      IF (ISVE .EQ. 0)  GO TO 1190                                      CPIN  371
      REWIND 88                                                         CPIN  372
      WRITE (88) (X(J),Y(J),Z(J),IO(J),IPP(J),J=1,NPT)                  CPIN  373
      REWIND 88                                                         CPIN  374
      READ (88) (XA(J),J=1,K)                                           CPIN  375
C     WRITE (15'IG15) (XA(J),J=1,K)                                     CPIN  376I
      CALL WRITMS (15,XA,K,IG15,1)                                      CPIN  377C
C                                                                       CPIN  378
C  CYCLE TO NEXT ROW                                                    CPIN  379
 1190 CONTINUE                                                          CPIN  380
      IF (MOREP .EQ. 1)  GO TO 1125                                     CPIN  381
      GO TO 1110                                                        CPIN  382
C                                                                       CPIN  383
C                                                                       CPIN  384
 1101 NN = IABS(NN)                                                     CPIN  385
      WRITE(TAPEOT,4090)  NN                                            CPIN  386
 4090 FORMAT(1H1,40HSPECIAL  OUTPUT  OF  FITTED  CP  SURFACE/           CPIN  387
     1        1H0,10H       AT   , I3,  15H  INPUT  POINTS/             CPIN  388
     2        1H0, 6X,1HX, 9X,1HY, 9X,1HZ, 9X,2HCP)                     CPIN  389
      IC = 0                                                            CPIN  390
      IT = 1                                                            CPIN  391
      DO 5030 II = 1,NN                                                 CPIN  392
      READ(TAPEIN,5000)  XX,YY,ZZ, IFLG,IROW,JJ,IPAN                    CPIN  393
 5000 FORMAT( 3F10.3, 24X,I3,4X,I4,3X,I4,4X,I4)                         CPIN  394
      CALL VALU2(IT,XX,YY,ZZ,XN,YN,ZN,IC,IT,INR,                        CPIN  395
     1     DINFL(1),DINFL(2),DINFL(3),DINFL(4),DINFL(5),DINFL(6))       CPIN  396
C                                                                       CPIN  397
C  CHECK IF POINT WAS WITHIN BOUNDARIES                                 CPIN  398
      IF (INR .EQ. 0)  IFLG = -1                                        CPIN  399
C                                                                       CPIN  400
      WRITE(TAPEOT,5010)  XX,YY,ZZ,DINFL(5),IFLG,IROW,JJ,IPAN           CPIN  401
 5010 FORMAT(1H0, 4F10.3, 10X,4H I/O,I3,4H ROW,I4,3H PT,I4,4H PAN,I4)   CPIN  402
      WRITE(TAPEG, 5020)  XX,YY,ZZ,DINFL(5),IFLG,IROW,JJ,IPAN           CPIN  403
 5020 FORMAT(     4F10.3, 10X,4H I/O,I3,4H ROW,I4,3H PT,I4,4H PAN,I4)   CPIN  404
 5030 CONTINUE                                                          CPIN  405
```

```
      GO TO 1110                                                        CPIN  406
 1102 CONTINUE                                                          CPIN  407
      NN2 = 2*NN                                                        CPIN  408
      NN3 = 3*NN                                                        CPIN  409
      IC = 1                                                            CPIN  410
      IT = 1                                                            CPIN  411
      DN = 1.0/(NN-1)                                                   CPIN  412
      XN = -DN                                                          CPIN  413
C                                                                       CPIN  414
      WRITE(TAPEOT,4000) NN,DN                                          CPIN  415
 4000 FORMAT(1H1,19X, 40HSPECIAL  OUTPUT  OF  FITTED  CP  SURFACE/      CPIN  416
     1        1H0,24X,  5HNN = ,I3, 12X,5HDY = , F6.3/)                 CPIN  417
C                                                                       CPIN  418
      DO 3100 II = 1,NN                                                 CPIN  419
      XN = XN + DN                                                      CPIN  420
      YN = -DN                                                          CPIN  421
      DO 3000 J = 1,NN                                                  CPIN  422
      YN = YN + DN                                                      CPIN  423
      CALL VALU2(IC,XX,YY,ZZ,XN,YN,ZN,IC,IT,INR,                        CPIN  424
     1    DINFL(1),DINFL(2),DINFL(3),DINFL(4),DINFL(5),DINFL(6))        CPIN  425
      XX = XX - XO                                                      CPIN  426
      YY = YY - YO                                                      CPIN  427
      ZZ = ZZ - ZO                                                      CPIN  428
      ELEM(1) = XX*AP11 + YY*AP12 + ZZ*AP13                             CPIN  429
      ELEM(2) = XX*AP21 + YY*AP22 + ZZ*AP23                             CPIN  430
      ELEM(3) = XX*AP31 + YY*AP32 + ZZ*AP33                             CPIN  431
      ELEM(4) = ELEM(1)                                                 CPIN  432
      ELEM(5) = SQRT(ELEM(2)**2 + ELEM(3)**2)                           CPIN  433
      IF (ELEM(2) .EQ. 0.0)  GO TO 2094                                 CPIN  434
      ELEM(6) = ATAN2(ELEM(2), -ELEM(3))                                CPIN  435
      GO TO 2095                                                        CPIN  436
 2094 ELEM(6) = 0.0                                                     CPIN  437
      IF (ELEM(3) .GT. 0.0)  ELEM(6) = 3.141592654                      CPIN  438
 2095 CONTINUE                                                          CPIN  439
      AA(J)      = ELEM(IX)                                             CPIN  440
      AA(NN +J)  = ELEM(IY)                                             CPIN  441
      AA(NN2+J)  = ELEM(IZ)                                             CPIN  442
      AA(NN3+J)  = DINFL(5)                                             CPIN  443
 3000 CONTINUE                                                          CPIN  444
C                                                                       CPIN  445
      WRITE(TAPEOT,4010)  XN                                            CPIN  446
 4010 FORMAT(1H0, 1X,5HXN = , F6.3)                                     CPIN  447
C                                                                       CPIN  448
      J2 = 0                                                            CPIN  449
 3010 J1 = J2 + 1                                                       CPIN  450
```

```
      J2 = J1 + 5                                                       CPIN  451
      IF (J2 .GT. NN)  J2 = NN                                          CPIN  452
      WRITE(TAPEOT,4020)  (J, J=J1,J2)                                  CPIN  453
 4020 FORMAT(1H0, 13XI3, 5(17XI3))                                      CPIN  454
      WRITE(TAPEOT,4030)  (AA(J), J=J1,J2)                              CPIN  455
 4030 FORMAT(3H X1, 6E20.6)                                             CPIN  456
      WRITE(TAPEOT,4040)  (AA(NN+J), J=J1,J2)                           CPIN  457
 4040 FORMAT(3H X2, 6E20.6)                                             CPIN  458
      WRITE(TAPEOT,4050)  (AA(NN2+J),J=J1,J2)                           CPIN  459
 4050 FORMAT(3H X3, 6E20.6)                                             CPIN  460
      WRITE(TAPEOT,4060)  (AA(NN3+J),J=J1,J2)                           CPIN  461
 4060 FORMAT(3H CP, 6E20.6)                                             CPIN  462
      IF (J2 .NE. NN)  GO TO 3010                                       CPIN  463
 3100 CONTINUE                                                          CPIN  464
C                                                                       CPIN  465
C                                                                       CPIN  466
C                                                                       CPIN  467
C  START CYCLE ON THE ELEMENTS                                          CPIN  468
 1110 REWIND INTAPE                                                     CPIN  469
      REWIND ITAPES                                                     CPIN  470
      ISAVET = ITAPES                                                   CPIN  471
      ICYCLE = ICYCLE + 1                                               CPIN  472
      IF (IPRINT .NE. 2)  GO TO 1111                                    CPIN  473
      WRITE(TAPEOT,9030)  INTAPE,ITAPES,LTOT                            CPIN  474
 9030 FORMAT(1H0,23HAT LOOP 1500, INTAPE = ,I3, 5X9HITAPES = ,I3,       CPIN  475
     1     5X7HLTOT = ,I5)                                              CPIN  476
 1111 CONTINUE                                                          CPIN  477
      DO 1500 II = 1,LTOT                                               CPIN  478
      IF (ICYCLE .GT. 1)  GO TO 1050                                    CPIN  479
C     READ(TAPEE,END=1510) IG4,ELEM                                     CPIN  480I
      READ(TAPEE) IG4,ELEM                                              CPIN  481C
      IF (EOF(TAPEE)) 1510,1012                                         CPIN  482C
 1012 LCOND(2) = ELEM(1)                                                CPIN  483
      GO TO 1090                                                        CPIN  484
 1050 CONTINUE                                                          CPIN  485
C     READ(INTAPE,END=1510)  IG4,FDATA                                  CPIN  486I
      READ(INTAPE)  IG4,FDATA                                           CPIN  487C
      IF (EOF(INTAPE)) 1510,1080                                        CPIN  488C
C                                                                       CPIN  489
C  CHECK IF ELEMENT ALREADY PROCESSED ON PREVIOUS CYCLE                 CPIN  490
 1080 IF (LCOND(1) .NE. 0)  GO TO 1200                                  CPIN  491
 1090 CONTINUE                                                          CPIN  492
      IF (ISYM .EQ. 1)  GO TO 1095                                      CPIN  493
C  REFLECT ELEMENT TO -Y SIDE AND CHANGE ORDER OF POINTS 2 AND 4        CPIN  494
      IG4 = -IG4                                                        CPIN  495
```

```
      ELEM(5)  = -ELEM(5)                                               CPIN  496
      ELEM(8)  = -ELEM(8)                                               CPIN  497
      ELEM(15) = -ELEM(15)                                              CPIN  498
      ELEM(16) = -ELEM(16)                                              CPIN  499
      ELEM(17) = -ELEM(17)                                              CPIN  500
      ELEM(18) = -ELEM(18)                                              CPIN  501
      ELT = ELEM(12)                                                    CPIN  502
      ELEM(12) = ELEM(14)                                               CPIN  503
      ELEM(14) = ELT                                                    CPIN  504
      ELT = ELEM(16)                                                    CPIN  505
      ELEM(16) = ELEM(18)                                               CPIN  506
      ELEM(18) = ELT                                                    CPIN  507
      ELT = ELEM(20)                                                    CPIN  508
      ELEM(20) = ELEM(22)                                               CPIN  509
      ELEM(22) = ELT                                                    CPIN  510
C                                                                       CPIN  511
C  INTERPOLATE FOR SURFACE PROPERTIES AT CENTROIDS.                     CPIN  512
 1095 CONTINUE                                                          CPIN  513
      IC = 0                                                            CPIN  514
      IT = 1                                                            CPIN  515
      DINFL(5) = 0.0                                                    CPIN  516
      CALL VALU2(1, ELEM(7), ELEM(8), ELEM(9), XN, YN, ZN,IC,IT,INR,    CPIN  517
     1    DINFL(1), DINFL(2), DINFL(3), DINFL(4), DINFL(5), DINFL(6))   CPIN  518
C                                                                       CPIN  519
      LCOND(1) = 0                                                      CPIN  520
      IF (INR .EQ. 0)  GO TO 1200                                       CPIN  521
      LCOND(1) = 1                                                      CPIN  522
      NTOT = NTOT + 1                                                   CPIN  523
C                                                                       CPIN  524
C  SAVE RESULTS CN TEMPORARY UNIT                                       CPIN  525
 1200 CONTINUE                                                          CPIN  526
      WRITE(ITAPES)  IG4,FDATA                                          CPIN  527
C                                                                       CPIN  528
C  GO TO NEXT ELEMENT                                                   CPIN  529
 1500 CONTINUE                                                          CPIN  530
 1510 CONTINUE                                                          CPIN  531
C                                                                       CPIN  532
C                                                                       CPIN  533
 1520 I = I + 1                                                         CPIN  534
      IF (I .GT. 5)  GO TO 1600                                         CPIN  535
      ISFR = ISF(I)                                                     CPIN  536
      IF (IPRINT .NE. 2)  GO TO 1525                                    CPIN  537
      WRITE(TAPEOT,9010)  I,ISFR,ICYCLE,NTOT,ERROR                      CPIN  538
 9010 FORMAT(1H0, 4HI = ,I3, 5X7HISFR = ,I3, 5X9HICYCLE = ,I3,          CPIN  539
     1  7HNTOT = ,I5, 5X8HERROR = ,I3)                                  CPIN  540
```

```
 1525 CONTINUE                                                          CPIN  541
      IF (ISFR .EQ. 0)  GO TO 1600                                      CPIN  542
      IF (ICYCLE .EQ. 0)  GO TO 1100                                    CPIN  543
      IF (NTOT .EQ. LTOT)  GO TO 1600                                   CPIN  544
C                                                                       CPIN  545
C   SWITCH TAPES (ONLY IF ERROR = 0)                                    CPIN  546
      IF (ERROR .NE. 0)  GO TO 1100                                     CPIN  547
      I1 = ITAPES                                                       CPIN  548
      ITAPES = INTAPE                                                   CPIN  549
      INTAPE = I1                                                       CPIN  550
      GO TO 1100                                                        CPIN  551
C                                                                       CPIN  552
C   SAVE RESULTS ON TAPEF                                               CPIN  553
 1600 CONTINUE                                                          CPIN  554
      IF (IPRINT .EQ. 0)  GO TO 1548                                    CPIN  555
      WRITE(TAPEOT,1549)  LTOT, NTOT                                    CPIN  556
 1549 FORMAT(1H0,38HTHE NUMBER OF ELEMENTS IN COMPONENT = , I5/         CPIN  557
     1       1H0,38HTHE NUMBER OF ELEMENTS INTERPOLATED = , I5)         CPIN  558
 1548 IF (ICYCLE .GT. 0)  GO TO 1560                                    CPIN  559
      WRITE(TAPEOT,1550)                                                CPIN  560
 1550 FORMAT(1H1,47HSURFACE  INTERPOLATION (CPINT) CANNOT BE MADE./     CPIN  561
     154H0CHECK THAT PROPER FLOW FIELD DATA HAS BEEN SPECIFIED.,        CPIN  562
     2 16H RUN IS STOPPED.)                                             CPIN  563
      STOP                                                              CPIN  564
C                                                                       CPIN  565
 1560 REWIND ISAVET                                                     CPIN  566
      IF (IPRINT .NE. 2)  GO TO 1601                                    CPIN  567
      WRITE(TAPEOT,9020)  LTOT,INTAPE,ISAVET                            CPIN  568
 9020 FORMAT(1H0, 7HLTOT = ,I5, 5X9HINTAPE = ,I3, 5X9HISAVET = ,I3)     CPIN  569
 1601 CONTINUE                                                          CPIN  570
      DO 1610 L = 1,LTOT                                                CPIN  571
C     READ(ISAVET,END=1620)  IG4,FDATA                                  CPIN  572I
      READ(ISAVET)  IG4,FDATA                                           CPIN  573C
      IF (EOF(ISAVET))  1620,1605                                       CPIN  574C
 1605 WRITE(TAPEF)  IG4,FDATA,NDSET,IABSET,NS,IPF,ISFR                  CPIN  575
      ICT = ICT + 1                                                     CPIN  576
      IF (IPRINT .NE. 2)  GO TO 1610                                    CPIN  577
      WRITE(TAPEOT,9000)  IG4,LCOND,ELEM,DINFL                          CPIN  578
 9000 FORMAT(1H0, 6HIG4 = ,I5, 5X8HLCOND = , 3I3/                       CPIN  579
     1  8H ELEM = , 10E12.4/1H ,7X10E12.4/1H , 7X,5E12.4/               CPIN  580
     2  9H DINFL = , 6E12.4)                                            CPIN  581
 1610 CONTINUE                                                          CPIN  582
 1620 CONTINUE                                                          CPIN  583
C                                                                       CPIN  584
C   CHECK FOR SYMMETRY REQUIREMENTS                                     CPIN  585
```

```
      IF (SYMFCT.EQ.1 .OR. BETAS.EQ.0.0 .OR. ISYM.EQ.2)  GO TO 1900
      REWIND TAPEE
      ISYM = 2
      GO TO 10
C
C
 1900 LTOTAB(IAB) = ICT
C  END OF ALPHA-BETA LOOP
 2000 CONTINUE
C
      REWIND TAPEE
      REWIND TAPEF
C
      RETURN
      END
```

```
      OVERLAY (MARK4,1,3)                                               CUBI  001C
      PROGRAM CUBIC                                                     CUBI  002C
C     SUBROUTINE CUBIC                                                  CUBI  003I
C THIS SUBPROGRAM CALCULATES THE QUADRILATERAL DATA FOR A SURFACE GIVEN CUBI  004
C BY THE COONS MIT SURFACE FIT TECHNIQUE.                               CUBI  005
C                                                                       CUBI  006
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR                                CUBI  007
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   CUBI  008
     1            TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                         CUBI  009
      DIMENSION XA(20),XB(20),YA(20),YB(20),ZA(20),ZB(20),XB1(4,20),YB1( CUBI  010
     14,20),ZB1(4,20),NPTS(4),D(4,9),TITLE(15),SECT(1)                  CUBI  011
      REAL L21,L31,L32,L1,M1,N1,L2,M2,N2,LN,MN,NN                       CUBI  012
      INTEGER STAT,STATT,TYPE,SEQ,CASE,PAGE,ERROR                       CUBI  013
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,       CUBI  014
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                            CUBI  015
C                                                                       CUBI  016
      WRITE (TAPEOT,10)                                                 CUBI  017
   10 FORMAT (1H1,///////,1H0,33HPARAMETRIC CUBIC GEOMETRY DATA IS      CUBI  018
     1     24H BEING GENERATED *******   )                              CUBI  019
C                                                                       CUBI  020
      TYPE=3                                                            CUBI  021
      SEQ=1                                                             CUBI  022
      LINE=100                                                          CUBI  023
      NREC=0                                                            CUBI  024
C                                                                       CUBI  025
C *******READ IN BOUNDARY CURVE DATA                                    CUBI  026
C     SET UP STARTING CONSTANTS                                         CUBI  027
C                                                                       CUBI  028
   20 CONTINUE                                                          CUBI  029
      N=-1                                                              CUBI  030
      L=0                                                               CUBI  031
   30 II=1                                                              CUBI  032
      ITRUE=0                                                           CUBI  033
      IFALSE=1                                                          CUBI  034
      READ (TAPEIN,40) (TITLE(K),K=1,12),NOW,NOW,LAST,ISOVR,IPRINT,CASE, CUBI  035
     1     SECT,ITYPE                                                   CUBI  036
   40 FORMAT (12A4,1X,I3,1X,I3,3X,3I1,2X12,1A4,I2)                      CUBI  037
      IF (ITYPE .NE. 6) GO TO 580                                       CUBI  038
C                                                                       CUBI  039
C     READ IN BOUNDARY CURVE DATA                                       CUBI  040
C                                                                       CUBI  041
   50 CONTINUE                                                          CUBI  042
      READ (TAPEIN,60) X,Y,Z,ISTAT,XX,YY,ZZ,ISTATT,ITYPE                CUBI  043
   60 FORMAT(3F10.4,I1,3F10.4,I1,8X12)                                  CUBI  044
      IF (ITYPE .NE. 7) GO TO 580                                       CUBI  045
```

```
      IRFLAG=IFALSE                                                     CUBI  046
      GO TO 150                                                         CUBI  047
   70 IF(IRFLAG)80,90,80                                                CUBI  048
   80 IRFLAG=ITRUE                                                      CUBI  049
      X=XX                                                              CUBI  050
      Y=YY                                                              CUBI  051
      Z=ZZ                                                              CUBI  052
      ISTAT=ISTATT                                                      CUBI  053
      GO TO 100                                                         CUBI  054
   90 IRFLAG=IFALSE                                                     CUBI  055
      READ (TAPEIN,60) X,Y,Z,ISTAT,XX,YY,ZZ,ISTATT,ITYPE                CUBI  056
      IF (ITYPE .NE. 7) GO TO 580                                       CUBI  057
  100 IF(ISTAT)110,250,110                                              CUBI  058
  110 IF(ISTAT-3)120,250,120                                            CUBI  059
  120 IF(ISTAT-2)130,270,130                                            CUBI  060
  130 IF(IAFLAG-1)140,270,140                                           CUBI  061
  140 MC=M                                                              CUBI  062
  150 M=1                                                               CUBI  063
      IF(ISTAT-2)160,230,160                                            CUBI  064
  160 IF(IBFLAG-1)170,200,170                                           CUBI  065
  170 DO 180 J=1,MC                                                     CUBI  066
      XA(J)=XB(J)                                                       CUBI  067
      YA(J)=YB(J)                                                       CUBI  068
  180 ZA(J)=ZB(J)                                                       CUBI  069
  190 XB(1)=X                                                           CUBI  070
      YB(1)=Y                                                           CUBI  071
      ZB(1)=Z                                                           CUBI  072
      GO TO 70                                                          CUBI  073
  200 IF(IAFLAG)210,220,210                                             CUBI  074
  210 IBFLAG=0                                                          CUBI  075
      GO TO 170                                                         CUBI  076
  220 IAFLAG=1                                                          CUBI  077
      GO TO 190                                                         CUBI  078
  230 IAFLAG=0                                                          CUBI  079
      IBFLAG=1                                                          CUBI  080
      N=N+1                                                             CUBI  081
  240 XA(M)=X                                                           CUBI  082
      YA(M)=Y                                                           CUBI  083
      ZA(M)=Z                                                           CUBI  084
      GO TO 70                                                          CUBI  085
  250 M=M+1                                                             CUBI  086
      IF(IAFLAG)260,240,260                                             CUBI  087
  260 XB(M)=X                                                           CUBI  088
      YB(M)=Y                                                           CUBI  089
      ZB(M)=Z                                                           CUBI  090
```

```
      IF(ISTAT-3)70,270,70                                              CUBI  091
  270 ML=MC                                                             CUBI  092
      MC=M                                                              CUBI  093
  280 N=N+1                                                             CUBI  094
C                                                                       CUBI  095
C  SET UP BOUNDARY CURVE COORDINATE ARRAYS                              CUBI  096
  290 CONTINUE                                                          CUBI  097
      IF(II-1)300,300,320                                               CUBI  098
  300 DO 310 I=1,ML                                                     CUBI  099
      XB1(II,I)=XA(I)                                                   CUBI  100
      YB1(II,I)=YA(I)                                                   CUBI  101
  310 ZB1(II,I)=ZA(I)                                                   CUBI  102
      NPTS(II)=ML                                                       CUBI  103
  320 II=II+1                                                           CUBI  104
      DO 330 I=1,MC                                                     CUBI  105
      XB1(II,I)=XB(I)                                                   CUBI  106
      YB1(II,I)=YB(I)                                                   CUBI  107
  330 ZB1(II,I)=ZB(I)                                                   CUBI  108
      NPTS(II)=MC                                                       CUBI  109
      IF(II-4)150,340,340                                               CUBI  110
  340 CONTINUE                                                          CUBI  111
      IF(ISTAT-3)30,350,350                                             CUBI  112
  350 CONTINUE                                                          CUBI  113
C                                                                       CUBI  114
C                                                                       CUBI  115
C *******CALCULATE BOUNDARY CURVE CONSTANTS                             CUBI  116
C  CALCULATE ARC LENGTH S ON BOUNDARY                                   CUBI  117
      NB=1                                                              CUBI  118
  360 S=0.0                                                             CUBI  119
      K=NPTS(NB)-2                                                      CUBI  120
      DO 370 I=2,K                                                      CUBI  121
  370 S=S+SQRT((XB1(NB,I+1)-XB1(NB,I))**2+(YB1(NB,I+1)-YB1(NB,I))**2+(ZB CUBI  122
     11(NB,I+1)-ZB1(NB,I))**2)                                          CUBI  123
C  CALCULATE TANGENT VECTORS AT THE START OF THE BOUNDARY               CUBI  124
      IFLAG1=0                                                          CUBI  125
      J1=1                                                              CUBI  126
      J2=2                                                              CUBI  127
      J3=3                                                              CUBI  128
  380 X2X1=XB1(NB,J2)-XB1(NB,J1)                                        CUBI  129
      X3X1=XB1(NB,J3)-XB1(NB,J1)                                        CUBI  130
      X3X2=XB1(NB,J3)-XB1(NB,J2)                                        CUBI  131
      Y2Y1=YB1(NB,J2)-YB1(NB,J1)                                        CUBI  132
      Y3Y1=YB1(NB,J3)-YB1(NB,J1)                                        CUBI  133
      Y3Y2=YB1(NB,J3)-YB1(NB,J2)                                        CUBI  134
      Z2Z1=ZB1(NB,J2)-ZB1(NB,J1)                                        CUBI  135
```

```
      Z3Z1=ZB1(NB,J3)-ZB1(NB,J1)                                        CUBI  136
      Z3Z2=ZB1(NB,J3)-ZB1(NB,J2)                                        CUBI  137
      L21=SQRT(X2X1*X2X1+Y2Y1*Y2Y1+Z2Z1*Z2Z1)                           CUBI  138
      L31=SQRT(X3X1*X3X1+Y3Y1*Y3Y1+Z3Z1*Z3Z1)                           CUBI  139
      L32=SQRT(X3X2*X3X2+Y3Y2*Y3Y2+Z3Z2*Z3Z2)                           CUBI  140
      L1=X3X1/L31                                                       CUBI  141
      M1=Y3Y1/L31                                                       CUBI  142
      N1=Z3Z1/L31                                                       CUBI  143
      L2=X2X1/L21                                                       CUBI  144
      M2=Y2Y1/L21                                                       CUBI  145
      N2=Z2Z1/L21                                                       CUBI  146
      LN=-(N1*(L1*N2-L2*N1)+M1*(L1*M2-L2*M1))                           CUBI  147
      MN=-(N1*(M1*N2-M2*N1)+L1*(L1*M2-L2*M1))                           CUBI  148
      NN=M1*(M1*N2-M2*N1)+L1*(L1*N2-L2*N1)                              CUBI  149
      COSEP1=(X2X1*X3X1+Y2Y1*Y3Y1+Z2Z1*Z3Z1)/(L21*L31)                  CUBI  150
      IF(COSEP1-0.999999)400,400,390                                    CUBI  151
  390 EPS1=0.0                                                          CUBI  152
      GO TO 430                                                         CUBI  153
  400 IF(COSEP1+0.999999)410,420,420                                    CUBI  154
  410 EPS1=0.0                                                          CUBI  155
      GO TO 430                                                         CUBI  156
  420 EPS1=ARCOS(COSEP1)                                                CUBI  157
  430 COSEP2=(X3X2*X3X1+Y3Y2*Y3Y1+Z3Z2*Z3Z1)/(L32*L31)                  CUBI  158
      IF(COSEP2-0.999999)450,450,440                                    CUBI  159
  440 EPS2=0.0                                                          CUBI  160
      GO TO 480                                                         CUBI  161
  450 IF(COSEP2+0.999999)460,470,470                                    CUBI  162
  460 EPS2=0.0                                                          CUBI  163
      GO TO 480                                                         CUBI  164
  470 EPS2=ARCOS(COSEP2)                                                CUBI  165
  480 DELTA=EPS1+EPS2                                                   CUBI  166
      TX=L1*COS(DELTA)+LN*SIN(DELTA)                                    CUBI  167
      TY=M1*COS(DELTA)+MN*SIN(DELTA)                                    CUBI  168
      TZ=N1*COS(DELTA)+MN*SIN(DELTA)                                    CUBI  169
C                                                                       CUBI  170
C  CALCULATE END POINT DERIVATIVES                                      CUBI  171
      IF(IFLAG1)490,490,500                                             CUBI  172
  490 X1V00=TX*S                                                        CUBI  173
      Y1V00=TY*S                                                        CUBI  174
      Z1V00=TZ*S                                                        CUBI  175
      J1=NPTS(NB)-2                                                     CUBI  176
      J2=NPTS(NB)-1                                                     CUBI  177
      J3=NPTS(NB)                                                       CUBI  178
      IFLAG1=1                                                          CUBI  179
      GO TO 380                                                         CUBI  180
```

```
  500 X1V01=TX*S                                                        CUBI  181
      Y1V01=TY*S                                                        CUBI  182
      Z1V01=TZ*S                                                        CUBI  183
C                                                                       CUBI  184
C                                                                       CUBI  185
C ********CALCULATE CONSTANTS FOR BOUNDARY CURVE                        CUBI  186
      D(NB,1)=2.0*(XB1(NB,2)-XB1(NB,J2))+X1V00+X1V01                    CUBI  187
      D(NB,2)=3.0*(XB1(NB,J2)-XB1(NB,2))-2.0*X1V00-X1V01                CUBI  188
      D(NB,3)=X1V00                                                     CUBI  189
      D(NB,4)=2.0*(YB1(NB,2)-YB1(NB,J2))+Y1V00+Y1V01                    CUBI  190
      D(NB,5)=3.0*(YB1(NB,J2)-YB1(NB,2))-2.0*Y1V00-Y1V01                CUBI  191
      D(NB,6)=Y1V00                                                     CUBI  192
      D(NB,7)=2.0*(ZB1(NB,2)-ZB1(NB,J2))+Z1V00+Z1V01                    CUBI  193
      D(NB,8)=3.0*(ZB1(NB,J2)-ZB1(NB,2))-2.0*Z1V00-Z1V01                CUBI  194
      D(NB,9)=Z1V00                                                     CUBI  195
      NB=NB+1                                                           CUBI  196
      IF (NB-4)360,360,510                                              CUBI  197
C                                                                       CUBI  198
C ********CALCULATE PATCH DATA                                          CUBI  199
  510 NOW=NOW/2*2+1                                                     CUBI  200
      DELU=1.0/FLOAT(NOU)                                               CUBI  201
      DELW=1.0/FLOAT(NOW)                                               CUBI  202
      NOU=NOU+1                                                         CUBI  203
      NOW=NOW+1                                                         CUBI  204
      STATT=0                                                           CUBI  205
      U=0.0                                                             CUBI  206
C                                                                       CUBI  207
      DO 560 I=1,NOU                                                    CUBI  208
      STAT=1                                                            CUBI  209
      INU=0                                                             CUBI  210
      W=0.0                                                             CUBI  211
C                                                                       CUBI  212
      DO 550 K=1,NOW                                                    CUBI  213
C                                                                       CUBI  214
C                                                                       CUBI  215
      W3=W**3                                                           CUBI  216
      W2=W**2                                                           CUBI  217
      U3=U**3                                                           CUBI  218
      U2=U**2                                                           CUBI  219
C  CALCULATE BLENDING FUNCTIONS                                         CUBI  220
      F1U=3.0*U2-2.0*U3                                                 CUBI  221
      F0U=1.0-3.0*U2+2.0*U3                                             CUBI  222
      F1W=3.0*W2-2.0*W3                                                 CUBI  223
      F0W=1.0-3.0*W2+2.0*W3                                             CUBI  224
C                                                                       CUBI  225
```

```
C  CALCULATE POINTS ON BOUNDARY CURVES
      XOW=D(1,1)*W3+D(1,2)*W2+D(1,3)*W+XB1(1,2)
      YOW=D(1,4)*W3+D(1,5)*W2+D(1,6)*W+YB1(1,2)
      ZOW=D(1,7)*W3+D(1,8)*W2+D(1,9)*W+ZB1(1,2)
      X1W=D(2,1)*W3+D(2,2)*W2+D(2,3)*W+XB1(2,2)
      Y1W=D(2,4)*W3+D(2,5)*W2+D(2,6)*W+YB1(2,2)
      Z1W=D(2,7)*W3+D(2,8)*W2+D(2,9)*W+ZB1(2,2)
      XUO=D(3,1)*U3+D(3,2)*U2+D(3,3)*U+XB1(3,2)
      YUO=D(3,4)*U3+D(3,5)*U2+D(3,6)*U+YB1(3,2)
      ZUO=D(3,7)*U3+D(3,8)*U2+D(3,9)*U+ZB1(3,2)
      XU1=D(4,1)*U3+D(4,2)*U2+D(4,3)*U+XB1(4,2)
      YU1=D(4,4)*U3+D(4,5)*U2+D(4,6)*U+YB1(4,2)
      ZU1=D(4,7)*U3+D(4,8)*U2+D(4,9)*U+ZB1(4,2)
      NPT1=NPTS(1)-1
      NPT2=NPTS(2)-1
C
C  CALCULATE POSITION OF A POINT ON THE SURFACE
     OXS  =  XOW*FOU+X1W*F1U+XUO*FOW+XU1*F1W-XB1(1,2)*FOU*FOW-XB1(1,NPT1)*
     1   FOU*F1W-XB1(2,2)*F1U*FOW-XB1(2,NPT2)*F1U*F1W
     OYS  =  YOW*FOU+Y1W*F1U+YUO*FOW+YU1*F1W-YB1(1,2)*FOU*FOW-YB1(1,NPT1)*
     1   FOU*F1W-YB1(2,2)*F1U*FOW-YB1(2,NPT2)*F1U*F1W
     OZS  =  ZOW*FOU+Z1W*F1U+ZUO*FOW+ZU1*F1W-ZB1(1,2)*FOU*FOW-ZB1(1,NPT1)*
     1   FOU*F1W-ZB1(2,2)*F1U*FOW-ZB1(2,NPT2)*F1U*F1W
C
      IF(INU-1)520,530,530
  520 XXS = XS
      YYS = YS
      ZZS = ZS
      INU=1
      GO TO 540
  530 IF (I.EQ.NOU .AND. K.EQ.NOW .AND. LAST.EQ.1) STATT = 3
      IF (I.EQ.NOU .AND. K.EQ.NOW .AND. LAST.EQ.3) STATT = 3
      IF (I.EQ.NOU .AND. K.EQ.NOW .AND. LAST.EQ.4) STATT = 3
      IF (I.EQ.1 .AND. K.EQ.2 .AND. ISOVR.EQ.0) STAT = 2
      CALL PUNCH (XXS,YYS,ZZS,STAT,XS,YS,ZS,STATT,SECT,TYPE,LINE,SEQ,
     1     LAST,IPRINT,NREC)
      INU=0
      STAT=0
  540 W=W+DELW
  550 CONTINUE
      U=U+DELU
  560 CONTINUE
C
      IF (STATT .NE. 3) GO TO 20
  570 GO TO 590
```

```
  580 ERROR = 1
C 590 RETURN
  590 CONTINUE
      END
```

```
CUBI   271
CUBI   272I
CUBI   273C
CUBI   274
```

```
      OVERLAY (MARK4,1,2)                                               ELLI  001C
      PROGRAM ELLIP                                                     ELLI  002C
C     SUBROUTINE ELLIP                                                  ELLI  003I
C     THIS SUBROUTINE PREPARES THE REQUIRED SURFACE ELEMENTS FOR        ELLI  004
C     CIRCULAR OR ELLIPTICAL ARC SECTIONS.  EACH CROSS-SECTION          ELLI  005
C     IS CONSIDERED SEPARATELY. DUMMY POINTS ARE COMPUTED SO THAT EACH  ELLI  006
C     SECTION IS FORCED TO HAVE AN EVEN NUMBER OF POINTS AND SO THAT    ELLI  007
C     POINTS IN A ROW ARE CORRECTLY MATCHED WITH POINTS IN AN ADJACENT  ELLI  008
C     ROW WHEN THESE ROWS CONTAIN AN UNEQUAL NUMBER OF POINTS.          ELLI  009
C                                                                       ELLI  010
C  THE PARAMETER DISCON WHICH IS SPECIFIED BY THE PROGRAMMER IS VALUED  ELLI  011
C     DEPENDING ON HOW THE POINTS ARE TO BE MATCHED                     ELLI  012
C          DISCON= 1  ALL THETAC AND THETAL ARE THE SAME. DELTHE MUST   ELLI  013
C                     DIVIDE THE ANGULAR INCREMENT THETAL - THETAO      ELLI  014
C                     EVENLY.                                           ELLI  015
C                 = 2  ALL THETAL ARE EQUAL BUT THETAO VARIES           ELLI  016
C                 = 3  ALL THETAO ARE EQUAL BUT THETAL VARIES           ELLI  017
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR                                ELLI  018
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   ELLI  019
     1            TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                         ELLI  020
      DIMENSION TITLE(15),AX(100),THETOX(100),THETLX(100),DELTHX(100),  ELLI  021
     1   NN(100),SECT(1),DELZX(100),DELYX(100),AA(100),BP(100)          ELLI  022
      INTEGER  STAT,STATT,STATD,STATC,ERROR                             ELLI  023
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,       ELLI  024
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                            ELLI  025
      INTEGER  STATA, STATB,PAGE,SEQ,TYPE,CASE,DISCON                   ELLI  026
      RADD(BBI,AAI,THP) = SQRT(BBI*BBI*COS(THP)*COS(THP) +              ELLI  027
     1                         AAI*AAI*SIN(THP)*SIN(THP) )              ELLI  028
      WRITE (TAPEOT,10)                                                 ELLI  029
   10 FORMAT (1H1,/////,1H0,33HELLIPTICAL GEOMETRY DATA IS BEING        ELLI  030
     1 21H GENERATED **********  )                                      ELLI  031
C     SET COUNTERS                                                      ELLI  032
      TYPE = 3                                                          ELLI  033
      NREC = 0                                                          ELLI  034
C  READ IN TITLE CARD                                                   ELLI  035
   20 READ (TAPEIN,30) (TITLE(L),L=1,12),DISCON,IPRINT,CASE,SECT,ITYPE  ELLI  036
   30 FORMAT (12A4,11X,2I1,3X,I2,1A4,I2)                                ELLI  037
      IF (ITYPE .NE. 4) GO TO 470                                       ELLI  038
      LINE = 100                                                        ELLI  039
      SEQ  = 1                                                          ELLI  040
C  READ IN ALL DATA CARDS FOR THE SECTION                               ELLI  041
      REWIND TAPEB                                                      ELLI  042
      I = 1                                                             ELLI  043
   40 READ (TAPEIN,50) X,THETC,THETL,NN(I),A,B,DELZ,DELY,LAST,ITYPE     ELLI  044
   50 FORMAT (F10.0,2F6.0,I3,2F10.0,2F7.0,I1,10X,I2)                    ELLI  045
```

```
      IF (ITYPE .NE. 5) GO TO 470                                       ELLI  046
      DELTH = (THETL-THETO)/FLOAT(NN(I))                                ELLI  047
      THETO  = THETO /57.2957795                                        ELLI  048
      THETL  = THETL /57.2957795                                        ELLI  049
      DELTH  = DELTH /57.2957795                                        ELLI  050
      AX(I) = X                                                         ELLI  051
      THETOX(I) = THETO                                                 ELLI  052
      THETLX(I) = THETL                                                 ELLI  053
      DELTHX(I) = DELTH                                                 ELLI  054
      AA(I) = A                                                         ELLI  055
      BB(I) = B                                                         ELLI  056
      DELYX(I) = DELY                                                   ELLI  057
      DELZX(I) = DELZ                                                   ELLI  058
      IF (LAST .EQ. 0)  GO TO 60                                        ELLI  059
      N = I                                                             ELLI  060
      GO TO 70                                                          ELLI  061
   60 I = I + 1                                                         ELLI  062
      GO TO 40                                                          ELLI  063
   70 I = 1                                                             ELLI  064
      M = 0                                                             ELLI  065
   80 IF (I .GT. N)  GO TO 110                                          ELLI  066
C                                                                       ELLI  067
      IF (NN(I)-M)100,100,90                                            ELLI  068
   90 M =NN(I)                                                          ELLI  069
  100 I = I + 1                                                         ELLI  070
      GO TO 80                                                          ELLI  071
C                                                                       ELLI  072
  110 GO TO (120,180,270), DISCON                                       ELLI  073
C                                                                       ELLI  074
  120 M = M + 1                                                         ELLI  075
C                                                                       ELLI  076
      DO 170 I=1,N                                                      ELLI  077
C                                                                       ELLI  078
      DO 160 J=1,M                                                      ELLI  079
C                                                                       ELLI  080
      XA = AX(I)                                                        ELLI  081
      THETA = THETOX(I) + (FLOAT(J-1)) * DELTHX(I)                      ELLI  082
      THETAP = ABS(THETA - 1.570796)                                    ELLI  083
      RAD = RADD(BB(I),AA(I),THETAP)                                    ELLI  084
      IF (RAD .NE. 0.0)  RAD = AA(I)*BB(I) / RAD                        ELLI  085
      YA = RAD * SIN(THETA)                                             ELLI  086
      ZA =-RAD * COS(THETA)                                             ELLI  087
      YA = YA + DELYX(I)                                                ELLI  088
      ZA = ZA + DELZX(I)                                                ELLI  089
C                                                                       ELLI  090
```

```
      IF (J .EQ. 1) GO TO 130
      STATA = 0
      GO TO 150
C
  130 IF (I .EQ. 1) GO TO 140
      STATA = 1
      GO TO 150
C
  140 STATA = 2
  150 WRITE (TAPEB) XA,YA,ZA,STATA
  160 CONTINUE
  170 CONTINUE
C
      GO TO 360
C
C
  180 DO 260 I=1,N
C
      LIM = M+1-NN(I)
C
      DO 220 J=1,LIM
      XA = AX(I)
      THETA = THETOX(I)
      THETAP = ABS(THETA - 1.570796)
      RAD = RADD(BB(I),AA(I),THETAP)
      IF (RAD .NE. 0.0)  RAD = AA(I)*BB(I) / RAD
      YA = RAD * SIN(THETA)
      ZA =-RAD * COS(THETA)
      YA = YA + DELYX(I)
      ZA = ZA + DELZX(I)
      IF ( J .EQ. 1)  GO TO 190
      STATA = 0
      GO TO 210
C
  190 IF (I .EQ. 1)  GO TO 200
      STATA = 1
      GO TO 210
C
  200 STATA = 2
  210 WRITE (TAPEB) XA,YA,ZA,STATA
C
  220 CONTINUE
      K=0
      LIM = LIM +1
      NM =M + 2
```

```
      DO 250  J = LIM,NM
      XA = AX(I)
      THETA = THETIX(I)-(FLOAT(NN(I)-K))*DELTHX(I)
      THETAP = ABS(THETA - 1.570796)
      RAD = RADD(BB(I),AA(I),THETAP)
      IF (RAD .NE. 0.0)  RAD = AA(I)*BB(I) / RAD
      YA = RAD * SIN(THETA)
      ZA =-RAD * COS(THETA)
      YA = YA + DELYX(I)
      ZA = ZA + DELZX(I)
C
      IF (J.EQ. 1)  GO TO 230
      STATA = 0
      GO TO 240
C
  230 STATA = 1
C
  240 WRITE (TAPER) XA,YA,ZA,STATA
      K = K + 1
  250 CONTINUE
C
  260 CONTINUE
      M = M + 2
      GO TO 360
C
  270 M = M +2
      DO 350 I = 1,N
      NM =NN(I) +1
C
      DO 310 J = 1,NM
      XA = AX(I)
      THETA = THETOX(I) + (FLOAT(J-1))*DELTHX(I)
      THETAP = ABS(THETA - 1.570796)
      RAD = RADD(BB(I),AA(I),THETAP)
      IF (RAD .NE. 0.0)  RAD = AA(I)*BB(I) / RAD
      YA = RAD * SIN(THETA)
      ZA =-RAD * COS(THETA)
      YA = YA + DELYX(I)
      ZA = ZA + DELZX(I)
      IF (J.EQ.1) GO TO 280
      STATA = 0
      GO TO 300
C
  280 IF (I.EQ.1) GO TO 290
      STATA = 1
```

```
      GO TO 300
C
  290 STATA = 2
  300 WRITE (TAPEB) XA,YA,ZA,STATA
C
  310 CONTINUE
      NM =NM+1
C
      DO 340  J = NM,M
      XA = AX(I)
      THETA = THETLX(I)
      THETAP = ABS(THETA - 1.570796)
      RAD = RADD(BB(I),AA(I),THETAP)
      IF (RAD .NE. 0.0)  RAD = AA(I)*BB(I) / RAD
      YA = RAD * SIN(THETA)
      ZA =-RAD * COS(THETA)
      YA = YA + DELYX(I)
      ZA = ZA + DELZX(I)
      IF (J.EQ.1) GO TO 320
      STATA = 0
      GO TO 330
C
  320 STATA = 1
  330 WRITE (TAPEB) XA,YA,ZA,STATA
C
  340 CONTINUE
C
  350 CONTINUE
C
C
  360 STATA = 3
      IF (LAST.EQ.0 .OR. LAST.EQ.2) STATA = 4
C
      BACKSPACE TAPEB
      READ (TAPEB) XA,YA,ZA,STAT
      BACKSPACE TAPEB
      WRITE (TAPEB) XA,YA,ZA,STATA
C
      REWIND TAPEB
      K = 1
C
  370 READ (TAPEB) X,Y,Z,STAT
C
      IF ( STAT .GT. 2)      GO TO 410
      READ (TAPEB) XX,YY,ZZ,STATT
```

```
C
      IF ( STATT .GT. 2)     GO TO 390
C
  380 CALL PUNCH (X,Y,Z,STAT,XX,YY,ZZ,STATT,SECT,TYPE,LINE,SEQ,
     1  LAST,IPRINT,NREC)
C
      GO TO (370,20,450), K
C
C
  390  IF (STATT.EQ. 3) GO TO 400
       STATT = 0
       K = 2
       GO TO 380
C
  400 K = 3
       GO TO 380
C
C
  410 XB = X
       YB = Y
       ZB = Z
       STATB = STAT
       GO TO 430
  420 BACKSPACE TAPEB
  430 BACKSPACE TAPEB
       READ (TAPEB) XA,YA,ZA,STATA
       IF (STATA.EQ.1 .OR. STATA.EQ.2) GO TO 440
       GO TO 420
  440 STATC = 0
       CALL PUNCH (XB,YB,ZB,STATC,XA,YA,ZA,STATA,SECT,TYPE,LINE,SEQ,
     1      LAST,IPRINT,NREC)
       STATD = 0
       IF (STAT .EQ. 3)  STATD = 3
       READ (TAPEB) XC,YC,ZC,STATT
       READ (TAPEB) XD,YD,ZD,STATT
       CALL PUNCH (XC,YC,ZC,STATC,XD,YD,ZD,STATD,SECT,TYPE,LINE,SEQ,
     1      LAST,IPRINT,NREC)

       IF (STAT .NE. 3)  GO TO 20
C
C
  450 CONTINUE
  460 GO TO 480
  470 ERROR = 1
C 480 RETURN
```

```
430 CONTINUE                                    ELLI   2710
    END                                         ELLI   272
```

```
C     SUBROUTINE EROR                                                   EROR  001I
C                                                                       EROR  002
C ***********************************************************************EROR  003
C  THIS ROUTINE IS USED TO CAUSE AN ABEND DUMP FOR SYSTEM ERRORS        EROR  004
C  AS DIRECTED BY THE FIRST DATA CARD.                                  EROR  005
C ***********************************************************************EROR  006
C                                                                       EROR  007
C     DIMENSION A(2)                                                    EROR  008I
C                                                                       EROR  009I
C   SET A PARAMETER THAT WILL USUALLY CAUSE AN ABEND                    EROR  010
C      I = 100000                                                       EROR  011I
C      A(I) = 0.0                                                       EROR  012I
C     IF (I .GT. 1) STOP                                                EROR  013I
C                                                                       EROR  014I
C     RETURN                                                            EROR  015I
C     END                                                               EROR  016I
```

```
      SUBROUTINE FFSURF                                                 FFSU  001
C                                                                       FFSU  002
C                                                                       FFSU  003
C   THIS ROUTINE IS USED TO TRANSFER SURFACE DATA (GENERATED AND        FFSU  004
C   STORED ON UNIT 4 BY FORCE) TO THE FLOW FIELD UNIT 10.               FFSU  005
C                                                                       FFSU  006
C                                                                       FFSU  007
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR                                FFSU  008
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   FFSU  009
     1            TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                         FFSU  010
      COMMON /FFIELD/TITLEM(10),TITLES(10),TITLEA(10),TITLER(10),       FFSU  011
     1        IMTAB(9),IDTYP(5),LOSET(5),LCAB(20),LORG(20),L,           FFSU  012
     2        DINF(6),IRW,LCNEXT,IREG,MACH,NAB,ALPHA,BETA,NREG,         FFSU  013
     3        NSREG,ITFLAG                                              FFSU  014
      COMMON/SUBIDX/SIDX04(1001),SIDX16(1001),SIDX21(1001)              FFSU  015C
      COMMON /MASTER/ MIDX04(21),INDEX9(501),INDX10(2001),INDX15(2001), FFSU  016C
     1  MIDX16(94),INDX20(2421),MIDX21(73)                              FFSU  017C
C                                                                       FFSU  018
      DIMENSION TITLE(15),IFC(5),IFD(5),LOSF(5),DAT(6),DATA(12),DATB(25) FFSU  019
      DIMENSION  IPANL(10),E(25),EF(25),ELEM(25),E2(25),E1(25)          FFSU  020
      DIMENSION LOFF(5),LOCD(5)                                         FFSU  021
C                                                                       FFSU  022
      INTEGER ERROR,PAGE                                                FFSU  023
      INTEGER TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,        FFSU  024
     1        TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                             FFSU  025
      INTEGER SIDX04,SIDX16,SIDX21                                      FFSU  026C
C                                                                       FFSU  027
C                                                                       FFSU  028
      DATA DATA/12*0.0/                                                 FFSU  029
C                                                                       FFSU  030
C   CHECK READ OR WRITE FLAG                                            FFSU  031
      IF (IRW .NE. 0) GO TO 227                                         FFSU  032
C                                                                       FFSU  033
C   READ FLOW FIELD CONTROL AND PLANE ORIENTATION CARD                  FFSU  034
      READ (TAPEIN,20) NSREG, IFC1, ITFLAG, DAT                         FFSU  035
   20  FORMAT (I1,I4,I1,4X,6F10.0)                                      FFSU  036
C                                                                       FFSU  037
C   SET SUB-REGION COUNTERS                                             FFSU  038
      IF (NSREG .LE. 0) NSREG = 1                                       FFSU  039
      ISR = 0                                                           FFSU  040
C                                                                       FFSU  041
      IG10 = LORG(IREG) + 1                                             FFSU  042
      CALL WRITMS (10,DAT,6,IG10)                                       FFSU  043C
C     WRITE (10'IG10) DAT                                               FFSU  044I
      IG10 = LCNEXT + NSREG - 1                                         FFSU  045
```

```
C
   25 ISR = ISR + 1
      LCNEXT = LORG(IREG) + 4 + ISR
C
      DO 30 I=1,5
       LOSF(I) = 0
       LOFF(I) = 0
       LOCD(I) = 0
      IFC(I) = 0
       DATB(I+5) = 0
   30  DATB(I+15) = 0
      IFC(1) = IFC1
C
      LOFF(1) = IG10 + 1
      IC = 1
      IDTYP(1) = 2
      IDTYP(2) = 2
C
C
C  CHECK IF DATA IS TO BE OBTAINED FROM INPUT UNIT OR UNIT 4
      IF (IFC(1) .EQ. 0) GO TO 39
C  READ IN DATA FROM INPUT UNIT
      IFR = IFC(1)
C
C     SPECIAL TAPE INPUT FROM UNIT NTAPE DEFINED IN SUB AERO.
      IF (ITFLAG .EQ. 0)  GO TO 32
      II=0
      DO 31 I=1,IFR
      READ (TAPEIN,35) X,Y,Z,CP
      II=II+1
      DATA(1)=X
      DATA(2)=Y
      DATA(3)=Z
      DATA(11)=CP
      IG10=IG10+1
      CALL WRITMS (10,DATA,12,IG10)
C     WRITE (10'IG10) DATA
   31 CONTINUE
      IFR=II
      IF (IFR.NE.0) GO TO 37
C     WRITE ERROR MESSAGE AND ABEND JOB
      WRITE (TAPEOT,999) IREG
  999 FORMAT ( 1X, 16H**ERROR** REGION, I5, 20H HAS NO INPUT DATA**)
C     CALL ERROR
      STOP
```

```
C
   32 CONTINUE
      DO 38 I=1,IFR
      READ (TAPEIN,35) DATA
   35  FORMAT (6F10.0)
      IG10 = IG10 + 1
      CALL WRITMS (10,DATA,12,IG10)
C     WRITE (10'IG10) DATA
   38  CONTINUE
C
   37 CONTINUE
C
      K = IFR
      GO TO 140
C
C
C  READ PANEL NUMBERS TO BE GROUPED TO FORM SURFACE COMPONENT
   39 CONTINUE
      READ (TAPEIN,40) IPANL
   40  FORMAT (10I2)
C  OBTAIN POINTER INFO FROM UNIT 4
      IG4 = 2
C     READ (4'IG4) E
      NSUB4=1
      NSREC4=2
      CALL STINDX (4,MIDX04,21)
      CALL READMS (4,SIDX04,1001,NSUB4)
      CALL STINDX (4,SIDX04,1001)
      CALL READMS (4,E,25,NSREC4)
      NREM = E(4)
      K = 0
C
C  START PANEL DO LOOP
      DO 100 I=1,10
      IF (IPANL(I) .EQ. 0) GO TO 140
      WRITE (TAPEOT,50) IPANL(I)
   50 FORMAT (1H0,49HSURFACE DATA ARE BEING TRANSFERED FROM UNIT 4 TO ,
     1  17HUNIT 10.  IPANL =,I3,/1H ,28HDATA ARRAY FOLLOWS FOR EACH ,
     2    6HPOINT. )
      IG4 = IPANL(I) * 5
C     READ (4'IG4) EP
      NSUB=(IG4-1)/1000+1
      NSREC4=IG4-(NSUB-1)*1000
      IF (NSUB.EQ.NSUB4) GO TO 1
      CALL STINDX (4,MIDX04,21)
```

```
      CALL READMS (4,SIDX04,1001,NSUB)                                  FFSU  136C
      CALL STINDX (4,SIDX04,1001)                                       FFSU  137C
      NSUB4=NSUB                                                        FFSU  138C
    1 CALL READMS (4,EP,25,NSREC4)                                      FFSU  139C
      ISTART = EP(3) -NREM                                              FFSU  140
      N = EP(4)                                                         FFSU  141
C                                                                       FFSU  142
C   START ELEMENT DO LOOP WITHIN A PANEL                                FFSU  143
      DO 99 II=1,N                                                      FFSU  144
      IG4 = ISTART + NREM*II                                            FFSU  145
C     READ (4'IG4) ELEM                                                 FFSU  146I
      NSUB=(IG4-1)/1000+1                                               FFSU  147C
      NSREC4=IG4-(NSUB-1)*1000                                          FFSU  148C
      IF (NSUB.EQ.NSUB4) GO TO 2                                        FFSU  149C
      CALL STINDX (4,MIDX04,21)                                         FFSU  150C
      CALL READMS (4,SIDX04,1001,NSUB)                                  FFSU  151C
      CALL STINDX (4,SIDX04,1001)                                       FFSU  152C
      NSUB4=NSUB                                                        FFSU  153C
    2 CALL READMS (4,ELEM,25,NSREC4)                                    FFSU  154C
C                                                                       FFSU  155
      IG41 = IG4 + 1                                                    FFSU  156
C                                                                       FFSU  157
C     READ (4'IG41) E2                                                  FFSU  158I
      NSUB=(IG41-1)/1000+1                                              FFSU  159C
      NSREC4=IG41-(NSUB-1)*1000                                         FFSU  160C
      IF (NSUB.EQ.NSUB4) GO TO 3                                        FFSU  161C
      CALL STINDX (4,MIDX04,21)                                         FFSU  162C
      CALL READMS (4,SIDX04,1001,NSUB)                                  FFSU  163C
      CALL STINDX (4,SIDX04,1001)                                       FFSU  164C
      NSUB4=NSUB                                                        FFSU  165C
    3 CALL READMS (4,E2,25,NSREC4)                                      FFSU  166C
      IG4S = E2(NAB)                                                    FFSU  167
C                                                                       FFSU  168
C     READ (4'IG4S) E1                                                  FFSU  169I
      NSUB=(IG4S-1)/1000+1                                              FFSU  170C
      NSREC4=IG4S-(NSUB-1)*1000                                         FFSU  171C
      IF (NSUB.EQ.NSUB4) GO TO 4                                        FFSU  172C
      CALL STINDX (4,MIDX04,21)                                         FFSU  173C
      CALL READMS (4,SIDX04,1001,NSUB)                                  FFSU  174C
      CALL STINDX (4,SIDX04,1001)                                       FFSU  175C
      NSUB4=NSUB                                                        FFSU  176C
    4 CALL READMS (4,E1,25,NSREC4)                                      FFSU  177C
C                                                                       FFSU  178
      DATA(1) = ELEM(7)                                                 FFSU  179
      DATA(2) = ELEM(8)                                                 FFSU  180
```

```
      DATA(3) = ELEM(9)                                                 FFSU  181
      DATA(4) = 0.0                                                     FFSU  182
      DATA(5) = 0.0                                                     FFSU  183
      DATA(6) = 0.0                                                     FFSU  184
      DATA(7) = E1(7)                                                   FFSU  185
      DATA(8) = E1(8)                                                   FFSU  186
      DATA(9) = E1(9)                                                   FFSU  187
      DATA(10) = E1(10)                                                 FFSU  188
      DATA(11) = E1(11)                                                 FFSU  189
      DATA(12) = E1(12)                                                 FFSU  190
      WRITE (TAPEOT,52) II,IG4S,DATA                                    FFSU  191
   52  FORMAT (1H ,2I5,3X,6E12.5,/1H ,13X,6E12.5)                       FFSU  192
C                                                                       FFSU  193
      IG10 = IG10 + 1                                                   FFSU  194
C                                                                       FFSU  195
      CALL WRITMS (10,DATA,12,IG10)                                     FFSU  196C
C     WRITE (10'IG10) DATA                                              FFSU  197I
      K = K + 1                                                         FFSU  198
C                                                                       FFSU  199
C                                                                       FFSU  200
   99 CONTINUE                                                          FFSU  201
  100 CONTINUE                                                          FFSU  202
  140 CONTINUE                                                          FFSU  203
      IFC(1) = K                                                        FFSU  204
C                                                                       FFSU  205
C  SAVE LOCATIONS AND COUNTERS FOR SUB-REGION                           FFSU  206
      DO 260 I = 1,5                                                    FFSU  207
      DATB(I) = LOFF(I)                                                 FFSU  208
      DATB(I+5) = IFC(I)                                                FFSU  209
      DATB(I+20) = LOSF(I)                                              FFSU  210
  260 DATB(I+10) = LOCD(I)                                              FFSU  211
C                                                                       FFSU  212
C                                                                       FFSU  213
      CALL WRITMS (10, DATB,25,LCNEXT)                                  FFSU  214C
C     WRITE(10'LCNEXT)  DATB                                            FFSU  215I
C                                                                       FFSU  216
      LCNEXT = IG10 + 1                                                 FFSU  217
      IF (ISR .LT. NSREG) GO TO 25                                      FFSU  218
      GO TO 600                                                         FFSU  219
C                                                                       FFSU  220
C  FLOW FIELD DATA WILL BE READ FROM UNIT 10                            FFSU  221
  227 WRITE (TAPEOT,217)                                                FFSU  222
  217 FORMAT (1H0,48HSURFACE DATA ARE BEING READ FROM FLOW-FIELD UNIT )  FFSU  223
      WRITE (TAPEOT,214) TITLEM,IMTAB, TITLES,MACH,NAB,LOAB,            FFSU  224
     1  TITLEA,ALPHA,BETA,NREG,LORG, TITLER,ICTYP                       FFSU  225
```

```
  214 FORMAT (1H0,7HTITLEM=,10A4,7H IMTAB=,9I4,/1H ,7HTITLES=,10A4,
     1 7H MACH =,F6.3,5H NAB=,I3,/1H ,4X,5HLOAB=,20I4,/1H ,7HTITLEA=,
     2 10A4,7H ALPHA=,F6.3,6H BETA=,F6.3,8H NREG  =,I3,/1H ,4X,5HLORG=,
     3 20I4,/1H ,7HTITLER=,10A4,7H IDTYP=,5I2)
      IG10 = LORG(IREG) + 1
      CALL READMS(10, DAT, 6, IG10)
C     READ (10'IG10)  DAT
      WRITE(TAPEOT,228) DAT
  228 FORMAT(1H0,
     2  3HXO=,F7.3,3X,3HYO=,F7.3,3X,3HZO=,F7.3,
     3  3X,5HPSIO=,F7.3,3X,6HTHETO=,F7.3,6X,5HPHIO=,F7.3)
C
C  READ IN SUB-REGIONS
      DO 800 ISR = 1,NSREG
      IG10 = LORG(IREG) + 4+ ISR
      ISF = 0
  700 CONTINUE
C
      CALL READMS (10, DATB,25,IG10)
C     READ(10'IG10) DATB
      DO 710 I = 1,5
      LOFF(I) = DATB(I)      + 0.001
      IFC(I) = DATB(I+5)     + 0.001
      LOCD(I) = DATB(I+10)   + 0.001
      IFD(I)  = DATB(I+15)   + 0.001
      LOSF(I) = DATB(I+20)   + 0.001
  710 CONTINUE
C
C
C  COUNT NUMBER OF SECONDARY FLOWS (SUB-REGIONS ONLY)
      IF (ISF .NE. 0)  GO TO 750
      NSECF = 0
  720 CONTINUE
  730 CONTINUE
      WRITE(TAPEOT,740)  IREG, ISR, NSECF
  740 FORMAT(1H1, 11HFLOW REGION, I3/1H0,  10HSUB-REGION, I3,
     1     20X  25HNUMBER OF SECCNDARY FLOWS, I3)
C
  750 WRITE(TAPEOT,760)  LOFF, IFC, LCCD, IFD
  760 FORMAT(1H0, 4HLOFF,  5I20/1H , 4H IFC, 5I20/
     1        1H0, 4HLOCD,  5I20/1H , 4H IFC, 5I20)
      WRITE(TAPEOT,761)  LOSF
  761 FORMAT(1H0, 4HLOSF, 5I20)
C
      IG10 = LOFF(1) - 1
```

```
      IFR = IFC(1)
      IC = 1
  231 WRITE (TAPEOT,237) IC
  237 FORMAT (1H0,9HBOUNDARY ,I2,5H DATA )
  236 DO 232 I=1,IFR
      IG10 = IG10 + 1
      CALL READMS (10,DATA,12,IG10)
C     READ (10'IG10) DATA
  232 WRITE (TAPEOT,233) DATA
  233 FORMAT (1H , 12F10.4)
      IF (IC .NE. 1) GO TO 234
      IC = 2
      IFR = IFC(2)
      IF (IFR .EQ. 0) GO TO 234
      GO TO 231
  234 IF (IC .NE. 2) GO TO 235
      IC = 3
      IFR = IFC(3)
      IF (IFR .EQ. 0) GO TO 235
      GO TO 231
  235 IF (IC .NE. 3) GO TO 239
      IC = 4
      IFR = IFC(4)
      IF (IFR .EQ. 0) GO TO 239
      GO TO 231
  239 IF (IC .NE. 4) GO TO 245
      IC = 5
      IFR = IFC(5)
      IF (IFR .EQ. 0) GO TO 245
      WRITE (TAPEOT,238)
  238 FORMAT (1H0,25HREMAINING FLOW-FIELD DATA  )
      GO TO 236
C
  245 CONTINUE
C
C  CHECK FOR SECONDARY FLOWS
      IF (ISF .EQ. NSECF)  GO TO 800
      ISF = ISF + 1
      WRITE(TAPEOT,770)  ISF
  770 FORMAT(1H1, 14HSECONDARY FLOW, I3)
C
C  SET LOCATIONS
      IG10 = LOSF(ISF)
      GO TO 700
C
```

```
  800 CONTINUE                                                          FFSU  316
C                                                                       FFSU  317
  600 CONTINUE                                                          FFSU  318
C                                                                       FFSU  319
C                                                                       FFSU  320
C                                                                       FFSU  321
      RETURN                                                            FFSU  322
      END                                                               FFSU  323
```

```
      OVERLAY (MARK4,2,1)
      PROGRAM FLOW
      SUBROUTINE FLOW
C
C
C  THIS IS THE EXECUTIVE CONTROL PROGRAM FOR THE FLOW FIELD ANALYSIS
C  ROUTINES
C
C
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      COMMON /TAGS/ITAG4,ITAG9,ITAG10
      COMMON /FFIELD/TITLEM(10),TITLES(10),TITLEA(10),TITLER(10),
     1          IMTAB(9),IDTYP(5),LOSET(5),LOAB(20),LORG(20),L,
     2          DINF(6),IRW,LONEXT,IREG,MACH,NAB,ALPHA,BETA,NREG,
     3          NSREG,ITFLAG
      COMMON /FLIGHT/EACH,ALT,PSTAG,TSTAG,V,RENO,PFS,TFS,AFS,RHOFS,VIS
      COMMON /FSBS/FS(8),BS(8)
      COMMON/SUBIDX/SIDX04(1001),SIDX16(1001),SIDX21(1001)
      COMMON /MASTER/ MIDX04(21),INDEX9(501),INDX10(2001),INDX15(2001),
     1  MIDX16(94),INDX20(2421),MIDX21(73)
C
      DIMENSION TITLE(15),E(12),E2(13),E3(17)
C
      REAL MACH
      INTEGER ERROR, PAGE
      INTEGER TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1        TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      INTEGER SIDX04,SIDX16,SIDX21
C
C
      WRITE (TAPEOT,5)
    5 FORMAT (1H1,23H** FLOW-FIELD OPTION **   )
C
C  MASTER DIRECTORY INFORMATION
C
      READ (TAPEIN,10) MFLAG,TITLEM
   10 FORMAT (1I1,19X,10A4)
C
C  CHECK UNIT STATUS.  MFLAG = 0 SET UP COMPLETE NEW UNIT, MFLAG = 1
C  USE OLD UNIT POINTERS.
      IG10 = 1
      IF (MFLAG .NE. 0) GO TO 30
C
C  INITIALIZE MASTER DIRECTORY TABLE ARRAYS, ETC.
```

```
C    MASTER DIRECTORY NAME
      CALL WRITMS (10,TITLEM,10,IG10)
C     WRITE (10'IG10) TITLEM
      DO 20 I=1,9
   20 IMTAB(I) = 0
      IMTAB(1) = 0
      IMTAB(2) = 4
      IMTAB(5) = 4
      LCNEXT = 4
      LOSET(1) = 4
      GO TO 33
C
C  OLD DATA UNIT. READ POINTERS FROM MASTER DIRECTORY TABLE (IMTAB)
   30 CONTINUE
      CALL READMS(10, TITLEM, 10, IG10)
C     READ(10'IG10)  TITLEM
      IG10 = 2
      CALL READMS (10,IMTAB,9,IG10)
C     READ (10'IG10) IMTAB
      LCNEXT = IMTAB(2)
C
   33 DO 35 I=1,5
   35 LOSET(I) = IMTAB(I+4)
      NDSET = IMTAB(1)
C
C
C  SET DIRECTORY
C
C  INPUT LAST SET FLAG (LASTS), NEW SET FLAG (NEWS), DATA SET NUMBER
C  (NSET), MACH NUMBER, SET TITLE (TITLES)
   40 READ (TAPEIN,50) LASTS,NEWS,NSET,MACH,TITLES
   50 FORMAT (2I1,1X,I1,F6.0,10X,10A4)
C
C  CHECK DATA SET STATUS.
      IF (NEWS .NE. 0) GO TO 70
C
      DO 60 I=1,20
   60 LOAD(I) = 0
      NDSET = 1
      IMTAB(5) = LCNEXT
      LOSET(1) = LCNEXT
      LCNEXT = LCNEXT + 3
      GO TO 90
C
```

```
C  READ POINTERS OF EXISTING SET TO GET TO LOWER DIRECTORY TABLES.
   70 IF (NEWS .NE. 1) GO TO 80
      IG10 = IMTAB(NSET+4)
      CALL READMS (10,E,12,IG10)
C     READ (10'IG1C) E
      DO 71 I=1,1C
   71 TITLES(I) = E(I)
      MACH = E(11)
      NAB = E(12)
      IG10 = IG10 + 1
      CALL READMS (1C,LOAB,20,IG10)
C     READ (10'IG1C) LOAB
      GO TO 90
C
C  ADD NEW SET TABLE DATA TO OLD
   80 NDSET = NDSET + 1
      DO 81 I=1,20
   81 LOAB(I) = 0
      LDSET(NSET) = LCNEXT
      IMTAB(NSET+4) = LCNEXT
      LCNEXT = LCNEXT + 3
C
C
C  INPUT LAST ALPHA-BETA FLAG (LASTAB), NEW ALPHA-BETA SET FLAG
C  (NEWAB), ALPHA-BETA SET NUMBER (IAB), ALPHA-BETA VALUES,
C  ALPHA-BETA SET TITLE (TITLEA).
   90 READ (TAPEIN,100) LASTAB,NEWAB,IAB,ALPHA,BETA,TITLEA
  100 FORMAT (2I1,I2,2F6.0,4X,10A4)
C
C  CHECK ALPHA-BETA SET STATUS
      IF (NEWAB .GT. 0) GO TO 120
C
C  INITIALIZE ALPHA-BETA DIRECTORY
      DO 110 I=1,20
  110 LORG(I) = 0
C
      NAB = 1
      LOAB(IAB) = LCNEXT
      LCNEXT = LCNEXT + 3
      GO TO 140
C
C
C  READ POINTERS OF EXISTING ALPHA-BETA DIRECTORY
  120 IF (NEWAB .NE. 1) GO TO 130
      IG10 = LOAB(IAB)
```

```
      CALL READMS (10,E2,13,IG10)                                       FLOW  136C
C     READ (10'IG10) E2                                                 FLOW  137I
      DO 121 I=1,10                                                     FLOW  138
  121 TITLEA(I) = E2(I)                                                 FLOW  139
      ALPHA = E2(11)                                                    FLOW  140
      BETA = E2(12)                                                     FLOW  141
      NREG = E2(13)                                                     FLOW  142
      IG10 = IG10 + 1                                                   FLOW  143
      CALL READMS (10,LORG,20,IG10)                                     FLOW  144C
C     READ (10'IG10) LORG                                               FLOW  145I
      GO TO 140                                                         FLOW  146
C                                                                       FLOW  147
C                                                                       FLOW  148
C  ADD NEW ALPHA-BETA DIRECTORY                                         FLOW  149
  130 NAB = NAB + 1                                                     FLOW  150
      DO 131 I=1,20                                                     FLOW  151
  131 LORG(I) = 0                                                       FLOW  152
      LOAB(IAB) = LCNEXT                                                FLOW  153
      LCNEXT = LCNEXT + 3                                               FLOW  154
C                                                                       FLOW  155
  140 DO 141 I=1,5                                                      FLOW  156
  141 IDTYP(I) = 0                                                      FLOW  157
C                                                                       FLOW  158
C  INPUT LAST REGION GROUP FLAG (LASTR),NEW REGION SET FLAG             FLOW  159
C  (NEWR), REGION SET NUMBER (IREG), DATA TYPE FLAGS (IDTYP),           FLOW  160
C  REGION TITLE (TITLER).                                               FLOW  161
      READ (TAPEIN,150) LASTR,NEWR,IREG,IDTYP(1),ISORCE,IRW,TITLER      FLOW  162
  150 FORMAT (2I1,I2,3I1,13X,10A4)                                      FLOW  163
C  CHECK REGION SET STATUS                                              FLOW  164
      IF (NEWR .NE. 0) GO TO 170                                        FLOW  165
C                                                                       FLOW  166
C  INITIALIZE FLOW FIELD DATA                                           FLOW  167
      NREG = 1                                                          FLOW  168
      NSREG = 0                                                         FLOW  169
      ITFLAG = 0                                                        FLOW  170
      IG10 = LCNEXT                                                     FLOW  171
      DO 161 I=1,10                                                     FLOW  172
  161 E3(I) = TITLER(I)                                                 FLOW  173
      DO 162 I=1,5                                                      FLOW  174
  162 E3(I+10) = IDTYP(I)                                               FLOW  175
      E3(16) = NSREG                                                    FLOW  176
      E3(17) = ITFLAG                                                   FLOW  177
      CALL WRITMS (10,E3,17,IG10)                                       FLOW  178C
C     WRITE (10'IG10) E3                                                FLOW  179I
      LORG(IREG) = IG10                                                 FLOW  180
```

```
      LCNEXT = IG10 + 5
      GO TO 190
C
C  READ POINTERS OF EXISTING FLOW FIELD DATA SET
 170  IF (NEWR .NE. 1) GO TO 180
      IG10 = LORG(IREG)
      CALL READMS (10,E3,17,IG10)
C     READ (10'IG10) E3
      DO 171 I=1,10
  171 TITLER(I) = E3(I)
      DO 172 I=1,5
  172 IDTYP(I) = E3(I+10)
      NSREG = E3(16)
      ITFLAG = E3(17)
      GO TO 190
C
C  ADD NEW FLOW FIELD TABLE ITEM
  180 NSREG = 0
      ITFLAG = 0
      IG10 = LCNEXT
      DO 182 I=1,10
  182 E3(I) = TITLER(I)
      DO 183 I=1,5
  183 E3(I+10) = IDTYP(I)
      E3(16) = NSREG
      E3(17) = ITFLAG
      CALL WRITMS (10,E3,17,IG10)
C     WRITE (10'IG10) E3
      NREG = NREG + 1
      LORG(IREG) = IG10
      LCNEXT = IG10 + 5
C
C
 190  CONTINUE
C  SELECT TYPE OF DATA TO BE READ IN OR GENERATED
C     IDTYP(1)   = 1 (FLOW FIELD),      =2 (SURFACE DATA),
C                = 3 (STREAMLINES),     =4 (SHOCK SHAPE)
      IF (IDTYP(1).EQ.2) GO TO 220
      WRITE (TAPEOT,202) IDTYP(1)
  202  FORMAT (1H0,18H**IDTYP(1) INPUT =,I5,17H IS OUT OF RANGE. ,
     1   14H PROGRAM STOP.  )
      STOP
C
C  SURFACE VELOCITY VECTOR DATA
  220 CALL FFSURF
```

```
C
      IF (IRW .NE. 0) GO TO 612
C  COMPLETE FLOW FIELD DATA TABLE
      IG10 = LORG(IREG)
      DO 601 I=1,10
  601 E3(I) = TITLER(I)
      DO 602 I=1,5
  602 E2(I+10) = IDTYP(I)
      E3(16) = NSREG
      E3(17) = ITFLAG
      CALL WRITMS (10,E3,17,IG10)
C     WRITE (10'IG10) E3
      WRITE (TAPEOT,610) TITLER,IDTYP,NSREG,ITFLAG
 610  FORMAT (1H0,5X,10H** TITLER=,10A4,5X,6HIDTYP=,5I2,3X,6HNSREG=,
     1 I3,3X,7HITFLAG=,I3)
C
C  CHECK FOR LAST REGION
 612  IF (LASTR .EQ. 0) GO TO 140
      IF (IRW .NE. 0) GO TO 622
C
C  COMPLETE REGION DIRECTORY
      IG10 = LOAB(IAB)
      DO 613 I=1,10
  613 E2(I) = TITLEA(I)
      E2(11) = ALPHA
      E2(12) = BETA
      E2(13) = NREG
      CALL WRITMS (10,E2,13,IG10)
C     WRITE (10'IG10) E2
      WRITE (TAPEOT,620) TITLEA,ALPHA,BETA,NREG,LORG
 620  FORMAT (1H ,10H** TITLEA=,10A4,7H ALPHA=,F6.3,6H BETA=,F6.3,
     1  8H NGROUP=,I4,/1H ,8H** LORG=,20I4)
      IG10 = IG10 + 1
      CALL WRITMS (10,LORG,20,IG10)
C     WRITE (10'IG10) LORG
C
C
C
C  CHECK FOR LAST ALPHA-BETA
 622  IF (LASTAB .EQ. 0) GO TO 90
      IF (IRW .NE. 0) GO TO 632
C
C  COMPLETE SET DIRECTORY
      IG10 = LOSET(NSET)
      DO 623 I=1,10
```

```
  623 E(I) = TITLES(I)                                                  FLOW   271
      E(11) = MACH                                                      FLOW   272
      E(12) = NAB                                                       FLOW   273
      CALL WRITMS (10,E,12,IG10)                                        FLOW   274C
C     WRITE (10'IG10) E                                                 FLOW   275I
      WRITE (TAPEOT,630) TITLES,MACH,NAB,LOAB                           FLOW   276
 630  FORMAT (1H ,10H** TITLES=,10A4,6H MACH=,F6.3,5H NAB=,I4,/1H ,     FLOW   277
     1  8H** LOAB=,20I4)                                                FLOW   278
      IG10 = IG10 + 1                                                   FLOW   279
      CALL WRITMS (10,LOAB,20,IG10)                                     FLOW   280C
C     WRITE (10'IG10) LOAB                                              FLOW   281I
C                                                                       FLOW   282
C  CHECK FOR LAST MASTER SET                                            FLOW   283
 632  IF (LASTS .EQ. 0) GO TO 40                                        FLOW   284
      IF (IRW .NE. 0) GO TO 650                                         FLOW   285
      IMTAB(1) = NDSET                                                  FLOW   286
      IMTAB(2) = ICNEXT                                                 FLOW   287
      IG10 = 2                                                          FLOW   288
      CALL WRITMS (10,IMTAB,9,IG10)                                     FLOW   239C
C     WRITE (10'IG10) IMTAB                                             FLOW   290I
      WRITE (TAPEOT,640) IMTAB                                          FLOW   291
 640  FORMAT (1H ,8H**IMTAB=,9I4)                                       FLOW   292
C                                                                       FLOW   293
C                                                                       FLOW   294
C650  RETURN                                                            FLOW   2951
  650 CONTINUE                                                          FLOW   296C
      END                                                               FLOW   297
```

```
      SUBROUTINE FORCE (NEXT,IG4S,NSUB4)
C
C
C*********************************************************************
C** THIS SUBROUTINE CALCULATES THE PRESSURE COEFFICIENT ON EACH SURFACE**
C**ELEMENT, RESOLVES IT INTO THE REQUIRED DIRECTIONS, AND ADDS UP THE **
C**CONTRIBUTIONS TO FIND THE TOTAL VEHICLE COEFFICIENTS.              **
C*********************************************************************
C
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR
      COMMON /FLIGHT/MACH,ALT,PSTAG,TSTAG,V,RENO,PFS,TFS,AFS,RHOFS,VIS
      COMMON /GDATA/LTOT,J,SYMFCT,ICRN,IGTYPE,L,LTOTAB(20)
      COMMON /REF/SREF,MAC,SPAN,XCG,YCG,ZCG
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      COMMON /ABDATA/NAB,ALPHA(20),BETA(20),ROL(20),CDELTA(20),QI(20),
     1               RI(20),PI(20)
      COMMON /FARRAY/F
      COMMON /GASD/GAM,GASCP,PRAN,IGAS,AV1,AV2,AV3,GTYPE(2)
      COMMON /MDATA/IMPACT,ISHAD,IPRINT,PDATA,ISAVE,IPIN
      COMMON /FSBS/FS,BS
      COMMON/SUBIDX/SIDX04(1001),SIDX16(1001),SIDX21(1001)
      COMMON /MASTER/ MIDX04(21),INDEX9(501),INDX10(2001),INDX15(2001),
     1  MIDX16(94),INDX20(2421),MIDX21(73)
      DIMENSION ANGLE(3),FS(8),BS(8)
      DIMENSION F(11),PDATA(6),E(25),TITLE(15),DINFL(6),LCOND(3),FSS(8)
      DIMENSION IG4S(1)
      DIMENSION E1(25),E2(25),E3(25),INFLOW(5)
      INTEGER  ERROR,SYMFCT,SYMFCC,CASE
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      INTEGER SIDX04,SIDX16,SIDX21
      REAL  MACH,MAC,NX,NY,NZ,MACHSQ
C
C
      RAD(A) = A /.572957795E+02
C   SET UP STARTING CONSTANTS
      MER = 0
      IFIRST = 0
      CA  = 0.0
      CY  = 0.0
      CN  = 0.0
      CLL = 0.0
      CLM = 0.0
      HML = 0.0
```

```
      HMR  = 0.0                                                        FORC  046
      CLN  = 0.0                                                        FORC  047
      DELTAS = 0.0                                                      FORC  048
      ARFAT = 0.0                                                       FORC  049
      NPRT = 51                                                         FORC  050
      NPCK = 51                                                         FORC  051
      IDERIV = 0                                                        FORC  052
      IPRCK = 0                                                         FORC  053
      IF (IPRINT .EQ. 2) IPRCK = 1                                      FORC  054
      SYMFCO = 0                                                        FORC  055
      ISHE1 = 0                                                         FORC  056
      CPSTAG = PDATA(1)                                                 FORC  057
      QQINF = PDATA(2)                                                  FORC  058
      ETAC = PDATA(3)                                                   FORC  059
      ENPM = PDATA(4)                                                   FORC  060
      IMPACI = PDATA(5)                                                 FORC  061
      ISHADI = PDATA(6)                                                 FORC  062
C                                                                       FORC  063
C  CALCULATE DIRECTION COSINES OF VELOCITY VECTOR                       FORC  064
      ALPHAR = RAD(ALPHA(J))                                            FORC  065
      BETAR  = RAD(BETA(J))                                             FORC  066
      ROLL = ROL(J)                                                     FORC  067
      PHIR = RAD(ROLL)                                                  FORC  068
      VXO = -V*COS(ALPHAR)*COS(BETAR)                                   FORC  069
      VYO =  V*(SIN(PHIR)*SIN(ALPHAR)*COS(BETAR)+COS(PHIR)*SIN(BETAR))  FORC  070
      VZO =  V*(COS(PHIR)*SIN(ALPHAR)*COS(BETAR)-SIN(PHIR)*SIN(BETAR))  FORC  071
C                                                                       FORC  072
      Q = QI(J)                                                         FORC  073
      R = RI(J)                                                         FORC  074
      P = PI(J)                                                         FORC  075
      LTS = LTOT                                                        FORC  076
      IF (IPIN .GT. 0)  LTS = LTOTAB(J)                                 FORC  077
C                                                                       FORC  078
C  INITIALIZE FREE STREAM DATA ARRAY                                    FORC  079
      FS(1) = RHOFS                                                     FORC  080
      FS(2) = PFS                                                       FORC  081
      FS(3) = TFS                                                       FORC  082
      FS(4) = AFS                                                       FORC  083
      FS(5) = VIS                                                       FORC  084
      FS(6) = MACH                                                      FORC  085
      FS(7) = MACH * FS(4)                                              FORC  086
      FS(8) = FS(1)*FS(7)/FS(5)                                         FORC  087
      GP1 = GAM + 1.0                                                   FORC  088
      GM1= GAM - 1.0                                                    FORC  089
      DO 9 I=1,8                                                        FORC  090
```

```
    9 FSS(I) = FS(I)                                                    FORC  091
      ICKF = 0                                                          FORC  092
      INL = 0                                                           FORC  093
C                                                                       FORC  094
C*****START OF ELEMENT DO LOOP *****                                    FORC  095
   10 DO 589 L=1,LTS                                                    FORC  096
      IF (IPIN .GT. 0)  GO TO 16                                        FORC  097
C   GET GEOMETRY DATA FROM NORMAL STORAGE ARRAYS                        FORC  098
      IG4 = IG4S(L)                                                     FORC  099
C     READ (4'IG4) E                                                    FORC  100I
      NSUB=(IG4-1)/1000+1                                               FORC  101C
      NSREC4=IG4-(NSUB-1)*1000                                          FORC  102C
      CALL STINDX (4,MIDX04,21)                                         FORC  103C
      CALL READMS (4,SIDX04,1001,NSUB)                                  FORC  104C
      CALL STINDX (4,SIDX04,1001)                                       FORC  105C
      NSUB4=NSUB                                                        FORC  106C
      CALL READMS (4,E,25,NSREC4)                                       FORC  107C
      N      = E(2)                                                     FORC  108
      M      = E(3)                                                     FORC  109
      NX     = E(4)                                                     FORC  110
      NY     = E(5)                                                     FORC  111
      NZ     = E(6)                                                     FORC  112
      XCENT  = E(7)                                                     FORC  113
      YCENT  = E(8)                                                     FORC  114
      ZCENT  = E(9)                                                     FORC  115
      AREA   = E(10)                                                    FORC  116
      IG41 = IG4 + 1                                                    FORC  117
      IG42 = IG4 + 2                                                    FORC  118
      GO TO 20                                                          FORC  119
C   READ GEOMETRY DATA FROM FLOW FIELD UNIT                             FORC  120
   16 READ (TAPEF) IG4,LCOND,E,DINFL,INFLOW                             FORC  121
      IF (LCOND(L) .NE. L)  GO TO 589                                   FORC  122
      IG41 = IABS(IG4) + 1                                              FORC  123
      IG42 = IG41 + 1                                                   FORC  124
C   DUMP GEOMETRY DATA INTO WORKING VARIABLES                           FORC  125
   18 N = E(2) + 0.00001                                                FORC  126
      M = E(3) + 0.0001                                                 FORC  127
      NX = E(4)                                                         FORC  128
      NY = E(5)                                                         FORC  129
      NZ = E(6)                                                         FORC  130
      XCENT = E(7)                                                      FORC  131
      YCENT = E(8)                                                      FORC  132
      ZCENT = E(9)                                                      FORC  133
      AREA = E(10)                                                      FORC  134
C   OBTAIN POINTER ARRAYS FOR SURFACE DATA SAVE                         FORC  135
```

```
   20 CONTINUE                                                          FORC  136
C     READ (4'IG41) E2                                                  FORC  137I
      NSUB=(IG41-1)/1000+1                                              FORC  138C
      NSREC4=IG41-(NSUB-1)*1000                                         FORC  139C
      IF (NSUB.EQ.NSUB4) GO TO 1                                        FORC  140C
      CALL STINDX (4,MIDX04,21)                                         FORC  141C
      CALL READMS (4,SIDX04,1001,NSUB)                                  FORC  142C
      CALL STINDX (4,SIDX04,1001)                                       FORC  143C
      NSUB4=NSUB                                                        FORC  144C
    1 CALL READMS (4,E2,25,NSREC4)                                      FORC  145C
C     READ (4'IG42) E3                                                  FORC  146I
      NSUB=(IG42-1)/1000+1                                              FORC  147C
      NSREC4=IG42-(NSUB-1)*1000                                         FORC  148C
      IF (NSUB.EQ.NSUB4) GO TO 2                                        FORC  149C
      CALL STINDX (4,MIDX04,21)                                         FORC  150C
      CALL READMS (4,SIDX04,1001,NSUB)                                  FORC  151C
      CALL STINDX (4,SIDX04,1001)                                       FORC  152C
      NSUB4=NSUB                                                        FORC  153C
    2 CALL READMS (4,E3,25,NSREC4)                                      FORC  154C
C  CHECK ON SYMMETRY REQUIREMENTS (CHANGE SIGNS IF REQUIRED)            FORC  155
      IF (SYMFCO .EQ. 0) GO TO 60                                       FORC  156
      NY = -NY                                                          FORC  157
       YCENT = -YCENT                                                   FORC  158
      INL = 1                                                           FORC  159
   60 CONTINUE                                                          FORC  160
      IF (IPIN .GT. 0)  GO TO 68                                        FORC  161
C                                                                       FORC  162
C  DETERMINE LOCAL EXTERNAL FLOW CONDITIONS                             FORC  163
      IF (L .GT. 1) GO TO 69                                            FORC  164
C                                                                       FORC  165
   68 VXI = VXO                                                         FORC  166
      VYI = VYO                                                         FORC  167
      VZI = VZO                                                         FORC  168
   71 VX = VXI                                                          FORC  169
      VY = VYI                                                          FORC  170
      VZ = VZI                                                          FORC  171
C                                                                       FORC  172
   69 CONTINUE                                                          FORC  173
C                                                                       FORC  174
      IF (Q.EQ.0.0 .AND. R.EQ.0.0 .AND. P.EQ.0.0) GO TO 70              FORC  175
C  CALCULATE VELOCITY COMPONENTS WITH VEHICLE ROTATION                  FORC  176
      VX = VXI+ (Q*(+ZCENT-ZCG)-R*(+YCENT-YCG))                         FORC  177
      VY = VYI+ (R*(+XCENT-XCG)+P*(+ZCENT-ZCG))                         FORC  178
      VZ = VZI- (P*(+YCENT-YCG)+Q*(+XCENT-XCG))                         FORC  179
C                                                                       FORC  180
```

```
   70 VLOCAL = SQRT(VX*VX + VY*VY + VZ*VZ)                              FORC  181
C  COMPUTE COSINE OF ANGLE BETWEEN UNIT VECTORS                         FORC  182
      COSDEL =(-NX*VX - NY*VY - NZ*VZ) / VLOCAL                         FORC  183
      IF (COSDEL.GT.-1.0001 .AND. COSDEL.LT.-1.0    ) COSDEL = -1.0     FORC  184
      IF (COSDEL.GT. 1.0    .AND. COSDEL.LT. 1.0001) COSDEL =  1.0     FORC  185
      IF (COSDEL.GE.-1.0 .AND. COSDEL.LE.1.0) GO TO 90                  FORC  186
      WRITE (6,80)                                                      FORC  187
   80 FORMAT (1H0,44H***** FORCE ROUTINE WILL ATTEMPT TO FIND THE       FORC  188
     162H ARCCOSINE OF AN ARGUMENT WHOSE ABSOLUTE VALUE IS GREATER THAN FORC  189
     21CH ONE *****,/1H ,15X30H*** JOB WILL BE TERMINATED ***  )        FORC  190
      ERROR = 1                                                         FORC  191
      GO TO 620                                                         FORC  192
C                                                                       FORC  193
C  COMPUTE ANGLE BETWEEN UNIT VECTORS                                   FORC  194
   90 THETA = ARCOS(COSDEL)                                             FORC  195
C                                                                       FORC  196
C                                                                       FORC  197
C                                                                       FORC  198
C  COMPUTE NEWTONIAN IMPACT ANGLE                                       FORC  199
      IF (IMPACT.NE.10 .AND. ENPM.NE.0.0) THETA = THETA / ENPM          FORC  200
      DELTAR = 1.570796327- THETA                                       FORC  201
      IF (DELTAR.GT.-0.00001  .AND. DELTAR.LT.0.00001 ) DELTAR = 0.0    FORC  202
C                                                                       FORC  203
C  CALCULATE NEWTONIAN IMPACT ANGLE IN DEGREES                          FORC  204
      DELTA = DELTAR * .572957795E+02                                   FORC  205
      ANGLE(2) = DELTA                                                  FORC  206
C                                                                       FORC  207
      DO 111 I=1,8                                                      FORC  208
  111  BS(I) = 0.0                                                      FORC  209
      TX = NY*VZ - NZ*VY                                                FORC  210
      TY = NZ*VX - NX*VZ                                                FORC  211
      TZ = NX*VY - NY*VX                                                FORC  212
      SX = TY*NZ - TZ*NY                                                FORC  213
      SY = TZ*NX - TX*NZ                                                FORC  214
      SZ = TX*NY - TY*NX                                                FORC  215
      STOTAL = SQRT(SX*SX + SY*SY + SZ*SZ)                              FORC  216
      IF (STOTAL .EQ. 0.0) GO TO 115                                    FORC  217
      SX = SX / STOTAL                                                  FORC  218
      SY = SY / STOTAL                                                  FORC  219
      SZ = SZ / STOTAL                                                  FORC  220
  115 CONTINUE                                                          FORC  221
C                                                                       FORC  222
C  CHECK IF SURFACE DATA TO BE USED                                     FORC  223
C                                                                       FORC  224
C  CP IS AVAILABLE AS DINFL(5)                                          FORC  225
```

```
  116 CP = DINFL(5)
      CPI = CP
      BS(2) = (0.5*GAM*CP*MACH**2 + 1.0)*PFS
      BS(3) = 1.0
      BS(6) = MACH
C
C*********************************************************************
C****** PRESSURE CALCULATION PART OF PROGRAM HAS BEEN COMPLETED *******
C*********************************************************************
C
C
C
C
C  CALCULATE SIX-COMPONENT FORCE CONTRIBUTIONS OF ELEMENT
  420 DELCA = NX     * (CP*AREA/SREF)
      DELCY = NY     * (CP*AREA/SREF)
      DELCN =-NZ     * (CP*AREA/SREF)
  440 DELCLL= DELCY * (ZCENT - ZCG)/SPAN
     1        +DELCN * (YCENT - YCG)/SPAN
      DELCLM= DELCN * (XCENT - XCG)/MAC
     1        +DELCA * (ZCENT - ZCG)/MAC
      DELCLN= DELCY * (XCENT - XCG)/SPAN
     1        -DELCA * (YCENT - YCG)/SPAN
C
C  SUM UP SIX-COMPONENT FORCE CONTRIBUTIONS
  480 CA = CA + DELCA
      CY = CY + DELCY
      CN = CN + DELCN
      CLL = CLL + DELCLL
      CLM = CLM + DELCLM
      CLN = CLN + DELCLN
      AREAT = AREAT + AREA
  490 IF (IPRINT .EQ. 0) GO TO 580
C  CHECK TO PRINT HEADER AT TOP OF PAGE
  500 IF (NPRT.LT.NPCK) GO TO 560
  510 NPRT = 0
      CALL HEADER
      WRITE (TAPEOT,520) (GTYPE(I),I=1,2)
  520 FORMAT (1H , 63X, 2A4)
  530 CONTINUE
      WRITE (TAPEOT,540) MACH,ALT,SREF,SPAN,IMPACT,IMPACI,
     1  XCG,YCG,ZCG,MAC,ISHAD,ISHADI
  540 FORMAT (1H0,20HELEMENT DATA    MACH=F7.3,7H  ALT =F8.0,9H  S REF =
     1 F8.1,8H  SPAN =F7.1,10H  IMPACT =I3,10H  IMPACI =I3,/1H ,
     2 15X5HXCG =F7.1,7H  YCG =F7.1,10H    ZCG  =F7.1,4X5HMAC =F7.1,
```

```
     3 10H    ISHAD =I3,10H  ISHADI =I3)
      WRITE (TAPEOT,550) ALPHA(J),BETA(J),CPSTAG,ETAC,DELTAS,
     1  IDERIV,Q,R,P
  550 FORMAT (1H ,5X17HANGLE OF ATTACK =F6.2,3X11HYAW ANGLE =F6.2,
     1  3X3HK =       F8.5,3X6HETAC =F8.4,3X,9HDELTA E =F6.2,/1H ,
     2  5X8HIDERIV =I3,3X3HQ =E12.5,3X,3HR =E12.5,3X3HP =E12.5,/
     3  1H0,2X,1HL,6X,6HDEL CA,8X
     4  6HDEL CY,8X6HDEL CN,7X7HDEL CLL,7X7HDEL CLM,7X7HDEL CLN,7X2HCP,
     5  13X4HAREA,/1H ,11X2HCA,12X2HCY,12X2HCN,11X3HCLL,11X3HCLM,
     6  12X3HCLN,8X5HDELTA/1H ,10X5HXCENT, 9X5HYCENT, 9X5H7CENT)
      NPRT = NPRT + 15
C
C  PRINT ELEMENT DATA
  560 WRITE (TAPEOT,570) L,DELCA,DELCY,DELCN,DELCLL,DELCLM,DELCLN,CP,
     1  AREA,CA,CY,CN,CLL,CLM,CLN,DELTA,XCENT,YCENT,ZCENT
  570 FORMAT (1H0,I4,8E14.5,/1H ,4X7E14.5/1H , 4X3E14.5)
      NPRT = NPRT + 4
C
C  SET UP AND SAVE ELEMENT RESULTS ON GEOMETRY STORAGE UNIT.
C  ONLY DATA FOR ELEMENTS ACTUALLY ON UNIT 4 ARE SAVED.
C  DATA FOR REFLECTED ELEMENTS, PARTIAL ELEMENTS, ETC., ARE NOT SAVED.
  580 IF (ISAVE .EQ. 0) GO TO 589
      IF (IG4 .LE. 0) GO TO 589
      E1( 1) = IG4
      E1( 2) = E(1)
      E1( 3) = 0.0
      E1( 4) = 0.0
      E1( 5) = CP
      E1( 6) = DELTA
      E1( 7) = BS(6)
      E1( 8) = SX
      E1( 9) = SY
      E1(10) = SZ
      E1(11) = BS(2)/PFS
      E1(12) = BS(3)/TFS
      E1(13) = DELCA
      E1(14) = DELCY
      E1(15) = DELCN
      E1(16) = DELCLL
      E1(17) = DELCLM
      E1(18) = DELCLN
      E1(19) = 0.0
      E1(20) = 0.0
      E1(21) = 0.0
      E1(22) = 0.0
```

```
      E1(23) = 0.0                                                      FORC  316
      E1(24) = 0.0                                                      FORC  317
      E1(25) = 0.0                                                      FORC  318
C   CHECK IF SPACE IS AVAILABLE FOR STORING ELEMENT DATA                FORC  319
C   INCREASE NUMBER OF RECORDS ON UNIT 4 FROM 3000 TO 20000 6/13/74 SDH FORC  320
      IF (NEXT .LE. 20000) GO TO 585                                    FORC  321
C     IF (NEXT .LE. 3000) GO TO 585                                     FORC  322
      WRITE (TAPEOT,583)                                                FORC  323
  583 FORMAT (1H0,42HSPACE ON UNIT 4 FOR SURFACE DATA HAS BEEN ,        FORC  324
     1 37HEXCEEDED. NO MORE DATA WILL BE SAVED. )                       FORC  325
      ISAVE = 0                                                         FORC  326
      GO TO 589                                                         FORC  327
  585 CONTINUE                                                          FORC  328
C     WRITE (4'NEXT) E1                                                 FORC  329I
      NSUB=(NEXT-1)/1000+1                                              FORC  330C
      NSREC4=NEXT-(NSUB-1)*1000                                         FORC  331C
      IF (NSUB.EQ.NSUB4) GO TO 3                                        FORC  332C
      CALL STINDX (4,MIDX04,21)                                         FORC  333C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                               FORC  334C
      CALL READMS (4,SIDX04,1001,NSUB)                                  FORC  335C
      CALL STINDX (4,SIDX04,1001)                                       FORC  336C
      NSUB4=NSUB                                                        FORC  337C
    3 CALL WRITMS (4,E1,25,NSREC4,0)                                    FORC  338C
C   RESET POINTER ARRAY FOR SAVED DATA                                  FORC  339
      IF (INL .EQ. 1) GO TO 587                                         FORC  340
C     E2(J) = NEXT                                                      FORC  341I
C     WRITE (4'IG41) E2                                                 FORC  342I
      SIDX16(L) = NEXT                                                  FORC  343C
      GO TO 588                                                         FORC  344
  587 CONTINUE                                                          FORC  345
C     E3(J) = NEXT                                                      FORC  346I
C     WRITE (4'IG42) E3                                                 FORC  347I
      SIDX21(L) = NEXT                                                  FORC  348C
  588 NEXT = NEXT + 1                                                   FORC  349
C     IF (IPRINT.EQ.0) GO TO 589                                        FORC  350I
C     WRITE (TAPEOT,591) E1,E2                                          FORC  351
C 591 FORMAT (1H ,3HE1=,8E12.5,/1H ,3X,8E12.5,/1H ,3X,9E12.5,/1H ,      FORC  352
C    1 3HE2=,8E12.5,/1H ,3X,8E12.5,/1H ,3X,9E12.5)                      FORC  353
C     NPRT = NPRT + 6                                                   FORC  354
  589 CONTINUE                                                          FORC  355
C ***** END OF DO LOOP FOR ALL ELEMENTS                                 FORC  356
C                                                                       FORC  357
C                                                                       FORC  358
C   SET UP ARRAY OF FINAL DATA                                          FORC  359
      F(1) = ALPHA(J)                                                   FORC  360
```

```
      F(4) = CA                                                         FORC  361
      F(5) = CY                                                         FORC  362
      F(6) = CN                                                         FORC  363
      F(7) = BETA(J)                                                    FORC  364
      F(9) = CLM                                                        FORC  365
      F(10)= CLL                                                        FORC  366
      F(11)= CLN                                                        FORC  367
C   RESOLVE NORMAL AND AXIAL FORCES IN LIFT AND DRAG DIRECTION          FORC  368
      CD = CA*COS(ALPHAR)*COS(BETAR)-CY*SIN(PHIR)*SIN(ALPHAR)*COS(BETAR) FORC  369
     1  -CY*COS(PHIR)*SIN(BETAR)+CN*COS(PHIR)*SIN(ALPHAR)*COS(BETAR)    FORC  370
     2  -CN*SIN(PHIR)*SIN(BETAR)                                        FORC  371
      CL = -CA*SIN(ALPHAR)-CY*SIN(PHIR)*COS(ALPHAR)+CN*COS(PHIR)        FORC  372
     1  *COS(ALPHAR)                                                    FORC  373
C   CALCULATE SIDE FORCE COEFFICIENT - WIND AXIS                        FORC  374
      CYPRIM = CA*COS(ALPHAR)*SIN(BETAR)-CY*SIN(PHIR)*SIN(ALPHAR)       FORC  375
     1  *SIN(BETAR)+CY*COS(PHIR)*COS(BETAR)+CN*COS(PHIR)*SIN(ALPHAR)    FORC  376
     2  *SIN(BETAR)+CN*SIN(PHIR)*COS(BETAR)                             FORC  377
C                                                                       FORC  378
C   SET UP REMAINING PARAMETERS                                         FORC  379
      F(2) = CD                                                         FORC  380
      F(3) = CL                                                         FORC  381
      IF (CD .EQ. 0.0) CD = 0.000001                                    FORC  382
      F(8) = CL / CD                                                    FORC  383
C                                                                       FORC  384
C                                                                       FORC  385
  620 RETURN                                                            FORC  386
      END                                                               FORC  387
```

```
      SUBROUTINE  FUSE                                                  FUSE  001
C                                                                       FUSE  002
      DIMENSION XFUS(4,30),NFORX(4),ZFUS(4,30),FUSARD(4,30),X(2),       FUSE  003
     1          YFUSY(4,30,30),ZFUSZ(4,30,30),Y(2),Z(2),NRADX(4)        FUSE  004
C                                                                       FUSE  005
      COMMON /ACFT/IPRINT,ITAPE,ITYPE,PI,DX,CASE                        FUSE  006
C                                                                       FUSE  007
      COMMON /FUSD/J2,J6,NFUS,NRADX,NFORX,XFUS,ZFUS,YFUSY,ZFUSZ,FUSARD  FUSE  008
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   FUSE  009
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                        FUSE  010
C                                                                       FUSE  011
      INTEGER STAT1,STAT2                                               FUSE  012
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,       FUSE  013
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                            FUSE  014
      INTEGER CASE                                                      FUSE  015
C                                                                       FUSE  016
C I IS THE FUSELAGE SEGMENT NUMBER                                      FUSE  017
C J IS THE AXIAL STATION NUMBER                                         FUSE  018
C N IS THE SURFACE POINT NUMBER                                         FUSE  019
C                                                                       FUSE  020
C THERE ARE TWO MAIN PARTS TO THIS ROUTINE                              FUSE  021
C     ONE SETS UP THE COORDS FOR AN ARBITRARY BODY                      FUSE  022
C     THE OTHER IS FOR A BODY OF REVOLUTION GIVEN THE AREA DISTRIBUTION. FUSE  023
C                                                                       FUSE  024
      K = 0                                                             FUSE  025
      I = 0                                                             FUSE  026
C TEST FOR BODY TYPE                                                    FUSE  027
      IF (J2 .NE. 1)  GO TO 90                                          FUSE  028
      IF (J6 .EQ. 1)  GO TO 90                                          FUSE  029
C                                                                       FUSE  030
C ARBITRARY BODY                                                        FUSE  031
   10 I = I + 1                                                         FUSE  032
      IF (I .GT. NFUS)  RETURN                                          FUSE  033
      ISEQ = 0                                                          FUSE  034
      STAT1 = 2                                                         FUSE  035
      STAT2 = 0                                                         FUSE  036
      J = 0                                                             FUSE  037
   20 J = J + 1                                                         FUSE  038
      IF (J .GT. NFORX(I)) GO TO 10                                     FUSE  039
      XO = XFUS(I,J)*DX                                                 FUSE  040
      N = 0                                                             FUSE  041
   30 K = 0                                                             FUSE  042
   40 N = N + 1                                                         FUSE  043
      IF (N .LE. NRADX(I)) GO TO 50                                     FUSE  044
C                                                                       FUSE  045
```

```
C SET UP DUMMY POINT.
      K = 2
      STAT1 = 0
      STAT2 = 0
      Y(K) = Y(1)
      Z(K) = Z(1)
      X(K) = X(1)
      GO TO 60
C
   50 K = K + 1
      Y(K) = YFUSY(I,J,N)
      Z(K) = ZFUSZ(I,J,N)
      X(K) = XO
      IF (K .EQ. 1) GO TO 40
C
C
C WRITE DATA
   60 ISEQ = ISEQ + 1
      WRITE(ITAPE,70) X(1),Y(1),Z(1),STAT1,
     1                X(2),Y(2),Z(2),STAT2,CASE,ITYPE,I,ISEQ
C
   70 FORMAT (3F10.4,I1,3F10.4,I1,2X12,5X11,3HFUS,I1,I4)
C
      IF (IPRINT .EQ. 1)
     1WRITE (TAPEOT,80) X(1),Y(1),Z(1),STAT1,
     2                X(2),Y(2),Z(2),STAT2,CASE,ITYPE,I,ISEQ
C
   80 FORMAT (1H ,3F10.4,I1,3F10.4,I1,2X12,5X11,3HFUS,I1,I4)
C
C
      IF (STAT1 .NE. 0) STAT1 = 0
      IF (STAT2 .NE. 0) STAT2 = 0
C CHECK FOR PROPER CYCLING
      IF (N .LT. NPADX(I))  GO TO 30
C
C HAVE COMPLETED AXIAL STATION. SET STATUS AND GO TO NEXT ONE.
      STAT1 = 1
      GO TO 20
C
C
C BODY OF REVOLUTION
   90 I = I + 1
      IF (I .GT. NFUS) RETURN
      STAT1 = 2
      STAT2 = 0
```

```
      ISEQ = 0
C
      DPHI = PI/(NRADX(I) -1)
      J = 0
  100 J = J + 1
      IF (J .GT. NFORX(I) ) GO TO 90
      XO=  XFUS(I,J)*DX
      N = 0
      PHI =  -PI/2. - DPHI
      R = SQRT(FUSARD(I,J)/PI)
  110 K = 0
  120 N = N + 1
      IF (N .LE. NRADX(I)) GO TO 130
C
C SET UP DUMMY POINT.
      K = 2
      STAT1 = 0
      STAT2 = 0
      Y(K) = Y(1)
      Z(K) = Z(1)
      X(K) = X(1)
      GO TO 140
C
  130 K = K + 1
      X(K) = XO
      PHI = PHI + DPHI
      Y(K) = R*COS(PHI)
      Z(K) = ZFUS(I,J) + R*SIN(PHI)
      IF (K .EQ. 1) GO TO 120
C
C
C WRITE DATA
  140 ISEQ = ISEQ + 1
      WRITE(ITAPE,70) X(1),Y(1),Z(1),STAT1,
     1                X(2),Y(2),Z(2),STAT2,CASE,ITYPE,I,ISEQ
C
C
      IF (IPRINT .EQ. 1)
     1WRITE (TAPEOT,80) X(1),Y(1),Z(1),STAT1,
     2                  X(2),Y(2),Z(2),STAT2,CASE,ITYPE,I,ISEQ
C
      IF (STAT1 .GT. 0) STAT1 = 0
      IF (STAT2 .GT. 0) STAT2 = 0
C
C CHECK FOR PROPER CYCLING
```

```
      IF (N .LT. NRADX(I)) GO TO 110                                    FUSE  136
C                                                                       FUSE  137
C HAVE COMPLETED AXIAL STATION. SET STATUS AND CONTINUE.                FUSE  138
      STAT1 = 1                                                         FUSE  139
      GO TO 100                                                         FUSE  140
C                                                                       FUSE  141
      END                                                               FUSE  142
```

```
      OVERLAY (MARK4,1,0)                                               GEOM  001C
      PROGRAM GEOM                                                      GEOM  002C
C     SUBROUTINE GEOM                                                   GEOM  003I
C                                                                       GEOM  004
C **** THIS ROUTINE CONTROLS THE EXECUTION OF THE GEOMETRY GENERATION   GEOM  005
C **** ROUTINES                                                         GEOM  006
C                                                                       GEOM  007
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR                                GEOM  008
      COMMON /GFLAG/IOUT,ISTAT3,IORIN,COMPIN                            GEOM  009
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   GEOM  010
     1            TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                         GEOM  011
      COMMON /TAGS/ITAG4,ITAG9,ITAG10                                   GEOM  012
C                                                                       GEOM  013
      COMMON/SUBIDX/SIDX04(1001),SIDX16(1001),SIDX21(1001)              GEOM  014C
      COMMON /MASTER/ MIDX04(21),INDEX9(501),INDX10(2001),INDX15(2001), GEOM  015C
     1  MIDX16(94),INDX20(2421),MIDX21(73)                              GEOM  016C
      DIMENSION  XA(250),XB(250),YA(250),YB(250),ZA(250),ZB(250),       GEOM  017
     1 XI(4),ETA(4),XIN(4),YIN(4),ZIN(4),TITLE(15),XPA(4),YPA(4),ZPA(4) GEOM  018
      DIMENSION CONFIG(15),IGEOM(10),E(25),EU(25),EP(25)                GEOM  019
      DIMENSION ELEM(25),PANEL(100),IORN(100),SYMFCT(100),IFA(100)      GEOM  020
C                                                                       GEOM  021
      REAL  NX,NY,NZ,LEFCT                                              GEOM  022
      LOGICAL  BFLAG, AFLAG, BFLAG                                      GEOM  023
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,       GEOM  024
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                            GEOM  025
      INTEGER  STAT, STATT, PAGE, CASE, TYPE, ERROR, PRINTS, SYMFCT, SEQ GEOM  026
      INTEGER SIDX04,SIDX16,SIDX21                                      GEOM  027C
C                                                                       GEOM  028
C                                                                       GEOM  029
      ISTAT3 = 0                                                        GEOM  030
C                                                                       GEOM  031
C  READ GEOMETRY CONTROL CARD                                           GEOM  032
      READ (TAPEIN,10) IOUT,IREW,PRINTS,IQUAD,I3MAX,NEW,NPMAX,CONFIG    GEOM  033
   10 FORMAT (1I2,5I1,1I2,15A4)                                         GEOM  034
C  INITIALIZE ELEMENT DATA STORAGE UNIT IF REQUIRED                     GEOM  035
      DO 13 I=1,25                                                      GEOM  036
      E (I) = 0.0                                                       GEOM  037
      EP(I) = 0.0                                                       GEOM  038
   13 EU(I) = 0.0                                                       GEOM  039
      IF (NEW .EQ. 1) GO TO 15                                          GEOM  040
      IF (NEW .EQ. 2)  GO TO 14                                         GEOM  041
      IG4 = 1                                                           GEOM  042
C     WRITE (4'IG4) CONFIG                                              GEOM  043I
      NSUB4=1                                                           GEOM  044C
      NSREC4=1                                                          GEOM  045C
```

```
      CALL STINDX (4,MIDX04,21)                                         GEOM  046C
      CALL READMS (4,SIDX04,1001,NSUB4)                                 GEOM  047C
      CALL STINDX (4,SIDX04,1001)                                       GEOM  048C
      CALL WRITMS (4,CONFIG,15,NSREC4,0)                                GEOM  049C
      IF (NPMAX .EQ. 0) NPMAX = 50                                      GEOM  050
C  NREM IS THE NUMBER OF RECORDS PER ELEMENT                            GEOM  051
      NREM = 3                                                          GEOM  052
      NEXT = (NPMAX+1)*5                                                GEOM  053
      NP = 0                                                            GEOM  054
      E(1) = NEXT                                                       GEOM  055
      E(2) = NP                                                         GEOM  056
      E(3) = NPMAX                                                      GEOM  057
      E(4) = NREM                                                       GEOM  058
      GO TO 19                                                          GEOM  059
   14 NREPL = NPMAX                                                     GEOM  060
      ISTAT3 = NREPL - 1                                                GEOM  061
   15 IG4 = 1                                                           GEOM  062
C     READ (4'IG4) CONFIG                                               GEOM  063I
      NSUB4=1                                                           GEOM  064C
      NSREC4=1                                                          GEOM  065C
      CALL STINDX (4,MIDX04,21)                                         GEOM  066C
      CALL READMS (4,SIDX04,1001,NSUB4)                                 GEOM  067C
      CALL STINDX (4,SIDX04,1001)                                       GEOM  068C
      CALL READMS (4,CONFIG,15,NSREC4)                                  GEOM  069C
      IG4 = 2                                                           GEOM  070
C     READ (4'IG4) E                                                    GEOM  071I
      NSREC4=2                                                          GEOM  072C
      CALL READMS (4,E,25,NSREC4)                                       GEOM  073C
      NEXT = E(1)                                                       GEOM  074
      NP   = E(2)                                                       GEOM  075
      NPMAX= E(3)                                                       GEOM  076
      NREM = E(4)                                                       GEOM  077
   19 CONTINUE                                                          GEOM  078
C                                                                       GEOM  079
C  CHECK IF QUADRILATERALS ONLY ARE TO BE CALCULATED                    GEOM  080
      IF (IQUAD .EQ. 1) GO TO 110                                       GEOM  081
      IF (IREW.EQ.0 .AND. IOUT.NE.5 .AND. IOUT.NE.1) REWIND IOUT        GEOM  082
C  READ COMPONENT IDENTIFICATION CARD                                   GEOM  083
   20 ISTAT3 = ISTAT3 + 1                                               GEOM  034
      READ (TAPEIN,30) PANEL(ISTAT3),LAST,IORN(ISTAT3),IGEOM,           GEOM  085
     1     SYMFCT(ISTAT3),IFA(ISTAT3),NADJ1,NADJ2,NADJ3,NADJ4           GEOM  036
   30  FORMAT (A4,2I1,10I1,2I1,4I2)                                     GEOM  087
      IORIEN = IORN(ISTAT3)                                             GEOM  088
      COMPIN = PANEL(ISTAT3)                                            GEOM  089
      IF (NEW .NE. 2)  GO TO 36                                         GEOM  090
```

```
      IG4 = NREPL*5                                                     GEOM  091
C     READ(4'IG4)  EP                                                   GEOM  092I
      NSUB=(IG4-1)/1000+1                                               GEOM  093C
      NSREC4=IG4-(NSUB-1)*1000                                          GEOM  094C
      IF (NSUB.EQ.NSUB4) GO TO 33                                       GEOM  095C
      CALL STINDX (4,MIDX04,21)                                         GEOM  096C
      CALL READMS (4,SIDX04,1001,NSUB)                                  GEOM  097C
      CALL STINDX (4,SIDX04,1001)                                       GEOM  098C
      NSUB4=NSUB                                                        GEOM  099C
   33 CALL READMS (4,EP,25,NSREC4)                                      GEOM  100C
      IF (COMPIN .EQ. EP(2))  GO TO 35                                  GEOM  101
      WRITE(TAPEOT,34)  NREPL, COMPIN, EP(2)                            GEOM  102
   34 FORMAT  (  47H0 IN REPLACING AN EXISTING COMPONENT (NEW = 2) ,    GEOM  103
     1            7H NUMBER, I3,18H INPUT PANEL ID ( ,A4, 2H )//       GEOM  104
     2           47H0 DID NOT AGREE WITH THE ID STORED ON UNIT 4 ( ,   GEOM  105
     3          A4,2H ))                                               GEOM  106
      STOP                                                              GEOM  107
C                                                                       GEOM  108
   35 NEXT = EP(3)                                                      GEOM  109
   36 CONTINUE                                                          GEOM  110
C                                                                       GEOM  111
C                                                                       GEOM  112
C ****************************                                          GEOM  113
C  READ OR GENERATE ELEMENTS FOR EACH VEHICLE COMPONENT                 GEOM  114
      DO 90 I=1,10                                                      GEOM  115
       IF (IGEOM(I) .EQ. 0) GO TO 100                                   GEOM  116
      IG = IGEOM(I)                                                     GEOM  117
      GO TO (40,50,60,70), IG                                           GEOM  118
C  40 CALL TELE                                                         GEOM  119I
   40 CALL OVERLAY (5HMARK4,1,1,6HRECALL)                               GEOM  120C
      GO TO 80                                                          GEOM  121
C                                                                       GEOM  122
C  50 CALL ELLIP                                                        GEOM  123I
   50 CALL OVERLAY (5HMARK4,1,2,6HRECALL)                               GEOM  124C
      GO TO 80                                                          GEOM  125
C  60 CALL CUBIC                                                        GEOM  126I
   60 CALL OVERLAY (5HMARK4,1,3,6HRECALL)                               GEOM  127C
      GO TO 80                                                          GEOM  128
C  70 CALL AIRCFT                                                       GEOM  129I
   70 CALL OVERLAY (5HMARK4,1,4,6HRECALL)                               GEOM  130C
C                                                                       GEOM  131
   80 IF (ERROR .NE. 0) GO TO 420                                       GEOM  132
   90 CONTINUE                                                          GEOM  133
C                                                                       GEOM  134
  100 IF (LAST..EQ. 0) GO TO 20                                         GEOM  135
```

```
C  THE GENERATION OF ALL ELEMENT DATA IS COMPLETED
      I3MAX = ISTAT3
C
  110 IF (IQUAD .EQ. 2) GO TO 420
      IF (IOUT .NE.5 .AND. IOUT .NE.1)  REWIND IOUT
      IG4 = I3MAX + 1
C
C
C**************************
C  ELEMENT DATA WILL NOW BE CONVERTED TO QUADRILATERALS
C     SET UP STARTING CONSTANTS
      ISTAT3 = 0
      IF (NEW .EQ. 2)  ISTAT3 = NREPL - 1
  120 CONTINUE
      ISTAT3 = ISTAT3 + 1
      IF (ISTAT3 .LE. NPMAX) GO TO 125
      WRITE (TAPEOT,122)
  122  FORMAT (1H0,46H**NUMBER OF PANELS EXCEEDS NPMAX**PROGRAM STOP   )
      STOP
  125 IFACT  = IFA(ISTAT3)
      IORIEN = IORN(ISTAT3)
C
       N = -1
      NN = -1
       KLCT = 0
       NPRT = 10
       L = 0
       IGT = 0
       AREAT = 0.0
       VOL = 0.0
C  READ SCALE FACTOR CARD IF REQUIRED
      IF (IFACT .EQ. 1) READ (TAPEIN,130) XSC,YSC,ZSC,DELX,DELY,DELZ
  130  FORMAT (6F10.0)
      IA = 0
      IB = 0
      IF (IORIEN .EQ. 2) IB = 1
      IF (IORIEN .EQ. 3) IA = 1
      IF (IORIEN .GT. 1) IORIEN = 1
      IF (IGTYPE .GT. 0) IGT = 1
C
C     READ IN ALL SURFACE DATA
      READ (IOUT,140) X,Y,Z,STAT,XX,YY,ZZ,STATT,CASE,SECT,TYPE,SEQ
  140 FORMAT(3F10.4,I1,3F10.4,I1,2X,I2,1A4,I2,4X,I4)
      STAT =IABS(STAT)
      STATT =IABS(STATT)
```

```
      IF (TYPE.NE.3) GO TO 430
      IF(STAT.EQ.2.OR.STATT.EQ.2)
     1        SECTS = SECT
       BFLAG = .FALSE.
       GO TO 180
  150 IF (BFLAG) GO TO 160
       BFLAG = .TRUE.
       X = XX
       Y = YY
       Z = ZZ
       STAT = STATT
       GO TO 170
  160 BFLAG = .FALSE.
      READ (IOUT,140) X,Y,Z,STAT,XX,YY,ZZ,STATT,CASE,SECT,TYPE,SEQ
      STAT =IABS(STAT)
      STATT =IABS(STATT)
      IF (TYPE.NE.3) GO TO 430
      IF(STAT.EQ.2.OR.STATT.EQ.2)
     1        SECTS = SECT
  170 IF (STAT .EQ. 0 .OR. STAT .EQ. 3) GO TO 280
      IF (STAT .EQ. 2) GO TO 270
      IF (.NOT. AFLAG) GO TO 270
       MC = M
  180 M = 1
       IF (STAT .EQ. 2) GO TO 240
       IF (.NOT. BFLAG) GO TO 220
  190 DO 200 J =1,MC
        XA(J) = XB(J)
        YA(J) = YB(J)
  200   ZA(J) = ZB(J)
  210 XB(1) = X
      YB(1) = Y
      ZB(1) = Z
      GO TO 150
  220 IF (AFLAG) GO TO 230
       BFLAG = .TRUE.
       GO TO 190
  230 AFLAG = .FALSE.
       GO TO 210
  240 AFLAG = .TRUE.
       BFLAG = .FALSE.
       N = N+1
      NN = NN + 1
  250 XA(M) = X
      YA(M) = Y
```

```
      ZA(M) = Z                                                         GEOM  226
      GO TO 150                                                         GEOM  227
  260 M = M + 1                                                         GEOM  228
      IF (AFLAG) GO TO 250                                              GEOM  229
       XB(M) = X                                                        GEOM  230
       YB(M) = Y                                                        GEOM  231
       ZB(M) = Z                                                        GEOM  232
      IF (STAT .NE. 3) GO TO 150                                        GEOM  233
  270 MMIN = MINO (M,MC) - 1                                            GEOM  234
       NN2 = 1                                                          GEOM  235
       MC = M                                                           GEOM  236
      N = N + 1                                                         GEOM  237
      NN = NN + 1                                                       GEOM  238
       KLCT = KLCT + 1                                                  GEOM  239
       JJ = 0                                                           GEOM  240
C                                                                       GEOM  241
C  BEGIN COMPUTATION OF SURFACE ELEMENT CHARACTERISTICS                 GEOM  242
      DO 390 I= 1,MMIN                                                  GEOM  243
      IIA = I + IA                                                      GEOM  244
      IIB = I + IB                                                      GEOM  245
      IF (IFACT.EQ.1) GO TO 280                                         GEOM  246
        XIN(1) = XA(IIA )                                               GEOM  247
        XIN(2) = XA(IIA +1)                                             GEOM  248
        XIN(3) = XB(IIB +1)                                             GEOM  249
        XIN(4) = XB(IIB )                                               GEOM  250
        YIN(1) = YA(IIA )                                               GEOM  251
        YIN(2) = YA(IIA +1)                                             GEOM  252
        YIN(3) = YB(IIB +1)                                             GEOM  253
        YIN(4) = YB(IIB )                                               GEOM  254
        ZIN(1) = ZA(IIA )                                               GEOM  255
        ZIN(2) = ZA(IIA +1)                                             GEOM  256
        ZIN(3) = ZB(IIB +1)                                             GEOM  257
        ZIN(4) = ZB(IIB )                                               GEOM  258
      GO TO 290                                                         GEOM  259
C                                                                       GEOM  260
  280   XIN(1) = XA(IIA )    * XSC + DELX                               GEOM  261
        XIN(2) = XA(IIA +1) * XSC + DELX                                GEOM  262
        XIN(3) = XB(IIB +1) * XSC + DELX                                GEOM  263
        XIN(4) = XB(IIB )    * XSC + DELX                               GEOM  264
        YIN(1) = YA(IIA )    * YSC + DELY                               GEOM  265
        YIN(2) = YA(IIA +1) * YSC + DELY                                GEOM  266
        YIN(3) = YB(IIB +1) * YSC + DELY                                GEOM  267
        YIN(4) = YB(IIB )    * YSC + DELY                               GEOM  268
        ZIN(1) = ZA(IIA )    * ZSC + DELZ                               GEOM  269
        ZIN(2) = ZA(IIA +1) * ZSC + DELZ                                GEOM  270
```

```
      ZIN(3) = ZB(IIB +1) * ZSC + DELZ                                  GEOM  271
      ZIN(4) = ZB(IIB )   * ZSC + DELZ                                  GEOM  272
C  FORM DIAGONAL VECTORS                                                GEOM  273
  290 T1X = XIN(3) - XIN(1)                                             GEOM  274
      T2X = XIN(4) - XIN(2)                                             GEOM  275
      T1Y = YIN(3) - YIN(1)                                             GEOM  276
      T2Y = YIN(4) - YIN(2)                                             GEOM  277
      T1Z = ZIN(3) - ZIN(1)                                             GEOM  278
      T2Z = ZIN(4) - ZIN(2)                                             GEOM  279
C                                                                       GEOM  280
C  FORM CROSS PRODUCT  N=T2 X T1                                        GEOM  281
      NX = T2Y*T1Z - T1Y*T2Z                                            GEOM  282
      NY = T1X*T2Z - T2X*T1Z                                            GEOM  283
      NZ = T2X*T1Y - T1X*T2Y                                            GEOM  284
      VN = SQRT ( NX*NX + NY*NY + NZ*NZ )                               GEOM  285
     IF (VN .EQ. 0.0) GO TO 300                                         GEOM  286
C                                                                       GEOM  287
C  FORM UNIT NORMAL VECTOR                                              GEOM  288
      NX = NX / VN                                                      GEOM  289
      NY = NY / VN                                                      GEOM  290
      NZ = NZ / VN                                                      GEOM  291
C                                                                       GEOM  292
C  COMPUTE AVERAGE POINT                                                GEOM  293
  300 AVX = 0.25 * (XIN(1) + XIN(2) + XIN(3) +  XIN(4) )                GEOM  294
      AVY = 0.25 * (YIN(1) + YIN(2) + YIN(3) + YIN(4) )                 GEOM  295
      AVZ = 0.25 * (ZIN(1) + ZIN(2) + ZIN(3) + ZIN(4) )                 GEOM  296
C                                                                       GEOM  297
C  COMPUTE PROJECTION DISTANCE                                          GEOM  298
      D = NX*(AVX - XIN(1)) + NY*(AVY - YIN(1)) + NZ*(AVZ-ZIN(1))       GEOM  299
      PD = ABS(D)                                                       GEOM  300
C                                                                       GEOM  301
      T = SQRT (T1X*T1X + T1Y*T1Y + T1Z*T1Z)                            GEOM  302
     IF (T .EQ. 0.0) GO TO 310                                          GEOM  303
     T1X = T1X / T                                                      GEOM  304
      T1Y = T1Y / T                                                     GEOM  305
      T1Z = T1Z / T                                                     GEOM  306
C                                                                       GEOM  307
  310 T2X = NY*T1Z - NZ*T1Y                                             GEOM  308
      T2Y = NZ*T1X - NX*T1Z                                             GEOM  309
      T2Z = NX*T1Y - NY*T1X                                             GEOM  310
C                                                                       GEOM  311
C  COMPUTE COORDINATES OF CORNER POINTS IN REFERENCE COORD. SYSTEM      GEOM  312
     DO 320 J = 1,4                                                     GEOM  313
      XPA(J) = XIN(J) + NX*D                                            GEOM  314
      YPA(J) = YIN(J) + NY*D                                            GEOM  315
```

```
      ZPA(J) = ZIN(J) + NZ*D                                            GEOM  316
      D = - D                                                           GEOM  317
      XDIF = XPA(J) - AVX                                               GEOM  318
      YDIF = YPA(J) - AVY                                               GEOM  319
      ZDIF = ZPA(J) - AVZ                                               GEOM  320
C                                                                       GEOM  321
C  TRANSFORM CORNER POINTS TO ELEMENT COORDINATE SYSTEM (XI,ETA) WITH   GEOM  322
C   AVERAGE POINT AS ORIGIN                                             GEOM  323
      XI(J) = T1X*XDIF + T1Y*YDIF + T1Z*ZDIF                            GEOM  324
  320 ETA(J)= T2X*XDIF + T2Y*YDIF + T2Z*ZDIF                            GEOM  325
     ETACK = ETA(2) - ETA(4)                                            GEOM  326
     IF (ETACK .NE.0.0) GO TO 330                                       GEOM  327
      XIO = 0.0                                                         GEOM  328
      GO TO 340                                                         GEOM  329
C                                                                       GEOM  330
C  COMPUTE CENTROID                                                     GEOM  331
  330 XIO = .333333333 * (XI(4) * (ETA(1)-ETA(2)) + XI(2)               GEOM  332
     1        * (ETA(4)-ETA(1))) / (ETA(2)-ETA(4))                     GEOM  333
  340 ETAO = -.333333333 * ETA(1)                                       GEOM  334
C                                                                       GEOM  335
C  OBTAIN CORNER POINTS IN SYSTEM WITH CENTROID AS ORIGIN               GEOM  336
     DO 350 J = 1,4                                                     GEOM  337
      XI(J) = XI(J) - XIO                                               GEOM  338
  350 ETA(J) = ETA(J) - ETAO                                            GEOM  339
C                                                                       GEOM  340
C  TRANSFORM CENTROID TO REFERENCE COORDINATE SYSTEM                    GEOM  341
      XCENT = AVX + T1X*XIO + T2X*ETAO                                  GEOM  342
      YCENT = AVY + T1Y*XIO + T2Y*ETAO                                  GEOM  343
      ZCENT = AVZ + T1Z*XIO + T2Z*ETAO                                  GEOM  344
C                                                                       GEOM  345
C  CONSTANTS FOR USE IN COMPUTING AREA OF ELEMENT                       GEOM  346
      XI3M1 = XI(3) - XI(1)                                             GEOM  347
      ETA2M4 = ETA(2) - ETA(4)                                          GEOM  348
C                                                                       GEOM  349
C  COMPUTE AREA AND VOLUME OF ELEMENTS                                  GEOM  350
      AREA = 0.5 * XI3M1 * ETA2M4                                       GEOM  351
     AREAT = AREAT + AREA                                               GEOM  352
      DELVOL = AREA * NY * YCENT                                        GEOM  353
     VOL = VOL + DELVOL                                                 GEOM  354
     L = L + 1                                                          GEOM  355
     II = I                                                             GEOM  356
     IF (PRINTS.EQ.0) GO TO 380                                         GEOM  357
C                                                                       GEOM  358
C                                                                       GEOM  359
C  PRINT RESULTS OF CALCULATIONS TO DETERMINE ELEMENT CHARACTERISTICS   GEOM  360
```

```
      IF (NPRT .GE.9)  GO TO 360                                        GEOM  361
       NPRT = NPRT + 1                                                  GEOM  362
      IF  (I .EQ. 1) GO TO 370                                          GEOM  363
      WRITE (TAPEOT,450) I,XIN,NX,XCENT,AREA,L,YIN,NY,YCENT,DELVOL,     GEOM  364
     1       ZIN,NZ,ZCENT,VOL                                           GEOM  365
      GO TO 380                                                         GEOM  366
  360 NPRT = 0                                                          GEOM  367
      CALL HEADER                                                       GEOM  368
      WRITE (TAPEOT,480)                                                GEOM  369
  370 WRITE (TAPEOT,460) N,I,XIN,NX,XCENT,AREA,L,YIN,NY,YCENT,DELVOL,   GEOM  370
     1ZIN,NZ,ZCENT,VOL                                                  GEOM  371
C                                                                       GEOM  372
C  SET UP DATA TO BE SAVED AND USED IN FORCE CALCULATIONS               GEOM  373
  380 ELEM(1)  = L                                                      GEOM  374
      ELEM(2)  = N                                                      GEOM  375
      ELEM(3)  = I                                                      GEOM  376
      ELEM(4)  = NX                                                     GEOM  377
      ELEM(5)  = NY                                                     GEOM  378
      ELEM(6)  = NZ                                                     GEOM  379
      ELEM(7)  = XCENT                                                  GEOM  380
      ELEM(8)  = YCENT                                                  GEOM  381
      ELEM(9)  = ZCENT                                                  GEOM  382
      ELEM(10)  = AREA                                                  GEOM  383
      ELEM(11)  = XIN(1)                                                GEOM  384
      ELEM(12)  = XIN(2)                                                GEOM  385
      ELEM(13)  = XIN(3)                                                GEOM  386
      ELEM(14)  = XIN(4)                                                GEOM  387
      ELEM(15)  = YIN(1)                                                GEOM  388
      ELEM(16)  = YIN(2)                                                GEOM  389
      ELEM(17)  = YIN(3)                                                GEOM  390
      ELEM(18)  = YIN(4)                                                GEOM  391
      ELEM(19)  = ZIN(1)                                                GEOM  392
      ELEM(20)  = ZIN(2)                                                GEOM  393
      ELEM(21)  = ZIN(3)                                                GEOM  394
      ELEM(22)  = ZIN(4)                                                GEOM  395
      ELEM(23)  = 0.0                                                   GEOM  396
      ELEM(24)  = 0.0                                                   GEOM  397
      ELEM(25)  = 0.0                                                   GEOM  398
C  SAVE ELEMENT DATA ON GEOMETRY SAVE UNIT                              GEOM  399
      IG4 = NEXT                                                        GEOM  400
C     WRITE (4'IG4) ELEM                                                GEOM  401I
      NSUB=(IG4-1)/1000+1                                               GEOM  402C
      NSREC4=IG4-(NSUB-1)*1000                                          GEOM  403C
      IF (NSUB.EQ.NSUB4) GO TO 383                                      GEOM  404C
      CALL STINDX (4,MIDX04,21)                                         GEOM  405C
```

```
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                                GEOM  406C
      CALL READMS (4,SIDX04,1001,NSUB)                                   GEOM  407C
      CALL STINDX (4,SIDX04,1001)                                        GEOM  408C
      NSUB4=NSUB                                                         GEOM  409C
  383 CALL WRITMS (4,ELEM,25,NSREC4,0)                                   GEOM  410C
      IF (L .EQ. 1) ISTART = IG4                                         GEOM  411
      IF (NREM .EQ. 1) GO TO 385                                         GEOM  412
      NREMM = NREM - 1                                                   GEOM  413
      DO 384 J=1,NREMM                                                   GEOM  414
      IG4 = IG4 + 1                                                      GEOM  415
C 384 WRITE (4'IG4) EQ                                                   GEOM  416I
      NSUB=(IG4-1)/1000+1                                                GEOM  417C
      NSREC4=IG4-(NSUB-1)*1000                                           GEOM  418C
      IF (NSUB.EQ.NSUB4) GO TO 384                                       GEOM  419C
      CALL STINDX (4,MIDX04,21)                                          GEOM  420C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                                GEOM  421C
      CALL READMS (4,SIDX04,1001,NSUB)                                   GEOM  422C
      CALL STINDX (4,SIDX04,1001)                                        GEOM  423C
      NSUB4=NSUB                                                         GEOM  424C
  384 CALL WRITMS (4,EQ,25,NSREC4,0)                                     GEOM  425C
  385 NEXT = NEXT + NREM                                                 GEOM  426
  390 CONTINUE                                                           GEOM  427
C                                                                        GEOM  428
C  CHECK TO IDENTIFY END OF INPUT ELEMENTS                               GEOM  429
  400 IF (STAT .LT.2) GO TO 410                                          GEOM  430
       NPRT = NPRT + 1                                                   GEOM  431
      WRITE (TAPEOT,470) SECTS,AREAT,L,VOL                               GEOM  432
      NN = NN - 1                                                        GEOM  433
      N  = - 1                                                           GEOM  434
C  TEST FOR END OF CASE                                                  GEOM  435
  410 IF (STAT .NE. 3) GO TO 180                                         GEOM  436
C                                                                        GEOM  437
C  RECORD TABLE OF CONTENTS DATA ON GEOMETRY SAVE UNIT                   GEOM  438
      EP(1) = ISTAT3                                                     GEOM  439
      EP(2) = PANEL(ISTAT3)                                              GEOM  440
      EP(3) = ISTART                                                     GEOM  441
      EP(4) = L                                                          GEOM  442
      EP(5) = IORN(ISTAT3)                                               GEOM  443
      EP(6) = SYMFCT(ISTAT3)                                             GEOM  444
      EP(7) = N                                                          GEOM  445
      EP(8) = I                                                          GEOM  446
      EP(9) = NADJ1                                                      GEOM  447
      EP(10)= NADJ2                                                      GEOM  448
      EP(11)= NADJ3                                                      GEOM  449
      EP(12)= NADJ4                                                      GEOM  450
```

```
      NP = NP + 1                                                       GEOM  451
      IG4 = NP*5                                                        GEOM  452
      IF (NEW .EQ. 2)  IG4 = NPEPL*5                                    GEOM  453
C     WRITE (4'IG4) EP                                                  GEOM  454I
      NSUB=(IG4-1)/1000+1                                               GEOM  455C
      NSREC4=IG4-(NSUB-1)*1000                                          GEOM  456C
      IF (NSUB.EQ.NSUB4) GO TO 411                                      GEOM  457C
      CALL STINDX (4,MIDX04,21)                                         GEOM  458C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                               GEOM  459C
      CALL READMS (4,SIDX04,1001,NSUB)                                  GEOM  460C
      CALL STINDX (4,SIDX04,1001)                                       GEOM  461C
      NSUB4=NSUB                                                        GEOM  462C
  411 CALL WRITMS (4,EP,25,NSREC4,0)                                    GEOM  463C
      IF (NEW .EQ. 2)  GO TO 490                                        GEOM  464
      IG1=IG4+1                                                         GEOM  465
      IG2=IG4+4                                                         GEOM  466
C     DO 419 IG4=IG1,IG2                                                GEOM  467I
C 419 WRITE (4'IG4) EO                                                  GEOM  468I
      NSUB=(IG4-1)/1000+1                                               GEOM  469C
      IF (NSUB.EQ.NSUB4) GO TO 414                                      GEOM  470C
      NSRECR=4                                                          GEOM  471C
  412 CONTINUE                                                          GEOM  472C
      NSREC2=NSRECR                                                     GEOM  473C
      CALL STINDX (4,MIDX04,21)                                         GEOM  474C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                               GEOM  475C
      CALL READMS (4,SIDX04,1001,NSUB)                                  GEOM  476C
      CALL STINDX (4,SIDX04,1001)                                       GEOM  477C
      NSUB4=NSUB                                                        GEOM  478C
      DO 413 NSREC4=1,NSREC2                                            GEOM  479C
  413 CALL WRITMS (4,EO,25,NSREC4,0)                                    GEOM  480C
      GO TO 417                                                         GEOM  481C
  414 CONTINUE                                                          GEOM  482C
      NSREC1=IG1-(NSUB-1)*1000                                          GEOM  483C
      NSREC2=NSREC1+3                                                   GEOM  484C
      NSRECR=0                                                          GEOM  485C
      IF (NSREC2.LE.1000) GO TO 415                                     GEOM  486C
      NSRECR=NSREC2-1000                                                GEOM  487C
      NSUB=NSUB+1                                                       GEOM  488C
      NSREC2=1000                                                       GEOM  489C
  415 DO 416 NSREC4=NSREC1,NSREC2                                       GEOM  490C
  416 CALL WRITMS (4,EO,25,NSREC4,0)                                    GEOM  491C
      IF (NSRECR.NE.0) GO TO 412                                        GEOM  492C
  417 CONTINUE                                                          GEOM  493C
C     IF (ISTAT3 .GT. NP) NP = ISTAT3                                   GEOM  494
      IF (ISTAT3 .LT. I3MAX) GO TO 120                                  GEOM  495
```

```
      E(1) = NEXT                                                       GEOM  496
      E(2) = NP                                                         GEOM  497
      IG4 = 2                                                           GEOM  498
C     WRITE (4'IG4) E                                                   GEOM  499I
      EO(1) = NEXT                                                      GEOM  500
      IG4 = 3                                                           GEOM  501
C     WRITE (4'IG4) EO                                                  GEOM  502I
      NSUB=1                                                            GEOM  503C
      IF (NSUB.EQ.NSUB4) GO TO 418                                      GEOM  504C
      CALL STINDX (4,MIDX04,21)                                         GEOM  505C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                               GEOM  506C
      CALL READMS (4,SIDX04,1001,NSUB)                                  GEOM  507C
      CALL STINDX (4,SIDX04,1001)                                       GEOM  508C
      NSUB4=NSUB                                                        GEOM  509C
  418 NSREC4=2                                                          GEOM  510C
      CALL WRITMS (4,E,25,NSREC4,0)                                     GEOM  511C
      NSREC4=3                                                          GEOM  512C
      CALL WRITMS (4,EO,25,NSREC4,0)                                    GEOM  513C
      CALL STINDX (4,MIDX04,21)                                         GEOM  514C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                               GEOM  515C
  420 GO TO 490                                                         GEOM  516
C                                                                       GEOM  517
C  ERROR CHECK ON READING CARDS                                         GEOM  518
  430 WRITE (TAPEOT,440)                                                GEOM  519
C                                                                       GEOM  520
  440 FORMAT (1H0,50H***** SURFACE DATA ROUTINE HAS ATTEMPTED TO READ A GEOM  521
     142H NON SURFACE CARD - CHECK YOUR CARDS *****  )                  GEOM  522
      ERROR = 1                                                         GEOM  523
      GO TO 490                                                         GEOM  524
C                                                                       GEOM  525
  450 FORMAT (1H0,7X, I4, 1P4E14.5,0PF10.6,1P2E14.5,I6,2(/12X,4E14.5,   GEOM  526
     1 0PF10.6,1P2E14.5)  )                                             GEOM  527
C                                                                       GEOM  528
  460 FORMAT (1H0,3X, 2I4,1P4E14.5,0PF10.6,1P2E14.5,I6,2(/12X,4E14.5,   GEOM  529
     1 0PF10.6,1P2E14.5)  )                                             GEOM  530
C                                                                       GEOM  531
  470 FORMAT (1H0, 8HSECTION=1A4,  33H   TOTAL AREA OF INPUT ELEMENTS = GEOM  532
     1 F12.3,6X26HTOTAL NUMBER OF ELEMENTS = I5/1H ,12X,                GEOM  533
     2 33H TOTAL VOLUME OF INPUT ELEMENTS =F12.3,/1H0,3(20X,            GEOM  534
     3 9H*********))                                                    GEOM  535
C                                                                       GEOM  536
  480 FORMAT (1H0,28H  INPUT SURFACE ELEMENT DATA/1H0,6X1HN3X1HM7X1HX,  GEOM  537
     1 3(13X,1HX),11X2HNX9X5HXCENT9X4HAREA8X1HL  ,/1H ,5X, 4(13X,1HY),  GEOM  538
     2 11X2HNY9X5HYCENT ,7X,7HDELTA V,/1H ,5X,4(13X,1HZ),11X2HNZ,       GEOM  539
     3 9X,5HZCENT ,7X,6HVOLUME,/1H  )                                   GEOM  540
```

```
C                                                                       GEOM   541
C  490 RETURN                                                           GEOM   542I
   490 CONTINUE                                                         GEOM   543C
       END                                                              GEOM   544
```

```
      SUBROUTINE GETT(IU, IT, N1, A1, N2, A2)
C
      DIMENSION  A1(N1), A2(N2)
C
      GO TO (10,20,30,40), IT
C
C  READ A1
   10 READ(IU) A1
      RETURN
C
C  READ N1 AND A1
   20 READ(IU) N1, A1
      RETURN
C
C  READ A1 AND A2
   30 READ(IU) A1, A2
      RETURN
C
C  READ IDUM AND A1
   40 READ(IU) IDUM, A1
      RETURN
      END
```

```
      SUBROUTINE HEADER                                                 HEAD  001
C                                                                       HEAD  002
C                                                                       HEAD  003
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR                                HEAD  004
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   HEAD  005
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                        HEAD  006
C                                                                       HEAD  007
      INTEGER  PAGE, ERROR, CASE                                        HEAD  008
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,       HEAD  009
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                            HEAD  010
      DIMENSION  TITLE(15)                                              HEAD  011
C                                                                       HEAD  012
C  PRINT OUT HEADER AT TOP OF EACH PAGE OF OUTPUT                       HEAD  013
      WRITE (TAPEOT,10) CASE,PAGE,TITLE                                 HEAD  014
   10 FORMAT (1H1,46HSUPERSONIC/HYPERSONIC ARBITRARY-BODY PROGRAM, ,    HEAD  015
     1 14HMARK IVS MOD 0,/,1H0,6HCASE   ,2X,I2,85X,5HPAGE ,I4,/1H0,14A4, HEAD  016
     2  1A3)                                                            HEAD  017
C                                                                       HEAD  018
C  STEP PAGE NUMBER BY ONE                                              HEAD  019
      PAGE = PAGE + 1                                                   HEAD  020
C                                                                       HEAD  021
      RETURN                                                            HEAD  022
      END                                                               HEAD  023
```

```
      SUBROUTINE HNGFT(IREAD)
C
C     THIS ROUTINE GENERATES ELEVCN HINGE MOMENT FACTORS NEEDED BY THE
C      SHEAR ROUTINE.
C
C
      COMMON /SUBIDX/SIDX04(1001),SIDX16(1001),SIDX21(1001)
      COMMON /MASTER/MIDX04(21),INDEX9(501),INDX10(2001),INDX15(2001),
     1  MIDX16(94),INDX20(2421),MIDX21(73)
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
C
      DIMENSION  NOFP(30), E(25), EP(25), ELEM(25)
      DIMENSION LAS(4),XHL(4,2),YHL(4,2),ZHL(4,2),NIPL(4),NOF(8,4)
C
      INTEGER SIDX04,SIDX16,SIDX21
      INTEGER TAPEIN
C
      DATA LAS /0,0,0,1/
      DATA XHL,YHL,ZHL/
     1   -104.166,  -99.000, -102.136,  -98.287,
     2   -102.136,  -98.287, -104.166,  -99.000,
     3      2.610,   17.167,  -11.167,  -27.808,
     4     11.167,   27.080,   -2.610,  -17.167,
     5      8.375,    8.266,    8.332,    8.252,
     6      8.332,    8.252,    8.375,    8.266/
      DATA NIPL/8,8,8,8/
      DATA NOF /19,20,23,24,27,28,31,32,
     1          21,22,25,26,29,30,33,34,
     2          53,54,57,58,61,62,65,66,
     3          55,56,59,60,63,64,67,68/
C
C     INPUT HINGE LINE COORDINATES & NUMBER OF PANELS
C           IF LAST .NE. 0 NO MORE CASSES ARE READ
C
      NSUB4 = 1
      IG4 = 2
C     READ (4'IG4) E
      CALL STINDX (4,MIDX04,21)
      CALL READMS (4,SIDX04,1001,NSUB4)
      CALL STINDX (4,SIDXC4,1001)
      CALL READMS (4,E,25,IG4)
```

```
      E(5) = 1.0                                                        HNGF  046
C     WRITE (4'IG4) E                                                   HNGF  047I
      CALL WRITMS (4,E,25,IG4,1)                                        HNGF  048C
C                                                                       HNGF  049
      J = 0                                                             HNGF  050
C                                                                       HNGF  051
 1100 CONTINUE                                                          HNGF  052
C                                                                       HNGF  053
      IF (IREAD.NE.1) GO TO 1160                                        HNGF  054
C                                                                       HNGF  055
      J = J + 1                                                         HNGF  056
C                                                                       HNGF  057
      LAST = LAS(J)                                                     HNGF  058
      XHL1 = XHL(J,1)                                                   HNGF  059
      YHL1 = YHL(J,1)                                                   HNGF  060
      ZHL1 = ZHL(J,1)                                                   HNGF  061
      XHL2 = XHL(J,2)                                                   HNGF  062
      YHL2 = YHL(J,2)                                                   HNGF  063
      ZHL2 = ZHL(J,2)                                                   HNGF  064
      NIP  = NIPL(J)                                                    HNGF  065
      MIP  = NIPL(J)                                                    HNGF  066
C                                                                       HNGF  067
C                                                                       HNGF  068
C     INPUT PANEL NUMBERS                                               HNGF  069
C                                                                       HNGF  070
      DO 1150 I=1,MIP                                                   HNGF  071
 1150 NOFP(I) = NOF(I,J)                                                HNGF  072
C                                                                       HNGF  073
      GO TO 1170                                                        HNGF  074
C                                                                       HNGF  075
C                                                                       HNGF  076
C   CARD INPUT                                                          HNGF  077
C                                                                       HNGF  078
 1160 CONTINUE                                                          HNGF  079
C                                                                       HNGF  080
      READ (TAPEIN,5000) LAST,XHL1,YHL1,ZHL1,XHL2,YHL2,ZHL2,NIP         HNGF  081
 5000 FORMAT (  I2, 6F10.0, I3)                                         HNGF  082
      READ (TAPEIN,5100) (NOFP(I), I=1,NIP)                             HNGF  083
 5100 FORMAT ( 30I2 )                                                   HNGF  084
C                                                                       HNGF  085
 1170 CONTINUE                                                          HNGF  086
C                                                                       HNGF  087
C                                                                       HNGF  088
C     CHECK PANEL NOS FOR YF-12A (STOP IF WRONG)                        HNGF  089
C                                                                       HNGF  090
```

```
      DO 1200 I=1,NIP                                                   HNGF  091
      NFP = NOFP(I)                                                     HNGF  092
      IF (( NFP.LT.19 ).OR.( NFP.GT.68 )) GO TO 1300                    HNGF  093
      IF (( NFP.GT.34 ).AND.( NFP.LT.53 )) GO TO 1300                   HNGF  094
      GO TO 1200                                                        HNGF  095
 1300 CONTINUE                                                          HNGF  096
      WRITE (6,6300) I,  NFP                                            HNGF  097
 6300 FORMAT ( 1X,11HPANEL WRCNG,2X,7HPANEL =,I4,2X,6HNOFP =,I4)        HNGF  098
      STOP                                                              HNGF  099
C                                                                       HNGF  100
 1200 CONTINUE                                                          HNGF  101
C     WRITE (6,6400) (NOFP(I), I=1,NIP), LAST                           HNGF  102
C6400 FORMAT (1X,10I6)                                                  HNGF  103
C                                                                       HNGF  104
C     AXIS TRANSFORMATION TO H. L.                                      HNGF  105
C                                                                       HNGF  106
      DX = XHL1 - XHL2                                                  HNGF  107
      DY = YHL1 - YHL2                                                  HNGF  108
      DZ = ZHL1 - ZHL2                                                  HNGF  109
      DXSQ = (DX)**2                                                    HNGF  110
      DYSQ = (DY)**2                                                    HNGF  111
      DZSQ = (DZ)**2                                                    HNGF  112
      HXY = SQRT( DXSQ + DYSQ )                                         HNGF  113
      HYZ = SQRT( DXSQ + DYSQ + DZSQ )                                  HNGF  114
      IF (HXY.NE.0.0) GO TO 1400                                        HNGF  115
      PSIR = 0.0                                                        HNGF  116
      SINS = 0.0                                                        HNGF  117
      GO TO 1500                                                        HNGF  118
 1400 SINS = -( DX ) / HXY                                              HNGF  119
      IF (ABS(SINS).GT.1.0) SINS = SINS/ABS(SINS)                       HNGF  120
      PSIR = ASIN(SINS)                                                 HNGF  121
      IF ( DY.GE.0.0 ) GO TO 1500                                       HNGF  122
      PSIR = 3.141593 - PSIR                                            HNGF  123
 1500 SINH = (DZ) / HYZ                                                 HNGF  124
      IF (ABS(SINH).GT.1.0) SINH = SINH/ABS(SINH)                       HNGF  125
      PHIR = ASIN(SINH)                                                 HNGF  126
      PSI = PSIR * 0.5729578E2                                          HNGF  127
      PHI = PHIR * 0.5729578E2                                          HNGF  128
      COSS = COS(PSIR)                                                  HNGF  129
      COSH = COS(PHIR)                                                  HNGF  130
C     WRITE (6,6100) XHL1,YHL1,ZHL1, XHL2, YHL2,ZHL2, HXY,HYZ,PSI,PHI   HNGF  131
C                                                                       HNGF  132
C     OBTAIN GEOMETRY DATA                                              HNGF  133
      DO 1600 I=1,NIP                                                   HNGF  134
C                                                                       HNGF  135
```

```
C     READ PANEL TABLE ARRAY                                          HNGF  136
      J2 = NOFP(I)                                                    HNGF  137
      IG4 = J2*5                                                      HNGF  138
C     READ (4'IG4) EP                                                 HNGF  139I
      NSUB = 1                                                        HNGF  140C
      IF (NSUB.EQ.NSUB4) GO TO 405                                    HNGF  141C
      CALL STINDX (4,MIDX04,21)                                       HNGF  142C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                             HNGF  143C
      CALL READMS (4,SIDX04,1001,NSUB)                                HNGF  144C
      CALL STINDX (4,SIDX04,1001)                                     HNGF  145C
      NSUB4 = NSUB                                                    HNGF  146C
  405 CONTINUE                                                        HNGF  147C
      CALL READMS (4,EP,25,IG4)                                       HNGF  148C
C     WRITE (6,6100) EP                                               HNGF  149
C6100 FORMAT ( 1X, 10E13.6)                                           HNGF  150
      NOCK = EP(1)                                                    HNGF  151
      IF (J2.EQ.NOCK) GO TO 1700                                      HNGF  152
      WRITE (6,6500) J2, NOCK                                         HNGF  153
 6500 FORMAT ( 1X,10HERROR, J2=,I4,2X,11HBUT, NOCK =,I4)              HNGF  154
      STOP                                                            HNGF  155
C                                                                     HNGF  156
 1700 CONTINUE                                                        HNGF  157
C                                                                     HNGF  158
C     NUMBER OF ELEMENTS IN PANEL                                     HNGF  159
      NOEL = EP(4)                                                    HNGF  160
      ISTART = EP(3)                                                  HNGF  161
      ISTART = ISTART - 3                                             HNGF  162
C                                                                     HNGF  163
C     CONSIDER EACH ELEMENT                                           HNGF  164
      DO 1800 K=1,NOEL                                                HNGF  165
C                                                                     HNGF  166
C     READ ELEMENT DATA ARRAY                                         HNGF  167
      ISTART = ISTART + 3                                             HNGF  168
C     READ (4'ISTART) ELEM                                            HNGF  169I
      NSUB = (ISTART-1)/1000 + 1                                      HNGF  170C
      NSREC4 = ISTART - (NSUB-1)*1000                                 HNGF  171C
      IF (NSUB.EQ.NSUB4) GO TO 400                                    HNGF  172C
      CALL STINDX (4,MIDX04,21)                                       HNGF  173C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                             HNGF  174C
      CALL READMS (4,SIDX04,1001,NSUB)                                HNGF  175C
      CALL STINDX (4,SIDX04,1001)                                     HNGF  176C
      NSUB4 = NSUB                                                    HNGF  177C
  400 CONTINUE                                                        HNGF  178C
      CALL READMS (4,ELEM,25,NSREC4)                                  HNGF  179C
C     WRITE (6,6100) ELEM                                             HNGF  180
```

```
      KC = ELEM(1)                                                      HNGF  181
      IF (K.EQ.KC) GO TO 410                                            HNGF  182
      WRITE (6,6700) KC, K                                              HNGF  183
 6700 FORMAT ( 1X,32HERROR, ELEMENT ARRAY (ELEM) NO =,I4,2X,22HBUT, THE HNGF  184
     1COUNTER NO IS,I4)                                                 HNGF  185
      STOP                                                              HNGF  186
C                                                                       HNGF  187
  410 CONTINUE                                                          HNGF  188
C                                                                       HNGF  189
C     USING COORDINATES OF ELEMENT CENTROIDS                            HNGF  190
      XP = ELEM(7) - XHL1                                               HNGF  191
      YP = ELEM(8) - YHL1                                               HNGF  192
      ZP = ELEM(9) - ZHL1                                               HNGF  193
C                                                                       HNGF  194
      IFLAG = 1                                                         HNGF  195
 2000 CONTINUE                                                          HNGF  196
C                                                                       HNGF  197
C     APPLY ROTATION MATRIX                                             HNGF  198
      XOP = XP*COSS + YP*SINS                                           HNGF  199
      YOP =-XP*SINS*COSH + YP*COSS*COSH + ZP*SINH                       HNGF  200
      ZOP = XP*SINS*SINH - YP*COSS*SINH + ZP*COSH                       HNGF  201
C     WRITE (6,6100) XOP, YOP, ZOP                                      HNGF  202
C                                                                       HNGF  203
      IF (IFLAG.NE.1) GO TO 1900                                        HNGF  204
      XOPH = XOP                                                        HNGF  205
      YOPH = YOP                                                        HNGF  206
      ZOPH = ZOP                                                        HNGF  207
C                                                                       HNGF  208
      IFLAG = 3                                                         HNGF  209
C                                                                       HNGF  210
C     USING COMPONENTS OF UNIT NORMAL                                   HNGF  211
      XP = ELEM(4)                                                      HNGF  212
      YP = ELEM(5)                                                      HNGF  213
      ZP = ELEM(6)                                                      HNGF  214
      GO TO 2000                                                        HNGF  215
C                                                                       HNGF  216
C     CALCULATE HINGE MOMENT FACTOR (ELEM(10)= AREA OF ELEMENT)         HNGF  217
 1900 HMFCT = (-XOP*ZOPH + ZOP*XOPH) * ELEM(10)                         HNGF  218
C                                                                       HNGF  219
C     SAVE HINGE MOMENT FACTOR                                          HNGF  220
      ELEM(23) = HMFCT                                                  HNGF  221
C     WRITE (6,6100) ELEM                                               HNGF  222
C     WRITE (4'ISTART) ELEM                                             HNGF  2231
      CALL WRITMS (4,ELEM,25,NSREC4,1)                                  HNGF  224C
 1800 CONTINUE                                                          HNGF  225
```

```
 1600 CONTINUE                                                          HNGF  226
      IF (LAST.EQ.0) GO TO 1100                                         HNGF  227
C                                                                       HNGF  228
C                                                                       HNGF  229
      CALL STINDX (4,MIDX04,21)                                         HNGF  230C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                               HNGF  231C
      NSUB4 = 1                                                         HNGF  232C
      CALL READMS (4,SIDX04,1001,NSUB4)                                 HNGF  233C
      CALL STINDX (4,SIDX04,1001)                                       HNGF  234C
C                                                                       HNGF  235
      RETURN                                                            HNGF  236
      END                                                               HNGF  237
```

```
      OVERLAY (MARK4,1,1)
      PROGRAM IELE
C     SUBROUTINE IELE
C
C  THIS ROUTINE IS USED TO TRANSFER ELEMENT DATA (TYPE 3 CARDS) FROM
C  AN INPUT UNIT TO THE ELEMENT DATA SAVE UNIT
C
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1            TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR
      COMMON /GFLAG/IOUT,ISTAT3,IORIN,COMPIN
      DIMENSION  TITLE(15),XYZ1(3),XYZ2(3)
      INTEGER  STAT,STATT,TYPE,CASE,ERROR,PAGE,SEQ
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
C
      IS3 = 0
C
C  READ ELEMENT CONTROL CARD
      READ (TAPEIN,10) I3MAX,IN,IREW,I5
   10 FORMAT (2I2,2I1)
      IF (IN .EQ. 0) IN = TAPEIN
      IF (IN.NE.5 .AND. IREW.EQ.1) REWIND IN
C
      WRITE (TAPEOT,20) IN,IOUT,I3MAX
   20 FORMAT (1H0,44HTYPE 3 CARDS ARE BEING TRANSFERED FROM UNIT ,I2,
     1  9H TO UNIT ,I2,5X,6HI3MAX=,I4)
C
      IF (I3MAX .EQ. 0) I3MAX = 1
C
   30 READ (IN,40) XYZ1,STAT,XYZ2,STATT,CASE,SECT,TYPE,SEQ
   40 FORMAT (3F10.4,I1,3F10.4,I1,2X,I2,1A4,I2,4X,I4)
      IF (STAT .LE. 0) STAT = 0
      IF (STATT .LE. 0) STATT = 0
C
      IF (STAT.NE.3 .AND.STATT.NE.3) GO TO 50
      IS3 = IS3 + 1
      IF (IS3.EQ.I3MAX .AND. I3.EQ.0) GO TO 60
      IF (STAT.EQ.3) STAT = 0
      IF (STATT.EQ.3) STATT = 0
C
   50 WRITE (IOUT,40)  XYZ1,STAT,XYZ2,STATT,CASE,SECT,TYPE,SEQ
      IF (IS3 .EQ. I3MAX) GO TO 70
      GO TO 30
C
```

```
   60 WRITE (IOUT,40)  XYZ1,STAT,XYZ2,STATT,CASE,SECT,TYPE,SEQ          IELE  046
C                                                                       IELE  047
C                                                                       IELE  048
C  70 RETURN                                                            IELE  049I
   70 CONTINUE                                                          IELE  050C
      END                                                               IELE  051
```

```
      OVERLAY (MARK4,2,2)
      PROGRAM INPUT
C     SUBROUTINE INPUT
C
C  THIS ROUTINE ASSEMBLES FIXED INPUT DATA (STORED ON UNITS
C  TAPEH,  TAPEI,  AND ICP)  INTO PROPER ORDER ON UNIT TAPEIN.
C
C
      COMMON /EXEC/CASE,TITLE,IPGE,IER
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      COMMON/INDATA/IRUNID,ICORID,IPLTCP,IBETA,BETA,IWRIT,
     1     ICKPRT,ISVFMS,ISVSPL
C
C
      DIMENSION  ISQ(8),IJ(4),BB(2),A(11),B(3),CP(615),
     1     CPS(524,8),ABET(8),CARD(20),CARD1(15),CARD2(19),CARD3(10),
     2     IBW(100),IDE(4),TITLE(15)
C
      EQUIVALENCE (IBW(1),CP(1)), (CARD(1),CARD1(1),CARD2(1),CARD3(1))
C
C
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
C
      DATA IDE/0,-5,-10,-15/,CPDUM/999.999/
C
C     ICP = TAPEJ
C
      WRITE(TAPEIN, 5) TITLE,ISVFMS,ISVSPL,
     1     IRUNID,ICORID,ICKPRT,IPLTCP,IBETA,IWRIT,BETA
    5 FORMAT( 10A4, 5A1, 2I1, 3XI3, 2XI4, 4I1, F9.4)
C
      IF (IBETA .NE. 0)  GO TO 200
C
C  ALPHA CASE.  BASIC DATA ON UNIT TAPEH,   CP DATA ON UNIT ICP.
C
C  FIRST, ESTABLISH CP DATA
      IT = 0
      IN1 = TAPEH
C   10 READ(ICP,END=20)  ICOR,CONF,A,B,CP
   10 READ(ICP)  ICOR,CONF,A,B,CP
      IF (EOF(ICP)) 20, 21
   21 CONTINUE
      IF (ICOR - ICORID)  10,50,20
```

```
   20 IF (IT .EQ. 1)  GO TO 30
      IT = 1
      REWIND  ICP
      GO TO 10
C
   30 IER = 1
      WRITE(TAPEOT,40)  ICORID
   40 FORMAT(1H1,12HCORRELATION ,I4,20H COULD NOT BE FOUND./
     1        1H0,28HNEXT CASE WILL BE ATTEMPTED.)
      GO TO 650
C
C
C  BETA CASE.  BASIC DATA ON UNIT TAPEI,   CP DATA ON UNIT ICP.
C
C  FIRST ESTABLISH CP DATA
  200 IT = 0
      IN1 = TAPEI
      NBETA = 0
      ALFA = 0.0
      AMACH = 0.0
      AMBP = 0.0
C
      REWIND ICP
C 210 READ(ICP, END=220)  ICOR,CONF,A,B,CP
  210 READ(ICP)  ICOR,CONF,A,B,CP
      IF (EOF(ICP)) 220, 221
  221 CONTINUE
C
C  CONVERT RUN NUMBER TO INTEGER FORM (FIRST 3 DIGITS OF B(3)).
      REWIND 88
      WRITE(88,8800) B(3)
 8800 FORMAT (A4)
      REWIND 88
      READ(88,8810)  IRUN
 8810 FORMAT (I3)
C
      IF (IRUN - IRUNID) 210,250,220
C
  220 IF (IT - 1)  230,230,270
C
  230 IER = 1
      WRITE(TAPEOT,240)  IRUNID
  240 FORMAT(1H0, 4HRUN , I3,20H COULD NOT BE FOUND./
     1        1H0,28HNEXT CASE WILL BE ATTEMPTED.)
      GO TO 650
```

```
C
  250 NBETA = NBETA + 1
      ABET(NBETA) = A(2)
      ALFA = ALFA + A(1)
      AMACH = AMACH + A(3)
      AMBP = AMBP + A(9)
      DO 260 I = 1,524
  260 CPS(I,NBETA) = CP(I)
      IT = 2
      GO TO 210
C
  270 CONTINUE
      A(1) = ALFA/NBETA
      A(2) = BETA
      A(3) = AMACH/NBETA
      A(9) = AMBP/NBETA
C
C  ARRANGE INDICES FOR ASCENDING ORDER OF ABET
      IF (NBETA .GE. 2)  GO TO 274
      IJ(4) = 1
      IJ(3) = 1
      IJ(2) = 1
      IJ(1) = 1
      BB(1) = 0.0
      BB(2) = 0.0
      GO TO 50
  274 DO 275 I = 1,NBETA
  275 ISQ(I) = I
      N = NBETA - 1
      DO 276 I = 1,N
      I1 = ISQ(I)
      J1 = I + 1
      DO 276 J = J1,NBETA
      I2 = ISQ(J)
      IF (ABET(I1) .LE. ABET(I2))  GO TO 276
      ISQ(I) = I2
      ISQ(J) = I1
      I1 = I2
  276 CONTINUE
C
C  SET UP COUNTERS AND CONSTANTS FOR CP INTERPOLATION (LINEAR)
      DO 290 I = 2,NBETA
      I1 = I - 1
      I2 = ISQ(I)
```

```
      IF (BETA .LT. ABET(I2))  GO TO 290
  280 CONTINUE
  290 IJ(1) = ISQ(I1)
      IJ(2) = I2
      RB(1) = (BETA-ABET(IJ(1)))/(ABET(IJ(2))-ABET(IJ(1)))
C
      BETAM = -BETA
      DO 300 I = 2,NBETA
      I1 = I - 1
      I2 = ISQ(I)
      IF (BETAM .LT. ABET(I2))  GO TO 310
  300 CONTINUE
  310 IJ(3) = ISQ(I1)
      IJ(4) = I2
      RB(2) = (BETAM-ABET(IJ(3)))/(ABET(IJ(4))-ABET(IJ(3)))
C
C  ESTABLISH ELEVON DEFLECTION
   50 ID = 1
      IF (IRUNID.GE.27 .AND. IRUNID.LE.164)  ID = 3
      IF (IRUNID.GE.41 .AND. IRUNID.LE.125)  ID = 1
      IF (IRUNID.GE.67 .AND. IRUNID.LE. 76)  ID = 3
      IF (IRUNID.GE.77 .AND. IRUNID.LE. 86)  ID = 4
      IF (IRUNID.GE.87 .AND. IRUNID.LE.101)  ID = 2
C
C  START ASSEMBLING CASE INPUT DATA
      REWIND IN1
C
C  AERO FLAG CARD (P.69)
      READ(IN1,1000)  CARD
 1000 FORMAT(20A4)
      WRITE(TAPEIN,1000)  CARD
C
C  FLIGHT CONDITION CARD (P.70)
      READ(IN1,1010)  AMACH,ALT,PSTAG,TSTAG,IGAS,NAB,AMBP
 1010 FORMAT( 4F10.4, I1,I2, 7XF10.4)
      WRITE(TAPEIN,1010) A(3),ALT,PSTAG,TSTAG,IGAS,NAB,A(9)
C
C  REFERENCE DIMENSION CARD (P.70)
      READ(IN1,1000)  CARD
      WRITE(TAPEIN,1000)  CARD
C
C  ALPHA-BETA CARD (P.71)
      READ(IN1,1020)  ALPHA,BETA,CARD1
 1020 FORMAT( 2F10.4, 15A4)
      WRITE(TAPEIN,1020) A(1), A(2), CARD1
```

```
C                                                                       INPT  181
C  FLOW FIELD MASTER DIRECTORY CARD (P.75)                              INPT  182
      READ (IN1,1000) CARD                                              INPT  183
      WRITE(TAPEIN,1000)  CARD                                          INPT  184
C                                                                       INPT  185
C  DATA SET DIRECTORY CARD (P.75)                                       INPT  186
   55 CONTINUE                                                          INPT  187
      READ(IN1,1025)  LASTS,NEWS,NSET,DMACH,CARD3                       INPT  188
 1025 FORMAT(2I1, 1XI1,F6.0, 10X, 10A4)                                 INPT  189
      WRITE(TAPEIN,1025)  LASTS,NEWS,NSET,DMACH,CARD3                   INPT  190
C                                                                       INPT  191
C  ALPHA-BETA DIRECTORY CARD (P.76)                                     INPT  192
      IAB = 0                                                           INPT  193
      IAR = 0                                                           INPT  194
   60 IAB = IAB + 1                                                     INPT  195
      READ(IN1,1030) LASTAB,NEWAB,IA,ALPHA,BETA,CARD3                   INPT  196
 1030 FORMAT(2I1,I2, 2F6.2, 4X10A4)                                     INPT  197
      WRITE(TAPEIN,1035)LASTAB,NEWAB,IAB,A(1),A(2),IPJNID,ICORID,IDEIID INPT  198
 1035 FORMAT(2I1,I2, 2F6.2, 4X, 4HRUN ,I3,4HCOR , I4,3HDE ,I3)          INPT  199
C                                                                       INPT  200
C  REGION DIRECTORY CARD (P.76)                                         INPT  201
   70 IAR = IAR + 1                                                     INPT  202
      READ(IN1,1036)  LASTR, CARD                                       INPT  203
 1036 FORMAT(I1,A3,19A4)                                                INPT  204
      WRITE(TAPEIN,1036)  LASTR, CARD                                   INPT  205
C                                                                       INPT  206
C  FLOW FIELD CONTROL AND ORIENTATION CARD (P.89)                       INPT  207
      READ(IN1,1040) NSREG,IFC,ITFLAG,CARD2                             INPT  208
 1040 FORMAT(I1,I4,I1,A2,18A4)                                          INPT  209
      ITFLAG = 1                                                        INPT  210
      WRITE(TAPEIN,1040) NSREG,IFC,ITFLAG,CARD2                         INPT  211
C                                                                       INPT  212
C                                                                       INPT  213
      IF (IBETA .NE. 0)  GO TO 400                                      INPT  214
      IF (IAR .GT. 18)  GO TO 90                                        INPT  215
C                                                                       INPT  216
      DO 80 I = 1,IFC                                                   INPT  217
      READ(IN1,1050) X,Y,Z,ITAP,LOC                                     INPT  218
 1050 FORMAT( 3F10.4, 2I5)                                              INPT  219
      WRITE(TAPEIN,1060) X,Y,Z,CP(LOC)                                  INPT  220
 1060 FORMAT( 4F10.4)                                                   INPT  221
   80 CONTINUE                                                          INPT  222
      GO TO 110                                                         INPT  223
C  ELEVON GEOMETRY                                                      INPT  224
   90 CONTINUE                                                          INPT  225
```

```
      DO 100 IE = 1,4                                                   INPT  226
      DO 100  I = 1,IFC                                                 INPT  227
      READ(IN1,1050)  X,Y,Z,ITAP,LOC                                    INPT  228
      IF (IE .NE. ID)  GO TO 100                                        INPT  229
      WRITE(TAPEIN,1060)  X,Y,Z,CP(LOC)                                 INPT  230
  100 CONTINUE                                                          INPT  231
  110 CONTINUE                                                          INPT  232
      IF (LASTR .EQ. 0)  GO TO 70                                       INPT  233
      IF (LASTAB.EQ. 0)  GO TO 60                                       INPT  234
      IF (LASTS .EQ. 0)  GO TO 55                                       INPT  235
C                                                                       INPT  236
C  PRESSURE TITLE CARD (P.101)                                          INPT  237
      READ(IN1,1070)  NCOMP,CARD                                        INPT  238
 1070 FORMAT(I2,A2,19A4)                                                INPT  239
      WRITE(TAPEIN,1070)  NCOMP,CARD                                    INPT  240
C                                                                       INPT  241
C                                                                       INPT  242
      DO 130 J = 1,NCOMP                                                INPT  243
C                                                                       INPT  244
C  COMPONENT ORGANIZATION CARD (P.101)                                  INPT  245
      READ(IN1,1075) IPAN1,IPAN2,CARD2                                  INPT  246
 1075 FORMAT( 2I2,19A4)                                                 INPT  247
C                                                                       INPT  248
C  SET ELEVON PANEL NUMBER                                              INPT  249
      IF (IPAN1 .LT. 19)  GO TO 115                                     INPT  250
      IF (IPAN1 .LT. 35)  GO TO 111                                     INPT  251
      IF (IPAN1 .LT. 53)  GO TO 115                                     INPT  252
  111 IPAN1 = IPAN1 + (ID-1)*4                                          INPT  253
C                                                                       INPT  254
  115 WRITE(TAPEIN,1075)  IPAN1,IPAN2,CARD2                             INPT  255
C                                                                       INPT  256
C                                                                       INPT  257
C  PRESSURE METHOD CARD (P.103)                                         INPT  258
      READ(IN1,1000)  CARD                                              INPT  259
      WRITE(TAPEIN,1000)  CARD                                          INPT  260
C                                                                       INPT  261
C  SURFACE PROPERTY ACCESS FLAG CARD (P.111)                            INPT  262
      READ(IN1,1080) CARD3, NN                                          INPT  263
 1080 FORMAT( 9A4, A3, I4)                                              INPT  264
      NS = NN                                                           INPT  265
      IF ((NS.EQ.1) .AND. (IPLTCP.EQ.0))  NN = 0                        INPT  266
      WRITE(TAPEIN,1080)  CARD3,NN                                      INPT  267
C                                                                       INPT  268
C  NORMALIZATION DATA CARDS (P.113)                                     INPT  269
      DO 120 I = 1,4                                                    INPT  270
```

```
      READ(IN1,1000)  CARD                                              INPT  271
      WRITE(TAPEIN,1000)  CARD                                          INPT  272
  120 CONTINUE                                                          INPT  273
C                                                                       INPT  274
      IF (NS .NE. 1)  GO TO 130                                         INPT  275
C  SPECIAL CP PLOT OPTION CARDS                                         INPT  276
  124 READ(IN1,1050)  NPAN,NR,MOREP                                     INPT  277
 1090 FORMAT(3I5)                                                       INPT  278
      READ(IN1,1100)  (IRW(I), I = 1,NR)                                INPT  279
 1100 FORMAT(14I5)                                                      INPT  280
C                                                                       INPT  281
      IF (IPLTCP .EQ. 0)  GO TO 125                                     INPT  282
      WRITE(TAPEIN,1090)  NPAN,NR,MOREP                                 INPT  283
      WRITE(TAPEIN,1100)  (IRW(I), I = 1,NR)                            INPT  284
  125 IF (MOREP .EQ. 1)  GO TO 124                                      INPT  285
C                                                                       INPT  286
  130 CONTINUE                                                          INPT  287
C                                                                       INPT  288
C  SPECIAL ROUTINES OPTION CARD (P.130)                                 INPT  289
      READ(IN1,1000)  CARD                                              INPT  290
      WRITE(TAPEIN,1000)  CARD                                          INPT  291
C                                                                       INPT  292
C  SUMMATION CONTROL CARDS (P.131)                                      INPT  293
  140 READ(IN1,1110)       LAST, CARD                                   INPT  294
      WRITE(TAPEIN,1110)  LAST, CARD                                    INPT  295
 1110 FORMAT(I1,A3,19A4)                                                INPT  296
      READ(IN1,1000)  CARD                                              INPT  297
      WRITE(TAPEIN,1000)  CARD                                          INPT  298
      IF (LAST .EQ. 0)  GO TO 140                                       INPT  299
C                                                                       INPT  300
C                                                                       INPT  301
      GO TO 600                                                         INPT  302
C                                                                       INPT  303
C                                                                       INPT  304
  400 IF (IAR .GT. 18)  GO TO 470                                       INPT  305
      DO 460 I = 1,IFC                                                  INPT  306
      ISYM = 0                                                          INPT  307
      READ(IN1,1050) X,Y,Z,ITAP,LOC                                     INPT  308
      IF (ITAP)  430,420,410                                            INPT  309
  410 I2 = IJ(4)                                                        INPT  310
      I1 = IJ(3)                                                        INPT  311
      IR = 2                                                            INPT  312
      GO TO 440                                                         INPT  313
C  DESIRED TAP NO. (ITAP) IS ZERO.  SET DUMMY CP AND CONTINUE           INPT  314
  420 CPINPT = CPDUM                                                    INPT  315
```

```
      GO TO 450
  430 I2 = IJ(2)
      I1 = IJ(1)
      IB = 1
  440 CONTINUE
      CPINPT = CPS(LOC,I1) + (CPS(LOC,I2)-CPS(LOC,I1))*BB(IB)
      IF (Y .NE. 0.0)  GO TO 450
      IF (ISYM .EQ. 1) GO TO 445
      ISYM = 1
      CP1 = CPINPT
      IF (IB-1) 410,410,430
  445 CPINPT = (CPINPT + CP1)*0.5
  450 WRITE(TAPFIN,1060)  X,Y,Z,CPINPT
  460 CONTINUE
      GO TO 110
C
C  ELEVON GEOMETRY
  470 CONTINUE
      DO 530 IE = 1,4
      DO 530  I = 1,IEC
      READ(IN1,1050) X,Y,Z,ITAP,LOC
      IF (IE .NE. ID)  GO TO 530
      ISYM = 0
      IF (ITAP)  500,490,480
  480 I2 = IJ(4)
      I1 = IJ(3)
      IB = 2
      GO TO 510
  490 CPINPT = CPDUM
      GO TO 520
  500 I2 = IJ(2)
      I1 = IJ(1)
      IB = 1
  510 CONTINUE
      CPINPT = CPS(LOC,I1) + (CPS(LOC,I2)-CPS(LOC,I1))*BB(IB)
      IF (Y .NE. 0.0)  GO TO 520
      IF (ISYM .EQ. 1) GO TO 515
      ISYM = 1
      CP1 = CPINPT
      IF (IB-1) 460,490,500
  515 CPINPT = (CPINPT + CP1)*0.5
  520 WRITE(TAPFIN,1060)  X,Y,Z,CPINPT
  530 CONTINUE
C
      GO TO 110
```

```
C                                                                       INPT  361
C                                                                       INPT  362
  600 IER = 0                                                           INPT  363
      REWIND TAPEIN                                                     INPT  364
  650 CONTINUE                                                          INPT  365
C     RETURN                                                            INPT  366I
      END                                                               INPT  367
```

```
      SUBROUTINE  MIS1(A, N, ND, B, MD, NX, MX, NERR, D)
C
C     MIS1 - MATRIX INVERT OR SOLVE
C
C     THIS SUBROUTINE SOLVES THE MATRIX EQUATION  A*C = B  WHERE A AND
C          B ARE INPUT.  IF B IS A UNIT MATRIX, THEN C IS THE INVERSE
C          OF A.
C
C          A     = IN-OUT = A TWO DIMENSIONAL ARRAY (DIMENSIONED ND BY
C                           ND) CONTAINING AN INPUT MATRIX OF ORDER N.
C                           THE A MATRIX IS DESTROYED BY THE SUBROUTINE.
C
C          N     = INPUT  = THE ORDER (INTEGER) OF A (N LESS THAN OR
C                           EQUAL TO ND) AND THE NUMBER OF ROWS IN B.
C
C          ND    = INPUT  = THE MAXIMUM DIMENSIONS (INTEGER) OF THE
C                           SQUARE MATRIX A.
C
C          B     = IN-OUT = A TWO DIMENSIONAL ARRAY (DIMENSIONED ND BY
C                           MD) CONTAINING THE SECOND INPUT MATRIX.
C                           THIS ARRAY IS MODIFIED AND CONTAINS (UPON
C                           OUTPUT) THE C MATRIX SOLUTION.
C
C          MD    = INPUT  = THE NUMBER (INTEGER) OF COLUMNS IN B.
C
C          NERR  = OUTPUT = OUTPUT CODE (INTEGER)
C                         = 0  IF A IS NON-SINGULAR
C                         = 1  IF A IS SINGULAR
C                         = 2  IF N IS LESS THAN OR EQUAL TO ZERO
C                         = 3  IF N IS GREATER THAN ND
C
C          D     = INPUT  = SCALE FACTOR TO SCALE DETERMINENT.
C                = OUTPUT = THE SCALED DETERMINENT OF A.
C
      DIMENSION  A(ND,ND), B(NX,MX)
      EQUIVALENCE  (L,FL), (K,FK)
C
C     CHECK THE VALUE OF N
C
      IF  ( N - 1 ) 10 , 20 , 30
C
   10 D = 0.0E0
      NERR = 2
      GO TO 190
```

```
C
   20 AIJMAX = A(1,1)
      IF  ( AIJMAX .EQ. 0.0E0 )  GO TO 60
      B(1,1) = B(1,1) / AIJMAX
      D = D * AIJMAX
      GO TO 190
C
   30 IF  ( N .LE. ND )  GO TO 40
      NERR = 3
      GO TO 190
C
C     START REDUCTION OF MATRIX A
C
   40 DO 130   I = 1, N
C
C     SEARCH FOR MAXIMUM ELEMENT IN ITH ROW OF A-MATRIX
C
      AIJMAX = A(I,1)
      JMAX = 1
      DO 50   J = 2, N
      IF  ( ABS( A(I,J) ) .LE. ABS( AIJMAX ) )  GO TO 50
      AIJMAX = A(I,J)
      JMAX = J
   50 CONTINUE
C
C     IF AIJMAX IS ZERO, THE MATRIX IS SINGULAR
C
      IF  ( ABS( AIJMAX ) .GT. 0.0E0 )  GO TO 70
   60 D = 0.0E0
      NERR = 1
      GO TO 190
C
C     NORMALIZE ITH ROW BY AIJMAX (JMAX ELEMENT OF ITH ROW)
C
   70 DO 80   J = 1, N
   80 A(I,J) = A(I,J) / AIJMAX
C
C     THIS CALCULATION VOIDED TO REMOVE UNDERFLOW ERRORS FOR MATRICES
C     THAT ARE ALMOST, BUT NOT QUITE, SINGULAR
C     D = D * AIJMAX
C
C     NORMALIZE ITH ROW OF B
C
      DO 90   J = 1, MD
   90 B(I,J) = B(I,J) / AIJMAX
```

```
C                                                                       MIS1  091
C     USE ROW TRANSFORMATIONS TO GET ZEROS ABOVE AND BELOW THE JMAX     MIS1  092
C     ELEMENT OF THE ITH ROW OF A.  APPLY SAME ROW TRANSFORMATIONS      MIS1  093
C     TO THE B MATRIX.                                                  MIS1  094
C                                                                       MIS1  095
      DO 120   K = 1, N                                                 MIS1  096
      IF  ( K .EQ. I )  GO TO 120                                       MIS1  097
      ARAT = -A(K,JMAX)                                                 MIS1  098
      DO 100   J = 1, N                                                 MIS1  099
      IF  ( ABS( A(I,J) ) .EQ. 0.0E0 )  GO TO 100                       MIS1  100
      A(K,J) = ARAT * A(I,J) + A(K,J)                                   MIS1  101
  100 CONTINUE                                                          MIS1  102
      A(K,JMAX) = 0.0E0                                                 MIS1  103
      DO 110   J = 1, MD                                                MIS1  104
      IF  ( ABS( B(I,J) ) .EQ. 0.0E0 )  GO TO 110                       MIS1  105
      B(K,J) = ARAT * B(I,J) + B(K,J)                                   MIS1  106
  110 CONTINUE                                                          MIS1  107
  120 CONTINUE                                                          MIS1  108
C                                                                       MIS1  109
C     STORE ROW COUNTER (I) IN TOP ELEMENT OF JMAX COLUMN. THUS,        MIS1  110
C     THE TOP ROW OF A WILL CONTAIN THE LOC OF THE PIVOT (UNITY)        MIS1  111
C     ELEMENT OF EACH COLUMN (AFTER REDUCTION).                         MIS1  112
C                                                                       MIS1  113
      I = I                                                             MIS1  114
  130 A(1,JMAX) = FL                                                    MIS1  115
C                   THIS STORES INTEGER IN TOP ROW OF A                 MIS1  116
C                                                                       MIS1  117
C     THE REDUCTION OF A IS NOW COMPLIETE. PERFORM ROW INTERCHANGES     MIS1  118
C     AS INDICATED IN THE FIRST ROW OF A.                               MIS1  119
C                                                                       MIS1  120
      DO 170   I = 1, N                                                 MIS1  121
      K = I                                                             MIS1  122
  140 FK = A(1,K)                                                       MIS1  123
C                   THIS PUTS THE INTEGER VALUE IN A INTO K             MIS1  124
      IF  ( K - I ) 140 , 170 , 150                                     MIS1  125
C                                                                       MIS1  126
C     IF K(1,I) IS LESS THAN I, THEN THAT ROW HAS ALREADY BEEN          MIS1  127
C     INVOLVED IN AN INTERCHANGE, AND WE USE K(1,K) UNTIL WE GET        MIS1  128
C     A VALUE OF K GREATER THAN I  (CORRESPONDING TO A ROW STORED       MIS1  129
C     BELOW THE ITH ROW).                                               MIS1  130
C                                                                       MIS1  131
  150 DO 160   J = 1, MD                                                MIS1  132
      ARAT = B(I,J)                                                     MIS1  133
      B(I,J) = B(K,J)                                                   MIS1  134
  160 B(K,J) = ARAT                                                     MIS1  135
```

```
      D = -D                                                    MISL  136
  170 CONTINUE                                                  MISL  137
  180 NERR = 0                                                  MISL  138
  190 RETURN                                                    MISL  139
      END                                                       MISL  140
```

```
      SUBROUTINE MONITR
C
C     THIS ROUTINE IS USED TO TRANSFER THE INPUT DATA TO A SCRATCH UNIT
C
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1            TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      INTEGER  ERROR,PAGE,CASE
      DIMENSION  CARD(20),TITLE(15)
      DATA BLANK/4H    /
      CASE = 0
      DO 5 I=1,15
    5 TITLE(I) = BLANK
C
   10 CALL HEADER
      WRITE (TAPEOT,20)
   20 FORMAT (1H0,50HSUPERSONIC-HYPERSONIC ARBITRARY-BODY PROGRAM INPUT,
     1  5H DATA ,/1H )
      WRITE (TAPEOT,25)
   25 FORMAT (1H ,49H0         1         2         3         4         ,
     1 31H5         6         7         8,/1H ,20H12345678901234567890,
     2 60H12345678901234567890123456789012345678901234567890123456789012345678901/)
      I = 10
C
C  30 READ (TAPEIN,40,END=60) CARD
   30 READ (TAPEIN,40) CARD
      IF (EOF(TAPEIN)) 60,41
   40  FORMAT (20A4)
   41 IF (TAPEIN.EQ.1) GO TO 42
      WRITE (1,40) CARD
   42 CONTINUE
      WRITE (TAPEOT,50) CARD
   50  FORMAT (1H ,20A4)
      I = I + 1
      IF (I .EQ. 51) GO TO 10
      GO TO 30
C
   60 REWIND 1
      TAPEIN = 1
      RETURN
      END
```

```
      SUBROUTINE  NACEL                                                 NACE  001
C                                                                       NACE  002
      DIMENSION  PODORG(9,6),XPOD(9,30),PODR(9,30),X(2),Y(2),Z(2),      NACE  003
     1           IOR(9),IEL(9),NEL(9)                                   NACE  004
C                                                                       NACE  005
      COMMON /ACFT/IPRINT,ITAPE,ITYPE,PI,DX,CASE                        NACE  006
C                                                                       NACE  007
      COMMON /NACELD/NP,NPODOR,PODORG,XPOD,PODR,IOR,IEL,NEL             NACE  008
C                                                                       NACE  009
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   NACE  010
     1            TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                         NACE  011
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,       NACE  012
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                            NACE  013
      INTEGER CASE                                                      NACE  014
      INTEGER STAT1,STAT2                                               NACE  015
C                                                                       NACE  016
C I IS THE NACELLE OR POD NUMBER                                        NACE  017
C J IS NUMBER IDENTIFYING AXIAL LOCATION                                NACE  018
C N IS THE POINT COUNTER ON THE SURFACE AT A GIVEN STATION              NACE  019
C                                                                       NACE  020
C                                                                       NACE  021
      I = 0                                                             NACE  022
   10 I = I + 1                                                         NACE  023
      IF (I .GT. NP)  GO TO 140                                         NACE  024
      ISEQ = 0                                                          NACE  025
C                                                                       NACE  026
C SET COUNTERS AND CONTROLS                                             NACE  027
      STAT1 = 2                                                         NACE  028
      STAT2 = 0                                                         NACE  029
      J = 1                                                             NACE  030
C                                                                       NACE  031
C SET POD ORIGIN QUANTITES                                              NACE  032
      XO = PODORG(I,1)                                                  NACE  033
      YO = PODORG(I,2)                                                  NACE  034
      ZO = PODORG(I,3)                                                  NACE  035
C                                                                       NACE  036
C SET UP THE ROTATION ANGLES FOR POD ORIENTATION                        NACE  037
      STHETA = 0.0                                                      NACE  038
      CTHETA = 1.0                                                      NACE  039
      SPSI = 0.0                                                        NACE  040
      CPSI = 1.0                                                        NACE  041
      IF (IOR(I) .NE. 1) GO TO 30                                       NACE  042
C                                                                       NACE  043
      DXX = (XO - PODORG(I,4))*DX                                       NACE  044
      DYY =  YO - PODORG(I,5)                                           NACE  045
```

```
      DZZ = -ZO + PODORG(I,6)
      DS1 = DXX**2 + DYY**2
      DS2 = SQRT(DS1 + DZZ**2)
      IF (DS1 .LT. 0.0001) GO TO 20
      DS1 = SQRT(DS1)
      CPSI = DXX/DS1
      SPSI = DYY/DS1
C
   20 CTHETA = DS1/DS2
      STHETA = DZZ/DS2
C
C CALCULATE ELEMENT SPACING, DPHI.
   30 NR = 12
      IF (IEL(I) .NE. 1) GO TO 40
      IF (NEL(I) .GT. 0 .AND. NEL(I) .LE. 36) NR = NEL(I)
   40 DPHI = PI/NR
C
C CHECK POD SYMMETRY WITH RESPECT TO X-Z ORIGIN PLANE.
      IF (ABS(YO) .GT. 1.E-5) GO TO 50
      IF (IOR(I) .NE. 1) GO TO 60
      IF (ABS(PODORG(I,5)) .LE. 1.E-5) GO TO 60
   50 NR = 2*NR
   60 NR = NR + 1
C
C NOTE. THE Y- AND Z-COORDINATES ARE DEFINED AS A FUNCTION
C       OF ANGLE PHI WHICH IS DEFINED TO BE -PI/2 AT THE
C       NEGATIVE Z-AXIS.  THIS DEFINITION ALLOWS THE SIGN
C       OF THE DELTA-Z INCREMENT TO BE AUTOMATICALLY ACCOUNTED FOR.
   70 N = 0
      PHI = -PI/2. -DPHI
C
   80 K = 0
   90 N = N + 1
      IF (N .LE. NR) GO TO 100
C
C SET DUMMY POINT.
      K = 2
      STAT2 = 0
      X(K) = X(1)
      Y(K) = Y(1)
      Z(K) = Z(1)
      GO TO 110
C
  100 K = K + 1
      PHI = PHI + DPHI
```

```
      YP = PODR(I,J)*COS(PHI)                                           NACE  091
      ZP = PODR(I,J)*SIN(PHI)                                           NACE  092
C                                                                       NACE  093
      X(K) = DX*XO - XPOD(I,J)*CTHETA*CPSI - YP*SPSI + ZP*STHETA*CPSI   NACE  094
C                                                                       NACE  095
      Y(K) = YO - XPOD(I,J)*CTHETA*SPSI + YP*CPSI + ZP*STHETA*SPSI      NACE  096
C                                                                       NACE  097
      Z(K) = ZO + XPOD(I,J)*STHETA + ZP*CTHETA                          NACE  098
C                                                                       NACE  099
      IF (K .EQ. 1) GO TO 90                                            NACE  100
C                                                                       NACE  101
C WRITE DATA                                                            NACE  102
  110 ISEQ = ISEQ + 1                                                   NACE  103
      WRITE(ITAPE,120) X(1),Y(1),Z(1),STAT1,                            NACE  104
     1                 X(2),Y(2),Z(2),STAT2,CASE,ITYPE,I,ISEQ           NACE  105
C                                                                       NACE  106
  120 FORMAT (3F10.4,I1,3F10.4,I1,2XI2,5XI1,3HPOD,I1,I4)                NACE  107
C                                                                       NACE  108
      IF (IPRINT .EQ. 1)                                                NACE  109
     1WRITE (TAPEOT,130) X(1),Y(1),Z(1),STAT1,                          NACE  110
     2                 X(2),Y(2),Z(2),STAT2,CASE,ITYPE,I,ISEQ           NACE  111
C                                                                       NACE  112
  130 FORMAT (1H ,3F10.4,I1,3F10.4,I1,2XI2,5XI1,3HPOD,I1,I4)            NACE  113
C                                                                       NACE  114
C RESET STATUS FLAG                                                     NACE  115
      IF (STAT1 .GT. 0) STAT1 = 0                                       NACE  116
C                                                                       NACE  117
C CHECK FOR PROPER CYCLING                                              NACE  118
      IF (N .LT.NR) GO TO 80                                            NACE  119
C                                                                       NACE  120
C HAVE COMPLETED AN AXIAL STATION.  CHECK FOR MORE                      NACE  121
      J = J + 1                                                         NACE  122
      IF (J .GT. NPODOR)  GO TO 10                                      NACE  123
C                                                                       NACE  124
C ONE OR MORE STATIONS LEFT.  SET STATUS FLAG AND CONTINUE              NACE  125
      STAT1 = 1                                                         NACE  126
      GO TO 70                                                          NACE  127
  140 RETURN                                                            NACE  128
      END                                                               NACE  129
```

```
      OVERLAY (MARK4,2,3)                                                PRES  001C
      PROGRAM PRES                                                       PRES  002C
C     SUBROUTINE PRES                                                    PRES  003I
C                                                                        PRES  004
C  THIS IS THE EXECUTIVE ROUTINE FOR THE INVISCID PRESSURE CALCULATIONS  PRES  005
C                                                                        PRES  006
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR                                 PRES  007
      COMMON /GDATA/LTOT,J,SYMFCT,IORN,IGTYPE,L,LTOTAB(20)               PRES  008
      COMMON /FLIGHT/MACH,ALT,PSTAG,TSTAG,V,RENO,PFS,TFS,AFS,RHOFS,VIS   PRES  009
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,    PRES  010
     1            TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                          PRES  011
      COMMON /ABDATA/NAB,ALPHA(20),BETA(20),ROL(20),CDELTA(20),QI(20),   PRES  012
     1               RI(20),PI(20)                                       PRES  013
      COMMON /REF/SREF,MAC,SPAN,XCG,YCG,ZCG                              PRES  014
      COMMON /FARRAY/F                                                   PRES  015
      COMMON /MDATA/IMPACT,ISHAD,IPRINT,PDATA,ISAVE,IPIN                 PRES  016
      COMMON /GASD/GAM,GASCP,PRAN,IGAS,AV1,AV2,AV3,GTYPE(2)              PRES  017
      COMMON/SURFN2/INORM,ISURF,IX,IY,IZ,U1,UL,V1,VL,CR,CT,CHX,N2,N3,    PRES  018
     1     XO,YO,ZO,AP11,AP12,AP13,AP21,AP22,AP23,AP31,AP32,AP33,        PRES  019
     2  FLOWC(7),XI(500),YI(500),B(503,7),XKF(503),AA(5999),             PRES  020
     A  XB(4), YB(4), ZB(4),                                             PRES  021
     3  NDSET,IABSET,IR,LORG(20),ISR(20),NS,ISFR,IVIS, IFTYP,IFLOW,      PRES  022
     4     BX(2,20), BY(2,20), NB                                        PRES  023
      COMMON/SAVSPL/NEXPAN,NUMPAN,NUMCOR,NEXSPL,IPDP(72),IPSP(72),       PRES  024
     1     IPAN(72),IPANL                                                PRES  025
      COMMON/SUBIDX/SIDX04(1001),SIDX16(1001),SIDX21(1001)               PRES  026C
      COMMON /MASTER/ MIDX04(21),INDEX9(501),INDX10(2001),INDX15(2001),  PRES  027C
     1  MIDX16(94),INDX20(2421),MIDX21(73)                               PRES  028C
      DIMENSION ELEM(25),                          TITLE(15),E(25),EP(25) PRES  029
      DIMENSION F(11),IPANL(10),PDATA(6),IPR(20),FF(11),IMP(20),ISA(20), PRES  030
     1  ISH(20),PDA(6,20),COM(10)                                        PRES  031
      DIMENSION LCOM(81),LSUM(41),TITLE9(5)                              PRES  032
      DIMENSION  INFI(22),DINFI(6),E1(8),E2(11),E4(41),E5(41),E6(41),    PRES  033
     1           E7(11),E8(14)                                           PRES  034
      DIMENSION IG4S(1000)                                               PRES  035
      DIMENSION  INP(20)                                                 PRES  036
      INTEGER ERROR,SYMFCT,PAGE,CASE                                     PRES  037
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,        PRES  038
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                             PRES  039
      INTEGER SIDX04,SIDX16,SIDX21                                       PRES  040C
      REAL  MACH,MAC                                                     PRES  041
      DATA ISIZE/1000/                                                   PRES  042
      DATA TITLE9/10HFORCE DATA,10H SAVE UNIT,10H                        PRES  043C
     1 10H          ,10H          /                                      PRES  044C
C     DATA TITLE9/4HFORC,4HE DA,4HTA S,4HAVE ,4HUNIT/                    PRES  045I
```

```
C                                                                       PRES  046
C                                                                       PRES  047
C   SET INITIAL CONSTANTS FOR START OF CASE                             PRES  048
      ERROR = 0                                                         PRES  049
      NCOM = 0                                                          PRES  050
      I9 = 0                                                            PRES  051
C                                                                       PRES  052
      WRITE (TAPEOT,10)                                                 PRES  053
   10  FORMAT (1H1,32H*** PRESSURE CALCULATION PROGRAM  )               PRES  054
C                                                                       PRES  055
C   READ TITLE CARD AND FLAGS                                           PRES  056
      READ (TAPEIN,20) NCOMP,IFSAVE,TITLE                               PRES  057
   20  FORMAT (I2,I1,3X,15A4)                                           PRES  058
C   SET UP CONSTANTS FOR FORCE DATA SAVE ON UNIT 9                      PRES  059
      IF (IFSAVE .EQ. 2) GO TO 55                                       PRES  060
      IF (IFSAVE .NE. 0) GO TO 40                                       PRES  061
C                                                                       PRES  062
C                                                                       PRES  063
C   INITIALIZE CONSTANTS FOR UNIT 9 FORCE DATA SAVE UNIT                PRES  064
C         NTOT - TOTAL NUMBER OF COMPONENTS SAVED                       PRES  065
C         ISUM - TOTAL NUMBER OF SUMMATION SETS                         PRES  066
C         NEXT - NEXT EMPTY RECORD ON UNIT 9                            PRES  067
C         LCOM - POINTERS TO COMPONENT DATA (41)  1ST RECORD IS DATA TYPE  PRES  068
C         LSUM - POINTERS TO SUMMATION DATA (41)  1ST RECORD IS DATA TYPE  PRES  069
C                (DATA TYPE=1 FORCE ONLY,  =2 FORCE + STABILITY DATA)   PRES  070
      NTOT = 0                                                          PRES  071
      ISUM = 0                                                          PRES  072
      NEXT = 10                                                         PRES  073
      DO 21 I=1,41                                                      PRES  074
       LCOM(I) = 0                                                      PRES  075
      LCOM(40+I) = 0                                                    PRES  076
   21  LSUM(I) = 0                                                      PRES  077
C                                                                       PRES  078
C   INITIALIZE DIRECTORIES                                              PRES  079
      E1(1)=NTOT                                                        PRES  080C
      E1(2)=ISUM                                                        PRES  081C
      E1(3)=NEXT                                                        PRES  082C
      DO 15 I=4,8                                                       PRES  083C
   15 E1(I)=0.0                                                         PRES  084C
      DO 16 I=1,41                                                      PRES  085C
      E5(I)=0.0                                                         PRES  086C
   16 E6(I)=0.0                                                         PRES  087C
      DO 17 I=1,14                                                      PRES  088C
   17 E8(I)=0.0                                                         PRES  089C
      CALL WRITMS (9,E1,8,1,0)                                          PRES  090C
```

```
      CALL WRITMS (9,E5,41,2,0)                                         PRES  091C
      CALL WRITMS (9,E6,41,3,0)                                         PRES  092C
      CALL WRITMS (9,E8,14,4,0)                                         PRES  093C
      CALL WRITMS (9,E5,41,5,0)                                         PRES  094C
C                                                                       PRES  095
      GO TO 50                                                          PRES  096
C                                                                       PRES  097
C  RETRIEVE PREVIOUSLY SAVED TABLE OF CONTENTS                          PRES  098
   40 IG9 = 1                                                           PRES  099
      CALL READMS (9,E1,8,IG9)                                          PRES  100C
C     READ (9'IG9) E1                                                   PRES  101I
      NTOT = E1(1)                                                      PRES  102
      ISUM = E1(2)                                                      PRES  103
      NEXT = E1(3)                                                      PRES  104
C   CHECK IF VALID RECORD                                               PRES  105
      IF (E1(4) .EQ. TITLE9(1)) GO TO 43                                PRES  106
      WRITE (TAPEOT,42)                                                 PRES  107
   42  FORMAT (1H0,47H***IFSAVE WAS INPUT = 1 IN PRES YET UNIT 9 HAS ,  PRES  108
     1 55HNEVER BEEN INITIALIZED WITH A IFSAVE = 0. PROGRAM STOP. )     PRES  109
      STOP                                                              PRES  110
   43 IG9 = 2                                                           PRES  111
      CALL READMS (9,E5,41,IG9)                                         PRES  112C
C     READ (9'IG9) E5                                                   PRES  113I
      DO 45 I=1,40                                                      PRES  114
   45  LCOM(I) = E5(I+1)                                                PRES  115
      IG9 = 5                                                           PRES  116
      CALL READMS(9,E5,41,IG9)                                          PRES  117C
C     READ (9'IG9) E5                                                   PRES  118I
      DO 47 I = 1,41                                                    PRES  119
   47 LCOM(40+I) = E5(I)                                                PRES  120
C                                                                       PRES  121
   50 DO 22 I=1,20                                                      PRES  122
       E4(I+1) = ALPHA(I)                                               PRES  123
   22  E4(I+21)= BETA (I)                                               PRES  124
      E4(1) = NAB                                                       PRES  125
      DO 24 I=1,5                                                       PRES  126
   24  E1(I+3) = TITLE9(I)                                              PRES  127
      DO 25 I=1,41                                                      PRES  128
   25  E6(I) = 0.0                                                      PRES  129
      DO 26 I=1,11                                                      PRES  130
   26  E7(I) = 0.0                                                      PRES  131
C                                                                       PRES  132
   55 IG4 = 2                                                           PRES  133
C     READ (4'IG4) E                                                    PRES  134I
      NSUB = 1                                                          PRES  135C
```

```
      NSREC4 = 2                                                        PRES  136C
      CALL STINDX (4,MIDX04,21)                                         PRES  137C
      CALL READMS (4,SIDX04,1001,NSUB)                                  PRES  138C
      CALL STINDX (4,SIDX04,1001)                                       PRES  139C
      NSUB4 = NSUB                                                      PRES  140C
      CALL READMS (4,E,25,NSREC4)                                       PRES  141C
      NEXTG = E(1)                                                      PRES  142
      NP = E(2)                                                         PRES  143
      NPMAX = E(3)                                                      PRES  144
      NREM = E(4)                                                       PRES  145
C                                                                       PRES  146
C  READ COMPONENT SELECTION CARD + INTERFERENCE FLAGS                   PRES  147
   60 READ (TAPEIN,70) IPANL,IPM,INT,ISHEF                              PRES  148
   70  FORMAT (10I2,3I1)                                                PRES  149
      NCOM = NCOM + 1                                                   PRES  150
C                                                                       PRES  151
C ********                                                              PRES  152
C  CHECK IF NEW PRESSURE DATA ARE TO BE READ                            PRES  153
      IF (IPM .GT. 1) GO TO 110                                         PRES  154
C  READ PRESSURE METHOD DATA (1 CARD OR NAB CARDS)                      PRES  155
      DO 100 I=1,NAB                                                    PRES  156
      IF (IPM.EQ.0 .OR. I.EQ.1) READ (TAPEIN,80) IMPACT,ISHAD,IPRINT,   PRES  157
     1    IPIN,ISAVE,PDATA                                              PRES  158
   80 FORMAT(2I2,3I1,3X,6F10.0)                                         PRES  159
      IMP(I) = IMPACT                                                   PRES  160
      ISH(I) = ISHAD                                                    PRES  161
      IPR(I) = IPRINT                                                   PRES  162
      ISA(I) = ISAVE                                                    PRES  163
      INP(I) = IPIN                                                     PRES  164
C                                                                       PRES  165
      DO 90 K=1,6                                                       PRES  166
   90  PDA(K,I) = PDATA(K)                                              PRES  167
  100 CONTINUE                                                          PRES  168
  110 CONTINUE                                                          PRES  169
C ********                                                              PRES  170
C                                                                       PRES  171
C                                                                       PRES  172
C  OBTAIN GEOMETRY DATA FROM SAVE UNIT FOR COMPLETE COMPONENT AND       PRES  173
C  PLACE IN GEOMETRY DATA WORKING ARRAYS                                PRES  174
      K = 0                                                             PRES  175
      NCS = 0                                                           PRES  176
      DO 130 I=1,10                                                     PRES  177
      IF (IPANL(I) .EQ. 0) GO TO 140                                    PRES  178
      NCS = NCS + 1                                                     PRES  179
      IG4 = IPANL(I)*5                                                  PRES  180
```

```
C       READ (4'IG4) EP                                                 PRES  181I
        NSUB=(IG4-1)/1000+1                                             PRES  182C
        NSREC4=IG4-(NSUB-1)*1000                                        PRES  183C
        IF (NSUB.EQ.NSUB4) GO TO 1                                      PRES  184C
        CALL STINDX (4,MIDX04,21)                                       PRES  185C
        CALL WRITMS (4,SIDX04,1001,NSUB4,1)                             PRES  186C
        CALL READMS (4,SIDX04,1001,NSUB)                                PRES  187C
        CALL STINDX (4,SIDX04,1001)                                     PRES  188C
        NSUB4=NSUB                                                      PRES  189C
      1 CALL READMS (4,EP,25,NSREC4)                                    PRES  190C
        ISTAT3 = EP(1)                                                  PRES  191
        COM(I) = EP(2)                                                  PRES  192
        ISTART = EP(3)                                                  PRES  193
        N = EP(4)                                                       PRES  194
        IORN = EP(5)                                                    PRES  195
        SYMFCT = EP(6)                                                  PRES  196
C                                                                       PRES  197
        K = K + 1                                                       PRES  198
        IG4S(K) = ISTART                                                PRES  199
        DO 120 II=2,N                                                   PRES  200
         K = K + 1                                                      PRES  201
        IF (K .GT. ISIZE) GO TO 150                                     PRES  202
        IG4S(K) = IG4S(K-1) + NREM                                      PRES  203
  120 CONTINUE                                                          PRES  204
  130 CONTINUE                                                          PRES  205
C                                                                       PRES  206
  140 LTOT = K                                                          PRES  207
      GO TO 170                                                         PRES  208
  150 WRITE (TAPEOT,160) ISIZE,ISIZE                                    PRES  209
  160  FORMAT (1H ,46H*** ERROR *** NUMBER OF ELEMENTS PER COMPONENT  ,  PRES  210
     1        13HCANNOT EXCEED ,I4,/1H ,14X,22HDIVIDE THE SHAPE INTO  ,  PRES  211
     2        47HMORE COMPONENTS WITH NO COMPONENT GREATER THAN  ,I4,   PRES  212
     3        9H ELEMENTS  )                                            PRES  213
      STOP                                                              PRES  214
  170 CONTINUE                                                          PRES  215
C                                                                       PRES  216
        ISRF = 0                                                        PRES  217
        DO 172 I = 1,NAB                                                PRES  218
        IF (INP(I) .GT. 0)  ISRF = 1                                    PRES  219
  172 CONTINUE                                                          PRES  220
        IF (ISRF .EQ. 0)  GO TO 175                                     PRES  221
C                                                                       PRES  222
C  ELEMENTS MUST BE WRITTEN ON TAPEE                                    PRES  223
C  CHECK IF ALREADY DONE FOR INTERFERRENCE FLOWS                        PRES  224
C                                                                       PRES  225
```

```
      DO 174 I = 1,LTOT                                                 PRES  226
      IG4 = IG4S(I)                                                     PRES  227
C     READ(4'IG4)  ELEM                                                 PRES  228I
      NSUB=(IG4-1)/1000+1                                               PRES  229C
      NSREC4=IG4-(NSUB-1)*1000                                          PRES  230C
      IF (NSUB.EQ.NSUB4) GO TO 3                                        PRES  231C
      CALL STINDX (4,MIDX04,21)                                         PRES  232C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                               PRES  233C
      CALL READMS (4,SIDX04,1001,NSUB)                                  PRES  234C
      CALL STINDX (4,SIDX04,1001)                                       PRES  235C
      NSUB4=NSUB                                                        PRES  236C
    3 CALL READMS (4,ELEM,25,NSREC4)                                    PRES  237C
      WRITE(TAPEE)  IG4,ELEM                                            PRES  238
  174 CONTINUE                                                          PRES  239
  175 CONTINUE                                                          PRES  240
C                                                                       PRES  241
C                                                                       PRES  242
C                                                                       PRES  243
C                                                                       PRES  244
C  CHECK FOR INPUT SURFACE PRESSURES                                    PRES  245
      IF (ISRF .EQ. 0)  GO TO 183                                       PRES  246
C  CALL PRE-PROCESSOR FOR SURFACE DATA                                  PRES  247
      CALL CPINPT(INP)                                                  PRES  248
  183 CONTINUE                                                          PRES  249
C                                                                       PRES  250
C                                                                       PRES  251
C                                                                       PRES  252
C  GEOMETRY DATA LOAD IS COMPLETE                                       PRES  253
C                                                                       PRES  254
      ISAS = 0                                                          PRES  255
      IF (IFSAVE .EQ. 2) GO TO 182                                      PRES  256
      NTOT = NTOT + 1                                                   PRES  257
      LCOM(NTCT) = NEXT                                                 PRES  258
      NEXT = NEXT + 2                                                   PRES  259
  182 CONTINUE                                                          PRES  260
C                                                                       PRES  261
C*********************************                                      PRES  262
C  START ALPHA-BETA LOOP (DO ALL ALPHA-BETAS FOR GIVEN COMPONENT)       PRES  263
      DO 190 J=1,NAB                                                    PRES  264
      IMPACT = IMP(J)                                                   PRES  265
      ISHAD = ISH(J)                                                    PRES  266
      IPRINT = IPR(J)                                                   PRES  267
      ISAVE = ISA(J)                                                    PRES  268
      IPIN = INP(J)                                                     PRES  269
      IF (ISAVE .EQ. 1) ISAS = 1                                        PRES  270
```

```
      DO 180 I=1,6                                                      PRES  271
  180 PDATA(I) = PDA(I,J)                                               PRES  272
C***GO TO FORCE PROGRAM                                                 PRES  273
      CALL FORCE (NEXTG,IG4S,NSUB4)                                     PRES  274
       IF (ERROR .NE. 0) GO TO 460                                      PRES  275
C                                                                       PRES  276
      IF (ISAVE.EQ.0) GO TO 187                                         PRES  277C
      DO 185 I=1,LTOT                                                   PRES  278C
      IG4 = IG4S(I) + 1                                                 PRES  279C
      E2(1) = SIDX16(I)                                                 PRES  280C
      NSUB = (IG4-1)/1000 + 1                                           PRES  281C
      NSREC4 = IG4 - (NSUB-1)*1000                                      PRES  282C
      IF (NSUB.EQ.NSUB4) GO TO 186                                      PRES  283C
      CALL STINDX (4,MIDX04,21)                                         PRES  284C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                               PRES  285C
      CALL READMS (4,SIDX04,1001,NSUB)                                  PRES  286C
      CALL STINDX (4,SIDX04,1001)                                       PRES  287C
      NSUB4 = NSUB                                                      PRES  288C
  186 CALL WRITMS (4,E2,25,NSREC4,1)                                    PRES  289C
  185 CONTINUE                                                          PRES  290C
C                                                                       PRES  291
      IF (SYMFCO.EQ.0) GO TO 187                                        PRES  292C
C                                                                       PRES  293
      DO 188 I=1,LTOT                                                   PRES  294C
      IG4 = IG4S(I) + 2                                                 PRES  295C
      E2(1) = SIDX21(I)                                                 PRES  296C
      NSUB = (IG4-1)/1000 + 1                                           PRES  297C
      NSREC4 = IG4 - (NSUB-1)*1000                                      PRES  298C
      IF (NSUB.EQ.NSUB4) GO TO 189                                      PRES  299C
      CALL STINDX (4,MIDX04,21)                                         PRES  300C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                               PRES  301C
      CALL READMS (4,SIDX04,1001,NSUB)                                  PRES  302C
      CALL STINDX (4,SIDX04,1001)                                       PRES  303C
      NSUB4 = NSUB                                                      PRES  304C
  189 CALL WRITMS (4,E2,25,NSREC4,1)                                    PRES  305C
  188 CONTINUE                                                          PRES  306C
C                                                                       PRES  307
  187 CONTINUE                                                          PRES  308C
C                                                                       PRES  309
C  SAVE FORCE DATA ON UNIT 9 IF REQUIRED                                PRES  310
      IF (IFSAVE .EQ. 2) GO TO 190                                      PRES  311
      CALL WRITMS (9,F,11,NEXT,0)                                       PRES  312C
C     WRITE (9'NEXT) F                                                  PRES  313I
      NEXT = NEXT + 1                                                   PRES  314
      CALL WRITMS (9,E7,11,NEXT,0)                                      PRES  315C
```

```
C       WRITE (9'NEXT) E7                                                PRES  316I
        NEXT = NEXT + 1                                                  PRES  317
  190 CONTINUE                                                           PRES  318
C**END OF ALPHA-BETA LOOP INTO FORCE                                     PRES  319
C*********************************                                      PRES  320
C                                                                        PRES  321
C                                                                        PRES  322
        IF (IFSAVE .EQ. 2) GO TO 450                                     PRES  323
C                                                                        PRES  324
C***OUTPUT DATA ******                                                   PRES  325
C  WRITE HEADER INFORMATION FOR FORCE OUTPUT PAGE                        PRES  326
        CALL HEADER                                                      PRES  327
        WRITE (TAPEOT,200) NCOM                                          PRES  328
  200  FORMAT (1H0,18HCOMPONENT NUMBER =, I3)                            PRES  329
        WRITE (TAPEOT,210) (COM(I),I=1,NCS)                              PRES  330
  210  FORMAT (1H0,12HPANEL ID =  , 20(A4,2X))                           PRES  331
        WRITE (6,230) MACH,V,RENO                                        PRES  332
  230 FORMAT (1H0,7H  MACH=F8.3,6H  VEL=F9.1,16H FT/SEC  RE/FT =E13.5 )  PRES  333
        IF (PSTAG .LT. 0.00001) WRITE (6,240) ALT                        PRES  334
  240 FORMAT (1H ,7H  ALT =F8.0 )                                        PRES  335
        IF (PSTAG .GT. 0.00001) WRITE (6,250) PSTAG,TSTAG                PRES  336
  250 FORMAT (1H ,16X7HP STAG=F7.1,16H ATMOS   T STAG=F7.1,6H DEG F )    PRES  337
        WRITE (TAPEOT,260) GTYPE                                         PRES  338
  260  FORMAT (1H ,2X,2A4)                                               PRES  339
        WRITE (6,270) SREF,SPAN,MAC,XCG,YCG,ZCG                          PRES  340
  270 FORMAT (1H0,9H  S REF =F9.2,8H  SPAN =F8.2,8H  MAC  =F8.2,/1H ,    PRES  341
     1  9H  X CG  =F9.2,8H  Y CG =F8.2,8H  Z CG =F8.2 )                  PRES  342
        WRITE (TAPEOT,280)                                               PRES  343
  280  FORMAT (1H0,10HFORCE DATA,/1H ,7H  ALPHA,4X,3HC D,7X,3HC L,7X,    PRES  344
     1 3HC A,7X,3HC Y,7X,3HC N,/1H ,7H  BETA ,4X,3HL/D,7X,3HC M,7X,       PRES  345
     2 4HC LL,6X,4HC LN)                                                 PRES  346
C                                                                        PRES  347
C  RETRIEVE OUTPUT DATA FROM UNIT 9 AND PRINT                            PRES  348
        DO 300 I=1,NAB                                                   PRES  349
         IG9 = LCCM(NTOT) + I*2                                          PRES  350
        CALL READMS (9,F,11,IG9)                                         PRES  351C
C       READ (9'IG9) F                                                   PRES  352I
C  WRITE OUTPUT FORCE DATA                                               PRES  353
        WRITE (TAPECT,290) F                                             PRES  354
  290  FORMAT (1H0,F7.2,5F10.5,/1H ,F7.2,4F10.5)                         PRES  355
  300 CONTINUE                                                           PRES  356
C**OUTPUT DATA COMPLETED FOR THIS COMPONENT                              PRES  357
C                                                                        PRES  358
C  SAVE COMPONENT INFORMATION IN TABLE OF CONTENTS                       PRES  359
        DO 191 I=2,11                                                    PRES  360
```

```
  191  E2(I) = IPANL(I-1)                                          PRES  361
       E2(1) = NTOT                                                PRES  362
       IG9 = LCCM(NTOT)                                            PRES  363
       CALL WRITMS (9,E2,11,IG9,0)                                 PRES  364C
C      WRITE (9'IG9) E2                                            PRES  365I
       IG9 = IG9 + 1                                               PRES  366
       CALL WRITMS (9,E4,41,IG9,0)                                 PRES  367C
C      WRITE (9'IG9) E4                                            PRES  368I
C                                                                  PRES  369
C  UPDATE MAIN TABLE OF CONTENTS                                   PRES  370
       E1(1) = NTOT                                                PRES  371
       E1(2) = ISUM                                                PRES  372
       E1(3) = NEXT                                                PRES  373
       IG9 = 1                                                     PRES  374
       CALL WRITMS (9,E1,8,IG9,1)                                  PRES  375C
C      WRITE (9'IG9) E1                                            PRES  376I
       DO 310 I=2,41                                               PRES  377
  310  E5(I) = LCOM(I-1)                                           PRES  378
       E5(1) = 0.0                                                 PRES  379
       IG9 = 2                                                     PRES  380
       CALL WRITMS (9,E5,41,IG9,1)                                 PRES  381C
C      WRITE (9'IG9) E5                                            PRES  382I
       IG9 = 3                                                     PRES  383
       CALL WRITMS (9,E6,41,IG9,1)                                 PRES  384C
C      WRITE (9'IG9) E6                                            PRES  385I
       IF (IFSAVE .EQ. 1)  GO TO 424                               PRES  386
       E8(1) = MACH                                                PRES  387
       E8(2) = V                                                   PRES  388
       E8(3) = RENO                                                PRES  389
       E8(4) = ALT                                                 PRES  390
       E8(5) = PSTAG                                               PRES  391
       E8(6) = TSTAG                                               PRES  392
       E8(7) = GTYPE(1)                                            PRES  393
       E8(8) = GTYPE(2)                                            PRES  394
       E8(9) = SREF                                                PRES  395
       E8(10) = SPAN                                               PRES  396
       E8(11) = MAC                                                PRES  397
       E8(12) = XCG                                                PRES  398
       E8(13) = YCG                                                PRES  399
       E8(14) = ZCG                                                PRES  400
       IG9 = 4                                                     PRES  401
       CALL WRITMS (9,E8,14,IG9,1)                                 PRES  402C
C      WRITE (9'IG9) E8                                            PRES  403I
C                                                                  PRES  404
  424 IG9 = 5                                                      PRES  405
```

```
      DO 425 I = 1,41                                                   PRES  406
  425 E5(I) = LCOM(40+I)                                                PRES  407
      CALL WRITMS (9,E5,41,IG9,1)                                       PRES  408C
C     WRITE (9'IG9) E5                                                  PRES  409I
C                                                                       PRES  410
      WRITE (TAPEOT,430) NTOT                                           PRES  411
  430  FORMAT (1H0,10X,35HDATA SAVED ON UNIT 9.  SET NUMBER =,I4)       PRES  412
C                                                                       PRES  413
C  CHECK IF LAST COMPONENT HAS BEEN COMPLETED                           PRES  414
  450 IF (NCOM .LT. NCOMP) GO TO 60                                     PRES  415
C                                                                       PRES  416
C    RESET COUNTERS IN GEOMETRY TABLE IF DATA WHERE SAVED               PRES  417
      IF (ISAS.EQ.0) GO TO 192                                          PRES  418
      E(1) = NEXTG                                                      PRES  419
      IG4 = 2                                                           PRES  420
C     WRITE (4'IG4) E                                                   PRES  421I
      NSUB = 1                                                          PRES  422C
      NSREC4 = 2                                                        PRES  423C
      CALL STINDX (4,MIDX04,21)                                         PRES  424C
      CALL WRITMS (4,SIDX04,1001,NSUB4,1)                               PRES  425C
      CALL READMS (4,SIDX04,1001,NSUB)                                  PRES  426C
      CALL STINDX (4,SIDX04,1001)                                       PRES  427C
      CALL WRITMS (4,E,25,NSREC4,1)                                     PRES  428C
      CALL STINDX (4,MIDX04,21)                                         PRES  429C
      CALL WRITMS (4,SIDX04,1001,NSUB,1)                                PRES  430C
  192 CONTINUE                                                          PRES  431
C                                                                       PRES  432
      GO TO 480                                                         PRES  433
C                                                                       PRES  434
C                                                                       PRES  435
C                                                                       PRES  436
C                                                                       PRES  437
  460 WRITE (TAPECT,470) ERROR,J                                        PRES  438
  470  FORMAT (1H ,10H** ERROR =,I2,26H ON RETURN FROM FORCE.  J= ,I1)  PRES  439
C 480 RETURN                                                            PRES  440I
  480 CONTINUE                                                          PRES  441C
      END                                                               PRES  442
```

```
      SUBROUTINE PUNCH (X1,Y1,Z1,NSTAT1,X2,Y2,Z2,NSTAT3,SECT,TYPE,
     1      LINE,SEQ,LAST,IPRINT,NREC)
C
C  THIS SUBROUTINE PREPARES VEHICLE GEOMETRY DATA IN THE PROPER FORM
C   FOR USE BY SDATA ROUTINE
C
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR
      COMMON /GFLAG/IOUT,ISTAT3,IORIN,COMPIN
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      DIMENSION  TITLE(15),SECT(1)
C
      INTEGER  PAGE,SEQ,ERROR,CASE
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
C
      NSTAT2 = NSTAT3
C
C  CHECK IF THIS IS THE LAST POINT OF THE ENTIRE VEHICLE
      IF (NSTAT3.EQ.3 .AND. LAST.EQ.1) NSTAT2 = 0
      IF (IPRINT .EQ. 0) GO TO 40
C
      IF  (LINE .LT. 50 )  GO TO 20
C
C  WRITE PAGE HEADER FOR STANDARD OUTPUT TAPE
      WRITE (TAPEOT,10) CASE,(TITLE(L),L=1,12),PAGE
   10 FORMAT (1H1,5X,36HANALYTICALLY GENERATED ELEMENT DATA ,/
     1   1H0,6H  CASE,3X,I2,17X,12A4,17X,5HPAGE ,I4,/
     2  1H0,5X1HX9X1HY9X1HZ4X1HS5X1HX9X1HY8X1HZ5X1HS10H CASE SECT,
     3  6X3HSEQ )
C
      PAGE = PAGE + 1
      LINE = 5
C
C  WRITE GEOMETRY CARDS ON STANDARD OUTPUT TAPE
   20 WRITE (TAPEOT,30) X1,Y1,Z1,NSTAT1,X2,Y2,Z2,NSTAT2,CASE,SECT,
     1       TYPE,SEQ
   30 FORMAT (1H0,3F10.4,I1,3F10.4,I1,2XI2,A4,1XI1,4HAERO,I4 )
C
      LINE = LINE + 2
C
C  WRITE GEOMETRY DATA ON GEOMETRY TAPE
   40 WRITE (IOUT,50)   X1,Y1,Z1,NSTAT1,X2,Y2,Z2,NSTAT2,CASE,SECT,
     1       TYPE,SEQ
   50 FORMAT(3F10.4,I1,3F10.4,I1,2XI2,A4,1XI1,4HAERO,I4 )
```

```
C
      NREC = NREC + 1                                          PUNC  046
      SEQ = SEQ + 1                                            PUNC  047
C                                                              PUNC  048
      RETURN                                                   PUNC  049
      END                                                      PUNC  050
                                                               PUNC  051
```

```
      SUBROUTINE  ROWFM1(N, M, XI, YI, FI, XKD, NX, MX, LS, IF1)
C
C  CALCULATES XKD ARRAY IN AUGMENTED FORM, ROW BY ROW.
C
C  LS = 0,  SURFACE SPLINE  X + Y VARIABLE
C  LS = 1,  LINEAR SPLINE  X CONSTANT
C  LS = 2,  LINEAR SPLINE  Y CONSTANT
C
C
      DIMENSION  XI(1), YI(1), XKD(1), FI(NX,MX)
      REWIND IF1
      N3 = N + 3
      N3M = N3 + M
C
      XKD(1) = 0.0
      XKD(2) = 0.0
      XKD(3) = 0.0
C
C  FIRST ROW
      DO 10 J = 4,N3
   10 XKD(J) = 1.0
C
C  AUGMENT BY RHS
      J = N3
      DO 20 I = 1,M
      J = J + 1
   20 XKD(J) = 0.0
C
      WRITE(IF1)   (XKD(J), J = 1,N3M)
C
C
C  CHECK FOR LINEAR SPLINE X CONSTANT
      IF (LS .NE. 1)  GO TO 25
C
C  LINEAR SPLINE X CONSTANT
      XKD(2) = 1.0
   25 JX = 0
      DO 30 J = 4,N3
      JX = JX + 1
   30 XKD(J) = XI(JX)
C
      WRITE(IF1)  (XKD(J), J = 1,N3M)
C
C  THIRD ROW
      XKD(2) = 0.0
```

```
C  CHECK FOR LINEAR SPLINE Y CONSTANT
   35 IF (LS .NE. 2)  GO TO 37
      XKD(3) = 1.0
   37 JX = 0
      DO 40 J = 4,N3
      JX = JX + 1
   40 XKD(J) = YI(JX)
C
      WRITE(IF1)  (XKD(J), J = 1,N3M)
C
C  FOURTH THROUGH N3 ROWS
      DO 100 I = 1,N
      XKD (1) = 1.0
      XKD(2) = XI(I)
      XKD(3) = YI(I)
      J3 = 3
      DO 80  J = 1,N
      J3 = J3 + 1
      AIJ = 0.0
      TR1 = 0.0
      TR2 = 0.0
      IF (I .EQ. J)  GO TO 70
      IF (XI(I) .EQ. XI (J))  GO TO 50
      TR1 = (XI(I) - XI(J))**2
   50 IF (YI(I) .EQ. YI(J))  GO TO 60
      TR2 = (YI(I) - YI(J))**2
   60 RAIJ = TR1 + TR2
      IF (RAIJ .EQ. 0)  GO TO 70
      AIJ = RAIJ*ALOG(RAIJ)
   70 XKD(J3) = AIJ
   80 CONTINUE
C
C  AUGMENT THE ROW
      DO 90 J = 1,M
      J3 = J3 + 1
   90 XKD(J3) = +FI(I,J)
C
      WRITE(IF1)  (XKD(J), J = 1,N3M)
C
  100 CONTINUE
C
C
      RETURN
      END
```

```
      SUBROUTINE  ROWFM2(N, NX, XI, YI, A, LS)
C
C  CALCULATES COEFFICIENT ARRAY A FOR SURFACE SPLINE
C  THIS ROUTINE USED WHEN N .LE. 74
C
C  LS = 0, SURFACE SPLINE  X AND Y VARIABLE
C  LS = 1, LINEAR SPLINE,  X CONSTANT
C  LS = 2, LINEAR SPLINE  Y CONSTANT
C
      DIMENSION XI(1), YI(1), A(NX,NX)
      N3 = N + 3
C
C  FIRST ROW
      I = 1
      A(I,1) = 0.0
      A(I,2) = 0.0
      A(I,3) = 0.0
      DO 10 J = 4,N3
   10 A(I,J) = 1.0
C
C  SECOND ROW
      I = 2
      A(I,2) = 0.0
      IF (LS .EQ. 1)  A(I,2) = 1.0
      A(I,1) = 0.0
      A(I,3) = 0.0
      JX = 0
      DO 30 J = 4,N3
      JX = JX + 1
   30 A(I,J) = XI(JX)
C
C  THIRD ROW
      I = 3
      A(I,3) = 0.0
      IF (LS .EQ. 2)  A(I,3) = 1.0
      A(I,1) = 0.0
      A(I,2) = 0.0
      JX = 0
      DO 40 J = 4,N3
      JX = JX + 1
   40 A(I,J) = YI(JX)
C
C  FOURTH THROUGH N3 ROWS
      DO 100 IX = 4,N3
```

```
      I = IX - 3                                                        ROW2  046
      A(IX,1) = 1.0                                                     ROW2  047
      A(IX,2) = XI(I)                                                   ROW2  048
      A(IX,3) = YI(I)                                                   ROW2  049
      J3 = 3                                                            ROW2  050
      DO 80 J = 1,N                                                     ROW2  051
      J3 = J3 + 1                                                       ROW2  052
      AIJ = 0.0                                                         ROW2  053
      TR1 = 0.0                                                         ROW2  054
      TR2 = 0.0                                                         ROW2  055
      IF (I .EQ. J)  GO TO 70                                           ROW2  056
      IF (XI(I) .EQ. XI(J))  GO TO 50                                   ROW2  057
      TR1 = (XI(I) - XI(J))**2                                          ROW2  058
   50 IF (YI(I) .EQ. YI(J))  GO TO 60                                   ROW2  059
      TR2 = (YI(I) - YI(J))**2                                          ROW2  060
   60 RAIJ = TR1 + TR2                                                  ROW2  061
      IF (RAIJ .EQ. 0.0)  GO TO 70                                      ROW2  062
      AIJ = RAIJ*ALOG(RAIJ)                                             ROW2  063
   70 A(IX,J3) = AIJ                                                    ROW2  064
   80 CONTINUE                                                          ROW2  065
  100 CONTINUE                                                          ROW2  066
C                                                                       ROW2  067
C                                                                       ROW2  068
      RETURN                                                            ROW2  069
      END                                                               ROW2  070
```

```
      SUBROUTINE  SAVE(IU, IT, N, N1, A1, N2, A2)
C
      DIMENSION  A1(N1), A2(N2)
C
      GO TO (10,20,30,40), IT
C
C  WRITE A1
   10 WRITE(IU) A1
      RETURN
C
C  WRITE N AND A1
   20 WRITE(IU)  N, A1
      RETURN
C
C  WRITE A1 AND A2
   30 WRITE(IU) A1, A2
      RETURN
C
C  WRITE N, A1, AND A2
   40 WRITE(IU) N, A1, A2
      RETURN
      END
```

```
      SUBROUTINE  SFNTR2
C
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      COMMON/SUFEN2/INORM,ISURF,IX,IY,IZ,UI,UL,VI,VL,CR,CI,CHX,N2,N3,
     1     XO,YO,ZO,AP11,AP12,AP13,AP21,AP22,AP23,AP31,AP32,AP33,
     2  FLOWC(7),XI(500),YI(500),B(503,7),XKF(503),AA(5999),
     A  XB(4), YB(4), ZB(4),
     3  NDSET,IABSET,IR,LURG(20),ISR(20),NS,ISFR,IVIS, IFTYP,IFLOW,
     4     BX(2,20), BY(2,20), NB
      COMMON/INDATA/IFUNID,ICORID,IPITCH,IBETA,BETA,IWRIT,
     1     IPRINT,ISVFMS,ISVSPL
      COMMON/SAVSPL/NEXPAN,NUMPAN,NUMCOR,NEXSPL,IPDP(72),IPSP(72),
     1     IPAN(72),IPANL(10)
      COMMON/SUPIDX/SIDX04(1001),SIDX16(1001),SIDX21(1001)
      COMMON /MASTER/ MIDX04(21),INDEXS(501),INDX10(2001),INDX15(2001),
     1  MIDX16(94),INDX20(2421),MIDX21(73)
C
      DIMENSION TITLE(15),E4(12),E4P(12),E1(25),LUSF(5),
     1     FLOW(503, 7),              XT(6), VT(6), LP(2)
      DIMENSION A(77,77)
      DIMENSION  SAVVAR(26), IG(100)
      DIMENSION ADUM(48), BDUM(1000), NDUM(607)
C
      INTEGER ERROR,        PAGE,CASE
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      INTEGER SIDX04,SIDX16,SIDX21
C
      EQUIVALENCE  (FLOW(1,1), B(1,1))
      EQUIVALENCE (A(1,1), AA(1))
      EQUIVALENCE (SAVVAR(1),INORM)
C
      DATA     NX, MX,      KD
     1     /                503, 7,    5999/
C
C  INITIALIZE CONSTANT ARRAY FLOWC
      DO 98 I = 1,7
   98 FLOWC(I) = 0.0
C
      IC = 0
      NB = 0
C
```

```
C  CYCLE ON SUB-REGIONS
      DO 600 II = 1,NS
      ISBR = ISR(II)
      IG10 = LORG(IR) + 4+ ISBR
      CALL READMS(10,E1,25,IG10)
C     READ(10'IG10)  E1
      LOFF = E1(1) + 0.01
      N1 = E1(6) + 0.01
C
      DO 500 J = 1,5
  500 LOSF(J) = E1(J+20) + 0.01
      IF (ISFR .EQ. 0)  GO TO 510
C
      IF (LOSF(ISFR) .LE. 0)  GO TO 600
C
C  READ IN SECONDARY FLOW POINTERS
      IG10 = LOSF(ISFR)
      CALL READMS(10,E1,25,IG10)
C     READ(10'IG10)  E1
      LOFF = E1(1) + 0.01
      N1 = E1(6) + 0.01
C
C  READ IN THE DATA
  510 ITAG10 = LOFF
      IBSF = 0
      IB = 0
      DO 310 I = 1,N1
      CALL READMS(10,E4,12,ITAG10)
C     READ(10'ITAG10)  E4
      ITAG10 = ITAG10 + 1
C
C  TRANSFORM DATA
  301 CONTINUE
      XX = E4(1) - X0
      YY = E4(2) - Y0
      ZZ = E4(3) - Z0
C
      XT(1) = XX*AP11 + YY*AP12 + ZZ*AP13
      XT(2) = XX*AP21 + YY*AP22 + ZZ*AP23
      XT(3) = XX*AP31 + YY*AP32 + ZZ*AP33
      XT(4) = XT(1)
      XT(5) = SQRT(XT(2)**2 + XT(3)**2)
      IF (XT(2) .EQ. 0.0)  GO TO 304
      XT(6) = ATAN2(XT(2),-XT(3))
      IF(XT(6).LT.0.0) XT(6) = 6.283185308 + XT(6)
```

```
      GO TO 305
  304 XT(6) = 0.0
      IF (XT(3) .GT. 0.0)  XT(6) = 3.141592654
  305 CONTINUE
      IF (IBSF .EQ. 1)  GO TO 523
C
C  NORMALIZE THE DATA
      IC = IC + 1
      IF (IC .LE. 500)  GO TO 303
      WRITE (TAPECT,302)
  302  FORMAT (1H0,48H*ROUTINE SFNTRP HAS ATTEMPTED TO LOAD MORE DATA ,
     1 62HPOINTS INTO INTERPOLATION ARRAYS THEN WE HAVE SPACE AVAILABLE.
     2 ,/1H ,52H*CALCULATIONS WILL CONTINUE WITH ONLY THE FIRST 100 ,
     3  7HPOINTS.  )
      IC = IC - 1
      GO TO 311
  303 YI(IC) = (XT(IY) - V1)/VL
      IF (ISURF .EQ. 1)  GO TO 306
      XI(IC) = (XT(IX) - U1)/UL
      GO TO 307
  306 CHX = CR + (CT-CR)*YI(IC)
      XLE = U1 + (UL-U1)*YI(IC)
      XI(IC) = (XT(IX) - XLE)/CHX
C
  307 CONTINUE
C  CHECK IF WITHIN SURFACE BOUNDARIES.
      IF (XI(IC) .LT. 0.0)  GO TO 308
      IF (XI(IC) .GT. 1.0)  GO TO 308
      IF (YI(IC) .LT. 0.0)  GO TO 308
      IF (YI(IC) .GT. 1.0)  GO TO 308
C
C  POINT WITHIN SURFACE BOUNDARIES.  SET UP FLOW BOUNDARIES --
C  -- INITIALIZE AS SURFACE BOUNDARIES.
      IF (IB .EQ. 1)  GO TO 535
      IB = 1
      NB = NB + 1
      BX(1,NB) = 0.0
      BY(1,NB) = YI(IC)
      IF (ISFR .EQ. 0)  GO TO 520
      BX(1,NB) = XI(IC)
C
  520 CONTINUE
      IF (IFTYP .EQ. 2)  GO TO 530
      IF (ISFR .EQ. 5)  GO TO 530
      IG10SF = LOSF(ISFR+1)
```

```
      IF (IG10SF .LE. 0)  GO TO 530
      CALL READMS(10,E1,25,IG10SF)
C     READ(10'IG10SF)  E1
      LOFF = E1(1) + 0.01
      CALL READMS(10,E4B,12,LOFF)
C     READ(10'LOFF)  E4B
      E4(1) = E4B(1)
      E4(2) = E4B(2)
      E4(3) = E4B(3)
      IBSF = 1
      GO TO 301
  523 IBSF = 0
      YIB = (XT(IY) - V1)/VL
      IF (ISURF .EQ. 1)  GO TO 526
      XIB = (XT(IX) - U1)/UL
      GO TO 527
  526 CHB = CR + (CT-CR)*YIB
      XLB = U1 + (UL-U1)*YIB
      XIB = (XT(IX) - XLB)/CHB
C
  527 CONTINUE
      BX(2,NB) = XIB
      BY(2,NB) = YIB
      GO TO 535
C
C
C
  530 CONTINUE
      BX(2,NB) = 1.0
      BY(2,NB) = BY(1,NB)
C  POINT WITHIN BOUNDARIES.  SET UP FLOW ARRAYS.
  535 CONTINUE
C
C  TRANSFORM FLOW DIRECTION COSINES
      VT(1) = E4(8)*AP11 + E4(9)*AP12 + E4(10)*AP13
      VT(2) = E4(8)*AP21 + E4(9)*AP22 + E4(10)*AP23
      VT(3) = E4(8)*AP31 + E4(9)*AP32 + E4(10)*AP33
      SINPH =  SIN(XT(6))
      COSPH =  COS(XT(6))
      VT(4) =  VT(1)
      VT(5) =  VT(2)*SINPH - VT(3)*COSPH
      VT(6) =  VT(2)*COSPH + VT(3)*SINPH
      FLOW(IC,1)   =  E4(7)
      FLOW(IC,2)   =  VT(IX)
      FLOW(IC,3)   =  VT(IY)
```

```
      FLOW(IC,4)   =  VT(IZ)                                            SFN2  181
      FLOW(IC,5)   =  E4(I1)                                            SFN2  182
      FLOW(IC,6)   =  E4(I2)                                            SFN2  183
      FLOW(IC,7)  = XT(IZ)                                              SFN2  184
      GO TO 310                                                         SFN2  185
C                                                                       SFN2  186
C  POINT NOT WITHIN BOUNDARIES.  RESET COUNTER AND GO TO NEXT POINT.    SFN2  187
  308 IC = IC - 1                                                       SFN2  188
C                                                                       SFN2  189
  310 CONTINUE                                                          SFN2  190
  600 CONTINUE                                                          SFN2  191
      IF (IC .GT. 2)  GO TO 311                                         SFN2  192
      WRITE(TAPEOT,1000)  IC                                            SFN2  193
 1000 FORMAT(1H1, 5HONLY ,I3,29H POINTS WERE FOUND IN SFNTR2 ,          SFN2  194
     1 56HTO BE WITHIN  SURFACE BOUNDARIES.  THIS IS INSUFFICIENT /     SFN2  195
     2 55H TO USE THE SURFACE SPLINE.  RUN CONTINUES, BUT USER IS,      SFN2  196
     3 38H CAUTIONED TO CHECK RESULTS CAREFULLY.)                       SFN2  197
      WRITE(TAPEOT,1005)  NDSET,IABSET,IR,ISR,ISFR                      SFN2  198
 1005 FORMAT(1H0,40HTHIS APPLIES TO THE FOLLOWING FLOW DATA /           SFN2  199
     1 12H DATA SET = ,I3, 10X17HALPHA-BETA SET = ,I3,                  SFN2  200
     2 10X14HFLOW REGION = ,I3/                                         SFN2  201
     3 15H SUB-REGIONS = , 2013/                                        SFN2  202
     4 18H SECONDARY FLOW = ,I3)                                        SFN2  203
      ERROR = 1                                                         SFN2  204
      RETURN                                                            SFN2  205
C                                                                       SFN2  206
C  ALL THE SURFACE DATA ARE IN PROPER ARRAYS.                           SFN2  207
C                                                                       SFN2  208
  311 N2 = IC                                                           SFN2  209
C                                                                       SFN2  210
C  CHECK BOUNDARY POINTS                                                SFN2  211
      IF (NB .GT. 1)  GO TO 700                                         SFN2  212
      NB = 2                                                            SFN2  213
      BX(1,2) = BX(1,1)                                                 SFN2  214
      BX(2,2) = BX(2,1)                                                 SFN2  215
      BY(1,2) = 1.0                                                     SFN2  216
      BY(2,2) = 1.0                                                     SFN2  217
C                                                                       SFN2  218
C  NORMALIZE TO FLOW BOUNDARIES (IN XI ONLY).                           SFN2  219
C  FIRST ARRANGE BOUNDARY DATA IN ORDER OF ASCENDING YI VALUES          SFN2  220
  700 IB = 0                                                            SFN2  221
  710 IB = IB + 1                                                       SFN2  222
      I1 = 1                                                            SFN2  223
  720 YY = BY(IB,I1)                                                    SFN2  224
      I2 = I1 + 1                                                       SFN2  225
```

```
      J = 0
      DO 730 I = I2,NB
      IF (YY .LT. BY(IB,I))  GO TO 730
      J = I
      YY = BY(IB,I)
  730 CONTINUE
C
      IF (J .EQ. 0)  GO TO 740
      BY(IB,J) = BY(IB,I1)
      BY(IB,I1) = YY
C
  740 I1 = I1 + 1
      IF (I1 .LT. NB)  GO TO 720
      IF (IB .EQ. 1)  GO TO 710
C
C  BOUNDARY DATA NOW IN ORDER.  PROCEED WITH NORMALIZATION.
      IF (IPRINT .EQ. 0)  GO TO 770
      WRITE(TAPEOT,750)  NB
  750 FORMAT(1H1,18HBOUNDARY  DATA,    , I3, 8H POINTS./
     1     1H0, 3X1HI, 7X2HX1, 12X2HY1, 12X2HX2, 12X2HY2//)
      WRITE(TAPEOT,760)  (I,BX(1,I),BY(1,I),BX(2,I),BY(2,I), I=1,NB)
  760 FORMAT(1H , I3, 4F14.6)
  770 CONTINUE
      IC = 0
      DO 900 I = 1,N2
      XX = XI(I)
      YY = YI(I)
      IB = 0
  800 IB = IB + 1
C
      DO 810 I1 = 1,NB
      J = I1
      IF (YY .LT. BY(IB,I1))  GO TO 820
  810 CONTINUE
C
  820 IF (J .EQ. 1)  J = 2
      DY = BY(IB,J) - BY(IB,J-1)
      IF (DY .EQ. 0.0)  GO TO 900
      DX = BX(IB,J) - BX(IB,J-1)
      BP(IB) = BX(IB,J-1) + (YY - BY(IB,J-1))*DX/DY
      GO TO (830,840), IB
  830 IF (XX .LT. BP(IB))  GO TO 900
      GO TO 800
  840 IF (XX .GT. BP(IB))  GO TO 900
C
```

```
C  POINT WITHIN FLOW BOUNDARIES
      IC = IC + 1
      XI(IC) = XX - BP(1)
      YI(IC) = YY
      IF (IVIS .EQ. 0)  GO TO 850
      XI(IC) = XI(IC)/(BX(2,1) - BX(1,1))
      GO TO 860
  850 XI(IC) = XI(IC)/(BP(2) - BP(1))
  860 FLOW(IC,1) = FLOW(I,1)
      FLOW(IC,2) = FLOW(I,2)
      FLOW(IC,3) = FLOW(I,3)
      FLOW(IC,4) = FLOW(I,4)
      FLOW(IC,5) = FLOW(I,5)
      FLOW(IC,6) = FLOW(I,6)
      FLOW(IC,7) = FLOW(I,7)
C
  900 CONTINUE
      IF (IC .GT. 2)  GO TO 910
      WRITE(TAPEOT,1010)  IC
 1010 FORMAT(1H1, 5HONLY ,I3, 29H POINTS WERE FOUND IN SFNTR2,
     151HTO BE WITHIN FLOW BOUNDARIES.  THIS IS INSUFFICIENT/
     2 55H TO USE THE SURFACE SPLINE.  RUN CONTINUES, BUT USER IS,
     3 38H CAUTIONED TO CHECK RESULTS CAREFULLY.)
      WRITE(TAPEOT,1005)  NDSET,IABSET,IR,ISR,ISFR
      ERROR = 1
      RETURN
  910 CONTINUE
C
C  CHECK IF SPLINE DATA TO BE SAVED
      IF (ISVSPL .EQ. 0)  GO TO 940
C
C  CHECK IF PANEL DEFINITIONS AND COORDS HAVE ALREADY BEEN SAVED.
      IPANID = IPANL(1)
      DO 920 I = 1,72
      IPNSEQ = I
      IF (IPAN(I) .EQ. IPANID)  GO TO 940
      IF (IPAN(I) .EQ. 0)  GO TO 930
  920 CONTINUE
C
C  PANEL ID COULD NOT BE FOUND AND ARRAY
C  IS FULL.  TURN OFF SAVE FLAG.
      ISVSPL = 0
      GO TO 940
C
C  NEW PANEL.  SAVE PANEL DEFINITIONS AND COORDS.
```

```
  930 IPAN(IPNSEQ) = IPANID                                          SFN2  316
      IPDP(IPNSEQ) = NEXPAN                                          SFN2  317
      NUMPAN = NUMPAN + 1                                            SFN2  318
      IG21 = NEXPAN                                                  SFN2  319
      NEXPAN = NEXPAN + 2                                            SFN2  320
      ICOP = NEXSPL                                                  SFN2  321
      NEXSPL = NEXSPL + (2*IC - 1)/36 + 1                            SFN2  322
C                                                                    SFN2  323
      REWIND 88                                                      SFN2  324
      WRITE (88) ICOP,IC,SAVVAR,XB,YB,ZB,                            SFN2  325
     1           ((BX(I,J),BY(I,J),I=1,2),J=1,2)                     SFN2  326
      REWIND 88                                                      SFN2  327
      READ (88) ADUM                                                 SFN2  328
C                                                                    SFN2  329
C     WRITE (21'IG21) ADUM                                           SFN2  330C
      NSUB21=(IG21-1)/1000+1                                         SFN2  331C
      NSRC21=IG21-(NSUB-1)*1000                                      SFN2  332C
      CALL STINDX (21,MIDX21,73)                                     SFN2  333C
      CALL READMS (21,SIDX21,1001,NSUB21)                            SFN2  334C
      CALL STINDX (21,SIDX21,1001)                                   SFN2  335C
      CALL WRITMS (21,ADUM,48,NSRC21,0)                              SFN2  336C
C                                                                    SFN2  337
      REWIND 88                                                      SFN2  338
      WRITE (88) (XI(I),YI(I),I=1,IC)                                SFN2  339
      REWIND 88                                                      SFN2  340
      J=IC*2                                                         SFN2  341
      READ (88) (BDUM(I), I=1,J)                                     SFN2  342
C                                                                    SFN2  343
C     WRITE (21'ICOP) (BDUM(I),I=1,J)                                SFN2  344C
      NSUB=(ICOP-1)/1000+1                                           SFN2  345C
      NSRC21=ICOP-(NSUB-1)*1000                                      SFN2  346C
      IF (NSUB.EQ.NSUB21) GO TO 6                                    SFN2  347C
      CALL STINDX (21,MIDX21,73)                                     SFN2  348C
      CALL WRITMS (21,SIDX21,1001,NSUB21,1)                          SFN2  349C
      CALL READMS (21,SIDX21,1001,NSUB)                              SFN2  350C
      CALL STINDX (21,SIDX21,1001)                                   SFN2  351C
      NSUB21=NSUB                                                    SFN2  352C
    6 CALL WRITMS (21,BDUM,J,NSRC21,0)                               SFN2  353C
      CALL STINDX (21,MIDX21,73)                                     SFN2  354C
      CALL WRITMS (21,SIDX21,1001,NSUB21,1)                          SFN2  355C
C                                                                    SFN2  356
  940 CONTINUE                                                       SFN2  357
C                                                                    SFN2  358
C  CYCLE THROUGH CP DATA TO ELIMINATE BAD POINTS                     SFN2  359
C  CP IS STORED IN FLOW(I,5)                                         SFN2  360
```

```
      NC = IC                                                           SFN2  361
      IC = 0                                                            SFN2  362
      DO 960 I = 1,NC                                                   SFN2  363
      IG(I) = 0                                                         SFN2  364
      XX = XI(I)                                                        SFN2  365
      YY = YI(I)                                                        SFN2  366
      IF (FLOW(I,5) .GT. 900.)  GO TO 950                               SFN2  367
      IC = IC + 1                                                       SFN2  368
      XI(IC) = XX                                                       SFN2  369
      YI(IC) = YY                                                       SFN2  370
      FLOW(IC,1) = FLOW(I,1)                                            SFN2  371
      FLOW(IC,2) = FLOW(I,2)                                            SFN2  372
      FLOW(IC,3) = FLOW(I,3)                                            SFN2  373
      FLOW(IC,4) = FLOW(I,4)                                            SFN2  374
      FLOW(IC,5) = FLOW(I,5)                                            SFN2  375
      FLOW(IC,6) = FLOW(I,6)                                            SFN2  376
      FLOW(IC,7) = FLOW(I,7)                                            SFN2  377
C                                                                       SFN2  378
      DO 300 II = 1,7                                                   SFN2  379
  300 FLOWC(II) = FLOWC(II) + FLOW(IC,II)                               SFN2  380
      GO TO 960                                                         SFN2  381
  950 CONTINUE                                                          SFN2  382
      IG(I) = 1                                                         SFN2  383
  960 CONTINUE                                                          SFN2  384
C                                                                       SFN2  385
C                                                                       SFN2  386
      N2 = IC                                                           SFN2  387
      N3 = N2 + 3                                                       SFN2  388
C  PRINT NORMALIZED ARRAYS                                              SFN2  389
      IF (IPRINT .EQ. 0)  GO TO 330                                     SFN2  390
      WRITE(TAPEOT,3060) NDSET, IABSET,IR                               SFN2  391
 3060 FORMAT(1H1,34HNORMALIZED SURFACE DATA.  NDSET = ,I2, 5X,          SFN2  392
     1  9HIABSET = ,I2, 5X, 9HREGION = ,I2/1H0, T4,1HI, T13,2HXI,       SFN2  393
     2  T27,2HYI, T39,4HMACH, T55,2HMX, T69,2HMY, T83,2HMZ,             SFN2  394
     3  T95,4HP/P1, T106,4HT/T1, T125,2HZI//)                           SFN2  395
      WRITE(TAPEOT,3070) (I,XI(I), YI(I), (FLOW(I,J),J=1,7), I=1,N2)    SFN2  396
 3070 FORMAT(1H , I3, 9F14.6)                                           SFN2  397
C                                                                       SFN2  398
C                                                                       SFN2  399
  330 CONTINUE                                                          SFN2  400
C                                                                       SFN2  401
      DO 350 J = 1,7                                                    SFN2  402
      FLOWC(J) = FLOWC(J)/N2                                            SFN2  403
      DO 340 I = 1,N2                                                   SFN2  404
  340 FLOW(I,J) = FLOW(I,J) - FLOWC(J)                                  SFN2  405
```

```
  350 CONTINUE
C
C
C  CALCULATE COEFFICIFNT ARRAY (SURFACE SPLINE).
      LS = 0
      IF (N2 .GT. 74)  GO TO 360
      N7 = 77
      CALL  ROWFM2(N2, N7, XI, YI, A, LS)
C
C  SHIFT B-ARRAY TO INCLUDE 3 ROWS OF ZEROES
      N1 = N2 + 1
      N4 = N2 + 4
      DO 355 J = 1,7
      DO 353 I = 1,N2
  353 B(N4-I,J) = B(N1-I,J)
      DO 354 I = 1,3
  354 B(I,J) = 0.0
  355 CONTINUE
C
      D = 1.0
      CALL  MIS1(A, N3, N7, B, MX, NX, MX, NERR, D)
      GO TO 370
  360 CONTINUE
      CALL ROWFM1 (N2,MX,XI,YI,FLOW,XKF,NX,MX,LS,TAPEB)
C
      CALL SOLVIT(AA,N3,MX,KD,TAPEB,TAPEC,TAPED,TAPEB,B,NX,MX,NERR)
  370 CONTINUE
      IF (NERR .NE. 0)  WRITE(TAPEOT,220)
  220 FORMAT(1H1, 16HMATRIX SINGULAR.)
C
C  COEFFICIENTS ARE IN ARRAY  B(I,J)
      IF (IPRINT .EQ. 0)  GO TO 400
      WRITE(TAPEOT,3080)
 3080 FORMAT(1H1,42HSURFACE INTERPOLATION COEFFICIENTS  B(I,J)//)
      WRITE(TAPEOT,3090)  (I, (B(I,J), J=1,7), I=1,N3)
 3090 FORMAT(1H , I5, 7F15.5)
      WRITE(TAPEOT,3100)  FLOWC
 3100 FORMAT(1H0, 5X, 7F15.5)
      WRITE(TAPEOT,3110) INORM,ISURF,IX,IY,IZ,UI,UL,VI,VL,CR,CT,CHX,N2,
     1    N3
 3110 FORMAT(1H1,23HINORM,ISURF,IX,IY,IZ = , 5I4/
     1        1H0,24HUI,UL,VI,VL,CR,CT,CHX = , 7F15.5/
     2        1H0, 5HN2 = ,I4,5X,5HN3 = ,I4)
      WRITE(TAPEOT,3120) XO,YO,ZO
 3120 FORMAT(1H0, 5HXO = , F15.5,5X5HYO = , F15.5,5X5HZO = ,F15.5)
```

```
      WRITE(TAPEOT,3130) AP11,AP12,AP13,AP21,AP22,AP23,AP31,AP32,AP33
 3130 FORMAT(1H0,42HTRANSFORMATION MATRIX,  APIJ, I=1,3, J=1,3/
     1       (1H0, 3F20.6))
C
  400 CONTINUE
C
C
      IF (ISVSPL .EQ. 0)  GO TO 450
C
C   SAVE SPLINE DATA
      IPSP(IPNSEQ) = NEXSPL
      IG21 = NEXSPL
C
      REWIND 88
      WRITE (88) NC,N3,IPNSEQ,FLOWC(5),(IG(I),I=1,NC),
     1           (B(I,5),I=1,N3)
      REWIND 88
      J=4+NC+N3
      READ (88) (NDUM(I), I=1,J)
C
C     WRITE (21'IG21) (NDUM(I),I=1,J)
      NSUB21=(IG21-1)/1000+1
      NSRC21=IG21-(NSUB21-1)*1000
      CALL STINDX (21,MIDX21,73)
      CALL READMS (21,SIDX21,1001,NSUB21)
      CALL STINDX (21,SIDX21,1001)
      CALL WRITMS (21,NDUM,J,NSRC21,0)
      CALL STINDX (21,MIDX21,73)
      CALL WRITMS (21,SIDX21,1001,NSUB21,1)
      NEXSPL = NEXSPL + (NC + N3 + 4 - 1)/36 + 1
C
C
  450 CONTINUE
C
C
      RETURN
      END
```

```
      OVERLAY (MARK4,2,7)                                               SHEA  001C
      PROGRAM SHEAR                                                     SHEA  002C
      SUBROUTINE SHEAR                                                  SHEA  003I
C                                                                       SHEA  004
C                                                                       SHEA  005
C     THIS SUBROUTINE WRITTEN SPECIFICALLY FOR THE YF-12 CONTRACT       SHEA  006
C     THE INDICES INCORPORATE THE FIVE KINDS OF SHEARS                  SHEA  007
C           1) FUSELAGE - - - - - 3 STATIONS                            SHEA  008
C           2) WING - - - - - - - 3 STATIONS (ELEVON HM)                SHEA  009
C           3) NACELLE VERTICAL - 1 STATION                             SHEA  010
C           4) NACELLE VENTRAL -  1 STATION                             SHEA  011
C           5) FUSELAGE VENTRAL - 1 STATION  ( BETA CASE )              SHEA  012
C                                                                       SHEA  013
C                                                                       SHEA  014
      COMMON /EXEC/CASE,TITLE,IPGE,IERR                                 SHEA  015
      COMMON/INDATA/IRUN,ICOR,IPLTCP,IAB,BETAIN,IWRIT,                  SHEA  016
     2    IPRINT,ISVFMS,ISVSPL                                          SHEA  017
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   SHEA  018
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                        SHEA  019
      COMMON/SUBIDX/SIDX04(1001),SIDX16(1001),SIDX21(1001)              SHEA  020C
      COMMON /MASTER/ MIDX04(21),INDEX9(501),INDX10(2001),INDX15(2001), SHEA  021C
     1  MIDX16(94),INDX20(2421),MIDX21(73)                              SHEA  022C
      DIMENSION XSHR(3,5), YSHR(3,5), ZSHR(3,5), NSHST(5),NSBTP(5,2),   SHEA  023
     1    NOFP(15,5,2), SSHR(3,5), CSHR(3,5), BSHR(3,5), CV(3,5,2),     SHEA  024
     2    CB(3,5,2), CT(3,5,2), HME(4), CHE(4), SEH(4), CEH(4)          SHEA  025
      DIMENSION  E9(41), EP(25), ELEM(25), E0(25), E1(25), ZERO(94)     SHEA  026
      DIMENSION TITLE(15),LSUM(40),E(25)                                SHEA  027
      DIMENSION IRUNID(216),NCORR(216),ICURID(108),IDAPT(108),F(11)     SHEA  028
      DIMENSION NDUM(2), MDUM(108)                                      SHEA  029
      INTEGER TAPEIN                                                    SHEA  030
      INTEGER SIDX04,SIDX16,SIDX21                                      SHEA  031C
      EQUIVALENCE  ( EP(25), ELEM(25), E0(25), E1(25))                  SHEA  032
      EQUIVALENCE (ZERO(1),CV(1)), (ZERO(31),CB(1)), (ZERO(61),CT(1)),  SHEA  033
     1            (ZERO(91),CHE(1))                                     SHEA  034
      EQUIVALENCE (ZERO(1),MDUM(15),CV(1)), (ZERO(31),CB(1)),           SHEA  035
     1            (ZERO(61),CT(1)), (ZERO(91),CHE(1)),                  SHEA  036
     2            (NEXRUN,NDUM(1)), (NEXDAT,NDUM(2)), (IRSQ,MDUM(1)),   SHEA  037
     3            (EMACH,MDUM(3)),(F(1),MDUM(4))                        SHEA  038
      DATA ZERO/94*0.0/,HME/4*0.0/                                      SHEA  039
      DATA XSHR/-420.0 , -587.0, -730.0, -970.0,-1018.0,-1130.0,-1142.0, SHEA  040
     1         0.0, 0.0,-1099.0, 0.0, 0.0,-1183.5, 0.0, 0.0 /,          SHEA  041
     2     YSHR/ 3*0.0, 35.0, 111.0, 215.0, 155.64, 2*0.0, 177.18,5*0.0/ SHEA  042
     3    ,ZSHR/ 3*100.0, 94.6, 95.6, 98.0, 151.84, 2*0.0, 74.86,2*0.0, SHEA  043
     4         68.0, 2*0.0 /                                            SHEA  044
      DATA SSHR/ 175.0, 341.0, 484.0, 720.0, 420.0, 137.0, 70.0, 2*0.0, SHEA  045
```

```
     1          22.0, 2*0.0, 44.0, 2*0.0/,                                  SHEA  046
     2      CSHR/ 330.0, 497.0, 640.0, 433.0, 332.0, 189.0, 140.0,2*0.0,    SHEA  047
     3          119.0, 2*0.0, 125.0, 2*0.0/,                                SHEA  048
     4      BSHR/ 330.0, 497.0, 640.0,319.0, 243.0, 139.0, 75.0, 2*0.0,     SHEA  049
     5          30.4, 2*0.0, 58.0, 2*0.0/                                   SHEA  050
      DATA NOFP/1,2,3,4,5,10*0,   6,7,8,9,10,11,12,13,19,20,21,22,3*0,      SHEA  051
     1          14,15,13*0,   17,18,13*0,   16,14*0,   35,36,37,38,39,      SHEA  052
     2          10*0,   40,41,42,43,44,45,46,47,53,54,55,56,3*0,            SHEA  053
     3          48,49,13*0,   51,52,13*0,   16,50,13*0/                     SHEA  054
      DATA NSHST/ 3,3, 1, 1, 1  /,  NSBTP/ 5,12,2*2,1,5,12,2*2,2/           SHEA  055
      DATA SEH/ 39.0,53.0,39.0,53.0/, CEH/ 52.0,71.0,52.0,71.0/             SHEA  056
      IERR = 0                                                              SHEA  057
C                                                                           SHEA  058
C     IWRIT INCONSISTENT WITH FLAGS IN HABP SO INVERT                       SHEA  059
C       (IE) IWRIT = 0 (NO PRINT) & IWRIT = 4 (ALL PRINT)                   SHEA  060
      IWRIT = 4 - IWRIT + (IWRIT/5) * 5                                     SHEA  061
C                                                                           SHEA  062
C                                                                           SHEA  063
C   IREAD-CARD INPUT OPTION FLAG                                            SHEA  064
C            IREAD = 0 (AUTO INPUT) & IREAD = 1 (CARD INPUT)                SHEA  065
C                                                                           SHEA  066
C                                                                           SHEA  067
      IREAD = IWRIT / 5                                                     SHEA  068
C                                                                           SHEA  069
C                                                                           SHEA  070
C                                                                           SHEA  071
      IF (IREAD.GT.0) GO TO 75                                              SHEA  072
C                                                                           SHEA  073
      IF (IWRIT.GT.0) GO TO 80                                              SHEA  074
C                                                                           SHEA  075
      GO TO 76                                                              SHEA  076
C                                                                           SHEA  077
   75 CONTINUE                                                              SHEA  078
C                                                                           SHEA  079
      DO 15 I=1,5                                                           SHEA  080
C                                                                           SHEA  061
C   READ COMPONENT DESCRIPTION CARD                                         SHEA  082
      READ (TAPEIN,5200) NSHST(I),NSBTP(I,1),NSBTP(I,2)                     SHEA  083
 5200 FORMAT (4012)                                                         SHEA  084
      NSH = NSHST(I)                                                        SHEA  085
C                                                                           SHEA  086
      DO 10 J=1,NSH                                                         SHEA  087
C                                                                           SHEA  088
C   READ SHEAR STATION DESCRIPTION CARDS                                    SHEA  089
      READ (TAPEIN,5400) XSHR(J,I),YSHR(J,I),ZSHR(J,I),                     SHEA  090
```

```
     1                    SSHR(J,I),CSHR(J,I),BSHR(J,I)
 5400 FORMAT (6F10.0)
C
   10 CONTINUE
C
      I2 = NSBTP(I,1)
C
C    READ FIRST PANEL NUMBER DEFINITICN CARD
      READ (TAPEIN,5200) (NOFP(K,I,1),K=1,12)
C
      I2 = NSBTP(I,2)
C
C    READ SECOND PANEL NUMBER DEFINITICN CARD
      READ (TAPEIN,5200) (NOFP(K,I,2),K=1,12)
C
   15 CONTINUE
C
      DO 20 I=1,3,2
C
C    READ ELEVON DEFINITION CARDS
      READ (TAPEIN,5500) SEH(I),CEH(I),SEH(I+1),CEH(I+1)
 5500 FORMAT (4F10.0)
C
   20 CONTINUE
C
      CALL HNGFT (IREAD)
C
      IF (IWRIT.GT.0) GO TO 80
   76 CONTINUE
C
C     WRITE GEOMETRY, NO.SHEAR STA,NO OF PANELS, PANEL NOS, & PARAMETERS
C
      WRITE (6,6101) XSHR
 6101 FORMAT (1H , 17H-X- SHEAR STATICN, 48X, 5E13.6/1X, 10E13.6)
      WRITE (6,6102) YSHR
 6102 FORMAT (1H , 17H-Y- SHEAR STATICN, 48X, 5E13.6/1X, 10E13.6)
      WRITE (6,6103) ZSHR
 6103 FORMAT (1H , 17H-Z- SHEAR STATICN, 48X, 5E13.6/1X, 10E13.6)
      WRITE (6,6201) NSHST
 6201 FORMAT (1H , 18HNO. OF SHEAR STA. , 7X, 5I5)
      WRITE (6,6202) NSBTP
 6202 FORMAT (1H , 19HNO. OF SHEAR PANELS, 6X, 10I5)
      WRITE (6,6203) NOFP
 6203 FORMAT (1H , 14HPANEL NUMBERS , 10(/1X, 20I5))
      WRITE (6,6104) SSHR
```

```
 6104 FORMAT (1H , 26HNONDIMENSIONALIZING AREAS , 39X,5E13.6/1X,10E13.6)   SHEA  136
      WRITE (6,6105) CSHR                                                  SHEA  137
 6105 FORMAT (1H , 26HNONDIMENSIONALIZING CHORDS, 39X,5E13.6/1X,10E13.6)   SHEA  138
      WRITE (6,6106) BSHR                                                  SHEA  139
 6106 FORMAT (1H , 25HNONDIMENSIONALIZING SPANS,  40X,5E13.6/1X,10E13.6)   SHEA  140
      WRITE (6,6107) SEH, CEH                                              SHEA  141
 6107 FORMAT (1H , 13HELEVON AREAS , 4E13.6, 4X, 9HCHORDS    , 4E13.6)     SHEA  142
C                                                                          SHEA  143
   80 CONTINUE                                                             SHEA  144
C                                                                          SHEA  145
      IG4 = 2                                                              SHEA  146
C     READ (4'IG4) E                                                       SHEA  147I
      CALL READMS (4,E,25,IG4)                                             SHEA  148C
      CALL STINDX (4,SIDX04,1001)                                          SHEA  149C
      CALL READMS (4,SIDX04,1001,NSUB4)                                    SHEA  150C
      CALL STINDX (4,MIDX04,21)                                            SHEA  151C
      NSUB4 = 1                                                            SHEA  152C
      NEHMF = E(5)                                                         SHEA  153
      IF (NEHMF.EQ.0) CALL HNGFT(IREAD)                                    SHEA  154
C                                                                          SHEA  155
C     CONVERT TO MODEL SCALE                                               SHEA  156
C                                                                          SHEA  157
      DO 50 J=1,5                                                          SHEA  158
      DO 60 I=1,3                                                          SHEA  159
      XSHR(I,J) = XSHR(I,J)/12.0                                           SHEA  160
      YSHR(I,J) = YSHR(I,J)/12.0                                           SHEA  161
      ZSHR(I,J) = ZSHR(I,J)/12.0                                           SHEA  162
      CSHR(I,J) = CSHR(I,J)/12.0                                           SHEA  163
      BSHR(I,J) = BSHR(I,J)/12.0                                           SHEA  164
   60 CONTINUE                                                             SHEA  165
   50 CONTINUE                                                             SHEA  166
      DO 70 I=1,4                                                          SHEA  167
      CEH(I) = CEH(I)/12.0                                                 SHEA  168
   70 CONTINUE                                                             SHEA  169
      IF (IWRIT.GT.1) GO TO 85                                             SHEA  170
      WRITE (6,6101) XSHR                                                  SHEA  171
      WRITE (6,6102) YSHR                                                  SHEA  172
      WRITE (6,6103) ZSHR                                                  SHEA  173
      WRITE (6,6104) SSHR                                                  SHEA  174
      WRITE (6,6105) CSHR                                                  SHEA  175
      WRITE (6,6106) BSHR                                                  SHEA  176
      WRITE (6,6107) SEH, CEH                                              SHEA  177
   85 CONTINUE                                                             SHEA  178
C                                                                          SHEA  179
C     READ NTOT, ISUM, NEXT(UNIT 9)                                        SHEA  180
```

```
C
      IG9 = 1
C     READ (9'IG9) E9
      CALL READMS (9,E9,41,IG9)
      IF (IWRIT.GT.2) GO TO 90
      WRITE (6,6110)
 6110 FORMAT (1H , 12H-E9- UNIT 9 )
      WRITE (6,6100) E9
 6100 FORMAT ( 1X, 10E13.6)
   90 CONTINUE
      NTOT = E9(1)
      IF (( NTOT.EQ.22 ) .OR. ( NTOT .EQ.44)) GO TO 25
      WRITE (6,6300) NTOT
 6300 FORMAT (1H , 11H   NTOT   =, I4, 2X, 12HNOT 22 OR 44)
      GO TO 1000
   25 CONTINUE
C
C     ALPHA OR BETA CASE
      KAB = IAB + 1
      KA = 1
      MP = 1
      IF (IWRIT.LE.1) WRITE (6,6210) KA, KAB
 6210 FORMAT (1H ,  5HKA = , I4, 8H, KAB = , I4)
C
C     CHECK ALPHA/BETA AND NUMBER OF PANELS
C
      IABCK = 22*KAB
      IF (IABCK.EQ.NTOT) GO TO 30
      WRITE (6,6330) IAB, NTOT
 6330 FORMAT (1H , 29HINPUT ALPHA/BETA CODE (IAB) =, I4, 69H IS INCONSIS
     1TENT WITH THE TOTAL NUMBER OF PANELS FROM UNIT 9 (NTOT) =, I4)
   30 CONTINUE
C
C     DETERMINE ELEVON DEFLECTICN CCDE
C
      KL = 0
      IF (IRUN.GE.27.AND.IRUN.LE.164) KL=2
      IF (IRUN.GE.41.AND.IRUN.LE.125) KL=0
      IF (IRUN.GE.67.AND.IRUN.LE. 76) KL=2
      IF (IRUN.GE.77.AND.IRUN.LE. 86) KL=3
      IF (IRUN.GE.87.AND.IRUN.LE.101) KL=1
C
C     CALCULATF ELEVON PANEL NUMBER USING DEFLECTION CODE
      DO 95 J = 1,KAB
      DO 96 I = 9,12
```

```
      NOFP(I,2,J) = I +10 + 34*(J - 1) + KL*4                           SHEA  226
   96 CONTINUE                                                          SHEA  227
   95 CONTINUE                                                          SHEA  228
      IF (IWRIT.LE.3) WRITE (6,6220) ((NOFP(I,2,J),I= 9,12),J=1,KAB)    SHEA  229
 6220 FORMAT (1H , 22HELEVON PANEL NOS USED , 8I5)                      SHEA  230
C                                                                       SHEA  231
C     CALCULATE SHEARS,BENDING MOMENTS, TORQUES, & HINGE MOMENTS        SHEA  232
C                                                                       SHEA  233
C         SET ELEVON INDEX                                              SHEA  234
      I7 = 1                                                            SHEA  235
C                                                                       SHEA  236
C     ESTABLISH SIN AND COS OF WING INCIDENCE                           SHEA  237
      WSIN = SIN(-0.020944)                                             SHEA  238
      WCOS = COS(-0.020944)                                             SHEA  239
C                                                                       SHEA  240
C         INITIATE COUNTER                                              SHEA  241
      I = 0                                                             SHEA  242
C                                                                       SHEA  243
  100 I = I + 1                                                         SHEA  244
C                                                                       SHEA  245
C     ESTABLISH SIN AND COS OF VERTICAL CANT                            SHEA  246
      IF (I.LT.3) GO TO 102                                             SHEA  247
      ANV =-0.2007129*(5-I)*MP                                          SHEA  248
      IF (I.EQ.3) ANV =-0.2617994*MP                                    SHEA  249
      VSN = SIN(ANV)                                                    SHEA  250
      VCS = COS(ANV)                                                    SHEA  251
  102 CONTINUE                                                          SHEA  252
C     NUMBER OF SHEAR STATIONS                                          SHEA  253
      I2 = NSHST(I)                                                     SHEA  254
C     NUMBER OF PANELS INVOLVED                                         SHEA  255
      I3 = NSBTP(I,KA)                                                  SHEA  256
      IF (IWRIT.LE.1) WRITE (6,6230) I2, I3, VSN, VCS                   SHEA  257
 6230 FORMAT (1H , 22HNO. OF SHEAR STA(I2) =, I4, 20H, NO OF PANELS(I3) SHEA  258
     1=, I4, 19H, VERT. CANT SIN = , F10.7, 8H, COS = , F10.7)          SHEA  259
C                                                                       SHEA  260
C     CONSIDER EACH PANEL                                               SHEA  261
      DO 300 J=1,I3                                                     SHEA  262
C                                                                       SHEA  263
C     READ PANEL TABLE ARRAY                                            SHEA  264
      KL = NOFP(J,I,KA)                                                 SHEA  265
      IG4 = KL*5                                                        SHEA  266
C     READ (4'IG4) EP                                                   SHEA  267I
      NSUB=(IG4-1)/1000+1                                               SHEA  268C
      NSREC4=IG4-(NSUB-1)*1000                                          SHEA  269C
      IF (NSUB.EQ.NSUB4) GO TO 1                                        SHEA  270C
```

```
      CALL STINDX (4,MIDX04,21)                                         SHEA  271C
      CALL READMS (4,SIDX04,1001,NSUB)                                  SHEA  272C
      CALL STINDX (4,SIDX04,1001)                                       SHEA  273C
      NSUB4=NSUB                                                        SHEA  274C
    1 CALL READMS (4,EP,25,NSREC4)                                      SHEA  275C
      IF (IWRIT.GT.2) GO TO 305                                         SHEA  276
      WRITE (6,6120)                                                    SHEA  277
 6120 FORMAT (1H , 30H -EP- UNIT 4 PANEL TABLE ARRAY)                   SHEA  278
      WRITE (6,6100) EP                                                 SHEA  279
  305 CONTINUE                                                          SHEA  280
C                                                                       SHEA  281
C     CHECK PANEL NO.                                                   SHEA  282
      IJ2 = EP(1)                                                       SHEA  283
      IF (IJ2.EQ.KL) GO TO 310                                          SHEA  284
      WRITE (6,6600) KL, IJ2                                            SHEA  285
 6600 FORMAT (1H , 22HHARK, INPUT PANEL NO =, I4, 2X, 25HBUT, THE TABLE SHEA  286
     1ARRAY NO =, I4)                                                   SHEA  287
      GO TO 1000                                                        SHEA  288
  310 CONTINUE                                                          SHEA  289
C                                                                       SHEA  290
C     NUMBER OF ELEMENTS IN PANEL                                       SHEA  291
      NOEL=EP(4)                                                        SHEA  292
C                                                                       SHEA  293
C     SET ISTART TO INCREMENT                                           SHEA  294
      ISTART = IFIX(EP(3)) - 3                                          SHEA  295
C                                                                       SHEA  296
C     CONSIDER EACH ELEMENT                                             SHEA  297
C                                                                       SHEA  298
      DO 400 K=1,NOEL                                                   SHEA  299
C                                                                       SHEA  300
C     READ ELEMENT DATA ARRAY                                           SHEA  301
      ISTART=ISTART+3                                                   SHEA  302
C     READ (4'ISTART) ELEM                                              SHEA  303I
      NSUB=(ISTART-1)/1000+1                                            SHEA  304C
      NSREC4=ISTART-(NSUB-1)*1000                                       SHEA  305C
      IF (NSUB.EQ.NSUB4) GO TO 2                                        SHEA  306C
      CALL STINDX (4,MIDX04,21)                                         SHEA  307C
      CALL READMS (4,SIDX04,1001,NSUB)                                  SHEA  308C
      CALL STINDX (4,SIDX04,1001)                                       SHEA  309C
      NSUB4=NSUB                                                        SHEA  310C
    2 CALL READMS (4,ELEM,25,NSREC4)                                    SHEA  311C
      IF (IWRIT.GT.2) GO TO 320                                         SHEA  312
      WRITE (6,6130)                                                    SHEA  313
 6130 FORMAT (1H , 32H-ELEM- UNIT 4 ELEMENT DATA ARRAY)                 SHEA  314
      WRITE (6,6100) ELEM                                               SHEA  315
```

```
  320 CONTINUE                                                          SHEA  316
C                                                                       SHEA  317
C     CHECK ELEMENT NUMBER                                              SHEA  318
      KL = ELEM(1)                                                      SHEA  319
      IF (K.EQ.KL) GO TO 410                                            SHEA  320
      WRITE (6,6700) KL, K                                              SHEA  321
 6700 FORMAT (1H , 31HHARK, ELEMENT ARRAY (ELEM) NO =, I4, 2X, 22HBUT, T SHEA  322
     1HE COUNTER NO IS, I4)                                             SHEA  323
      GO TO 1000                                                        SHEA  324
  410 CONTINUE                                                          SHEA  325
      XCENT=ELEM(7)                                                     SHEA  326
      YCENT=ELEM(8)                                                     SHEA  327
      ZCENT=ELEM(9)                                                     SHEA  328
      HMF = ELEM(23)                                                    SHEA  329
C                                                                       SHEA  330
C     READ POINTERS FOR SURFACE PROPERTY DATA                           SHEA  331
      IG4=ISTART+1                                                      SHEA  332
C     READ (4'IG4) E0                                                   SHEA  333I
      NSUB=(IG4-1)/1000+1                                               SHEA  334C
      NSREC4=IG4-(NSUB-1)*1000                                          SHEA  335C
      IF (NSUB.EQ.NSUB4) GO TO 3                                        SHEA  336C
      CALL STINDX (4,MIDX04,21)                                         SHEA  337C
      CALL READMS (4,SIDX04,1001,NSUB)                                  SHEA  338C
      CALL STINDX (4,SIDX04,1001)                                       SHEA  339C
      NSUB4=NSUB                                                        SHEA  340C
    3 CALL READMS (4,E0,25,NSREC4)                                      SHEA  341C
      IF (IWRIT.GT.1) GO TO 415                                         SHEA  342
      WRITE (6,6140)                                                    SHEA  343
 6140 FORMAT (1H , 47H -E0- UNIT 4 POINTERS FOR SURFACE PROPERTY DATA)  SHEA  344
      WRITE (6,6100) E0                                                 SHEA  345
  415 CONTINUE                                                          SHEA  346
C                                                                       SHEA  347
C      READ SURFACE PROPERTY DATA                                       SHEA  348
      IG4=E0(1)                                                         SHEA  349
C     READ (4'IG4) E1                                                   SHEA  350I
      NSUB=(IG4-1)/1000+1                                               SHEA  351C
      NSREC4=IG4-(NSUB-1)*1000                                          SHEA  352C
      IF (NSUB.EQ.NSUB4) GO TO 4                                        SHEA  353C
      CALL STINDX (4,MIDX04,21)                                         SHEA  354C
      CALL READMS (4,SIDX04,1001,NSUB)                                  SHEA  355C
      CALL STINDX (4,SIDX04,1001)                                       SHEA  356C
      NSUB4=NSUB                                                        SHEA  357C
    4 CALL READMS (4,E1,25,NSREC4)                                      SHEA  358C
      IF (IWRIT.GT.2) GO TO 417                                         SHEA  359
      WRITE (6,6150)                                                    SHEA  360
```

```
 6150 FORMAT (1H , 35H -E1- UNIT 4 SURFACE PROPERTY DATA )
      WRITE (6,6100) E1
  417 CONTINUE
C
C     CHECK ELEMENT NO
      IJ2 = E1(2)
      IF (K.EQ.IJ2) GO TO 420
      WRITE (6,6800) IJ2, K
 6800 FORMAT (1H , 36HHARK, SURF. PROPERTY ARRAY (E1) NO =, I4, 2X, 21HB
     1UT THE COUNTER NO IS, I4)
      GO TO 1000
  420 CONTINUE
C
C     SUM SHEAR, BENDING MOMENT, TORQUE, AND HINGE MOMENTS
      IF (I-2) 110,120,130
C
  110 CONTINUE
C     FUSELAGE SHEARS AND BENDING MOMENTS
      DO 500 L=1,I2
      DIST=XCENT-XSHR(L,I)
C
C     CHECK ELEMENT CENTROID FOR SHEAR CALCULATION (X NEG. AFT)
      IF (DIST.LE.0.0) GO TO 500
      CV(L,I,KA) = CV(L,I,KA) + E1(15)
      CB(L,I,KA) = CB(L,I,KA) + E1(15)*DIST + E1(13)*(ZCENT-ZSHR(L,I))
  500 CONTINUE
      GO TO 400
C
  120 CONTINUE
C     WING SHEARS, BENDING MOMENTS, AND TORQUES (WITHOUT VERT. PANELS)
C         NOTE - VERTICAL PANELS ADDED FOR I GREATER THAN 2
      DO 600 L=1,I2
      DIST=ABS(YCENT)-ABS(YSHR(L,I))
      IF (DIST.LE.0.0) GO TO 600
      DDC = E1(15)*WCOS + E1(13)*WSIN
      CV(L,I,KA) = CV(L,I,KA) + DDC
      CB(L,I,KA) = CB(L,I,KA) + DDC*DIST*MP + E1(14)*
     1             (ZCENT - (6.19338 - 0.02094 * XCENT)) * WCOS
      CT(L,I,KA) = CT(L,I,KA) + E1(15)*(XCENT-XSHR(L,I)) +
     1             E1(13)*(ZCENT-ZSHR(L,I))
  600 CONTINUE
C
C     ELEVON HINGE MOMENTS TOO
      IF (J.GT.8 ) GO TO 140
      GO TO 400
```

```
C
  130 CONTINUE
C     VERTICAL SURFACE SHEARS, BENDING MOMENTS, AND TORQUES
      DO 700 L=1,I2
      DIST =ZCENT-ZSHR(L,I)
      IF(I.GT.3) GO TO 710
      IF(DIST.LE.0.0) GO TO 725
      GO TO 720
  710 CONTINUE
      IF(DIST.GT.0.0) GO TO 725
  720 CONTINUE
      SA = (ABS(YCENT)-ABS(YSHR(L,I)))*MP
      DDC = E1(14)*VCS + E1(15)*VSN
      CV(L,I,KA) = CV(L,I,KA) + DDC
      CB(L,I,KA) = CB(L,I,KA) + E1(14)*DIST + E1(15)*SA
      CT(L,I,KA) = CT(L,I,KA) + DDC*(XCENT - XSHR(L,I)) +
     1             E1(13)*(DIST*VSN - SA*VCS)
  725 CONTINUE
C
C     ADD VERTICAL FORCES AND MOMENTS TO WING DATA
      N = 0
  730 CONTINUE
      N = N + 1
      DIST = ABS(YCENT) - ABS(YSHR(N,2))
      IF (DIST.LE.0.0) GO TO 740
      DDC = E1(15)*WCOS + E1(13)*WSIN
      CV(N,2,KA) = CV(N,2,KA) + DDC
      CB(N,2,KA) = CB(N,2,KA) + DDC*DIST*MP + E1(14)*
     1             (ZCENT - (6.19338 - 0.02094 * XCENT)) * WCOS
      CT(N,2,KA) = CT(N,2,KA) + E1(15)*(XCENT - XSHR(N,I)) +
     1             E1(13)*(ZCENT - ZSHR(N,I))
  740 CONTINUE
      IF (N.LT.NSHST(2)) GO TO 730
  700 CONTINUE
      GO TO 400
C
  140 CONTINUE
C     ELEVON HINGE MOMENT
      IF (J.GT.10) I7=KA *2
      HME(I7) = HME(I7) +HMF*E1(5)
  400 CONTINUE
C
C     INTERMEDIATE WRITE FOR DATA CHECK OUT
      IF (IWRIT.GT.3) GO TO 300
      WRITE (6,6460) I, J, NOFP(J,I,KA)
```

```
 6460 FORMAT (1H , 4HI = , I3, 6H, J = , I3, 20H, PANEL NO. (J,I) = ,I4)  SHEA  451
      WRITE (6,6410)                                                      SHEA  452
 6410 FORMAT (1H , 15X, 3(1HV, 11X, 1HB, 11X, 1HT, 11X))                  SHEA  453
      WRITE (6,6400) (((CV(N,M,L), CB(N,M,L), CT(N,M,L),                  SHEA  454
     1       N=1,3), M=1,5), L=1,KA)                                      SHEA  455
 6400 FORMAT (1H , 8HFUSELAGE, 9F12.5/1X, 4HWING, 4X, 9F12.5/1X,8HVERTIC  SHEA  456
     1AL, 9F12.5, /1X, 8HNAC VENT, 9F12.5/1X, 8HFUS VENT, 9F12.5)         SHEA  457
      WRITE (6,6420) (HME(M), M=1,I7)                                     SHEA  458
 6420 FORMAT (1H , 15HIB ELEVON HM = , F12.5,16H, OB ELEVON HM =, F12.5)  SHEA  459
  300 CONTINUE                                                            SHEA  460
C                                                                         SHEA  461
C     BOOK KEEPING TO GET ALL DATA FOR ALPHA OR BETA                      SHEA  462
C                                                                         SHEA  463
      IF (I.LT.4) GO TO 100                                               SHEA  464
      IF (KAB .EQ.1) GO TO 180                                            SHEA  465
      IF (KA.EQ.KAB) GO TO 160                                            SHEA  466
      KA  = KAB                                                           SHEA  467
      MP  = -1                                                            SHEA  468
      I7 = 3                                                              SHEA  469
      I  = 0                                                              SHEA  470
      GO TO 100                                                           SHEA  471
  160 CONTINUE                                                            SHEA  472
      IF (I.EQ.4) GO TO 100                                               SHEA  473
  180 CONTINUE                                                            SHEA  474
C                                                                         SHEA  475
C   END OF SUMMATIONS FINALIZE COEFFICIENTS                               SHEA  476
C                                                                         SHEA  477
      K  = 3 + KAB                                                        SHEA  478
      DO 800 J=1,K                                                        SHEA  479
      I2 = NSHST(J)                                                       SHEA  480
      DO 850 I=1,I2                                                       SHEA  481
      SA = 1794.24 / SSHR(I,J)                                            SHEA  482
      SBND = SA/BSHR(I,J)                                                 SHEA  483
      SCRD = SA/CSHR(I,J)                                                 SHEA  484
      IF (IWRIT.LT.1) WRITE(6,6450)SSHR(I,J),CSHR(I,J),BSHR(I,J),SBND,SA  SHEA  485
     1                                          ,SCRD                    SHEA  486
 6450 FORMAT (1H , 7HSSHR = , F12.5, 9H, CSHR = , F12.5, 9H, BSHR = ,     SHEA  487
     1 F12.5, 9H, SBND = , F12.5, 7H, SA = , F12.5, 9H, SCRD = , F12.5)   SHEA  488
      DO 875 L=1,KA                                                       SHEA  489
      CV(I,J,L) = CV(I,J,L) * SA                                          SHEA  490
      CB(I,J,L) = CB(I,J,L) * SBND                                        SHEA  491
      IF (J.EQ.1) GO TO 875                                               SHEA  492
      CT(I,J,L) = CT(I,J,L) * SCRD                                        SHEA  493
  875 CONTINUE                                                            SHEA  494
  850 CONTINUE                                                            SHEA  495
```

```
  800 CONTINUE                                                          SHEA  496
      I7 = 2*KAB                                                        SHEA  497
      DO 900 I=1,I7                                                     SHEA  498
      CHE(I) = HME(I)/(SEH(I) * CEH(I))                                 SHEA  499
  900 CONTINUE                                                          SHEA  500
C                                                                       SHEA  501
C     FINAL FUSELAGE DATA CALCULATION FOR ALPHA AND BETA                SHEA  502
      I3 = (3 - KAB)                                                    SHEA  503
      J = 0                                                             SHEA  504
  820 CONTINUE                                                          SHEA  505
      J = J + 1                                                         SHEA  506
      I2 = NSHST(1)                                                     SHEA  507
      DO 825 I = 1,I2                                                   SHEA  508
      CV(I,1,J) = CV(I,1,1)*I3 + CV(I,1,2)*(J-1)                        SHEA  509
      CB(I,1,J) = CB(I,1,1)*I3 + CB(I,1,2)*(J-1)                        SHEA  510
  825 CONTINUE                                                          SHEA  511
      IF (J.LT.KAB) GO TO 820                                           SHEA  512
C                                                                       SHEA  513
C     WRITE FINALIZED COEFFICIENTS                                      SHEA  514
      WRITE (6,6430)                                                    SHEA  515
 6430 FORMAT (1H , 14X, 3(2HCV, 10X, 2HCB, 10X, 2HCT, 10X))             SHEA  516
      WRITE (6,6400)(((CV(N,M,L),CB(N,M,L),CT(N,M,L),N=1,3),M=1,K),     SHEA  517
     1      L=1,KAB)                                                    SHEA  518
      WRITE (6,6440) (          CHE(M), M=1,17)                         SHEA  519
 6440 FORMAT (1H , 15HIB ELEVON CH = , F12.5, 16H, OB ELEVON CH =,F12.5) SHEA  520
C                                                                       SHEA  521
C   CHECK IF CASE DATA TO BE SAVED                                      SHEA  522
C     ISVFMS = 0, DO NOT SAVE CASE                                      SHEA  523
C            = 1, SAVE NEW CASE ON EXISTING DATA SET                    SHEA  524
C            = 2, REPLACE CASE ON  EXISTING DATA SET                    SHEA  525
C            = 3, GENERATE A NEW DATA SET                               SHEA  526
C                                                                       SHEA  527
      IF (ISVFMS .EQ. 0)  GO TO 9000                                    SHEA  528
      IF (ISVFMS .NE. 3)  GO TO 8010                                    SHEA  529
C   SET UP NEW DATA SET                                                 SHEA  530
      NEXRUN = 1                                                        SHEA  531
      NEXDAT = 51                                                       SHEA  532
      IG20=1                                                            SHEA  533C
      CALL WRITMS (20,NDUM,2,IG20,0)                                    SHEA  534C
C   ZERO OUT RUN ID AND CORRELATION COUNT ARRAYS                        SHEA  535
      DO 8000 I = 1,216                                                 SHEA  536
      IRUNID(I) = 0                                                     SHEA  537
 8000 NCORR(I)  = 0                                                     SHEA  538
      IG20=3                                                            SHEA  539C
      CALL WRITMS (20,IRUNID,216,IG20,0)                                SHEA  540C
```

```
      IG20=5                                                            SHEA   541C
      CALL WRITMS (20,NCORR,216,IG20,0)                                 SHEA   542C
      DO 8005 I = 1,108                                                 SHEA   543
      ICORID(I) = 0                                                     SHEA   544
 8005 IDAPT(I)  = 0                                                     SHEA   545
C                                                                       SHEA   546
C  INITIALIZE CORRELATION AND POINTER DIRECTORIES                       SHEA   547
      DO 8006 I = 7,50                                                  SHEA   548
      IG20 = I                                                          SHEA   549
C     WRITE(20'IG20)  ICORID                                            SHEA   550I
      CALL WRITMS (20,ICORID,108,IG20,0)                                SHEA   551C
 8006 CONTINUE                                                          SHEA   552
      GO TO 8015                                                        SHEA   553
C                                                                       SHEA   554
C  ADD CASE DATA TO AN EXISTING DATA SET                                SHEA   555
 8010 IG20 = 1                                                          SHEA   556
C     READ (20'IG20) NDUM                                               SHEA   557I
      CALL READMS (20,NDUM,2,IG20)                                      SHEA   558C
      IG20 = 3                                                          SHEA   559
C     READ(20'IG20)  IRUNID                                             SHEA   560I
      CALL READMS (20,IRUNID,216,IG20)                                  SHEA   561C
      IG20 = 5                                                          SHEA   562
C     READ(20'IG20)  NCORR                                              SHEA   563I
      CALL READMS (20,NCORR,216,IG20)                                   SHEA   564C
C                                                                       SHEA   565
C  DETERMINE RUN SEQUENCE NUMBER                                        SHEA   566
 8015 DO 8020 I = 1,216                                                 SHEA   567
      IRSQ = I                                                          SHEA   568
      IF (IRUNID(I) .EQ. IRUN)  GO TO 8040                              SHEA   569
      IF (IRUNID(I) .EQ. 0)     GO TO 8030                              SHEA   570
 8020 CONTINUE                                                          SHEA   571
C                                                                       SHEA   572
C  RUN ID COULD NOT BE FOUND.  SAVE AS NEW RUN                          SHEA   573
 8030 IRSQ = NEXRUN                                                     SHEA   574
      NEXRUN = NEXRUN + 1                                               SHEA   575
      IRUNID(IRSQ) = IRUN                                               SHEA   576
C                                                                       SHEA   577
C  DETERMINE RECORD NUMBER FOR CORRELATION ID AND DATA POINTERS         SHEA   578
 8040 IREC = 7 + (IRSQ - 1)/7                                           SHEA   579
      IDPT = IREC + 22                                                  SHEA   580
C                                                                       SHEA   581
C  DETERMINE SEQUENCE NUMBER FOR POSITION OF FIRST CORRELATION NO.      SHEA   582
      ICSQ = 1 + ((IRSQ-1) - 7*(IREC-7))*15                             SHEA   583
C                                                                       SHEA   584
C     READ(20'IREC)  ICORID                                             SHEA   585I
```

```
      CALL READMS (20,ICORID,108,IREC)                                  SHEA   586C
C     READ(20'IDPT)  IDAPT                                              SHEA   587I
      CALL READMS (20,IDAPT,108,IDPT)                                   SHEA   588C
C                                                                       SHEA   589
      I1 = ICSQ                                                         SHEA   590
      IF (ISVFMS .NE. 2)  GO TO 8060                                    SHEA   591
C                                                                       SHEA   592
C  DETERMINE CORRELATION SEQUENCE NUMBER                                SHEA   593
      IF (NCORR(IRSQ) .EQ. 0)  GO TO 8060                               SHEA   594
      I2 = I1 + NCORR(IRSQ) - 1                                         SHEA   595
      DO 8050 I = I1,I2                                                 SHEA   596
      ICSQ = I                                                          SHEA   597
      IF (ICORID(I) .EQ. ICOR)  GO TO 8070                              SHEA   598
      IF (ICORID(I) .EQ. 0)     GO TO 8060                              SHEA   599
 8050 CONTINUE                                                          SHEA   600
 8060 NCORR(IRSQ) = NCORR(IRSQ) + 1                                     SHEA   601
      ICSQ = I1 + NCORR(IRSQ) - 1                                       SHEA   602
      ICAPT(ICSQ) = NEXDAT                                              SHEA   603
      NEXDAT = NEXDAT + 1                                               SHEA   604
 8070 ICORID(ICSQ) = ICOR                                               SHEA   605
C  RETRIEVE FORCE AND MOMENT DATA FROM UNIT 9                           SHEA   606
      IG9 = 3                                                           SHEA   607
C     READ(9'IG9)  E9                                                   SHEA   608I
      CALL READMS (9,E9,41,IG9)                                         SHEA   609C
      DO 8100 I = 2,41                                                  SHEA   610
 8100 LSUM(I-1) = E9(I)                                                 SHEA   611
      IG9 = 4                                                           SHEA   612
C     READ(9'IG9)  E9                                                   SHEA   613I
      CALL READMS (9,E9,41,IG9)                                         SHEA   614C
      EMACH = E9(1)                                                     SHEA   615
      ISUM = 1                                                          SHEA   616
      IF (IAB .GT. 0)  ISUM = 3                                         SHEA   617
      IG9 = LSUM(ISUM) + 2                                              SHEA   618
C     READ(9'IG9)  F                                                    SHEA   619I
      CALL READMS (9,F,11,IG9)                                          SHEA   620C
C                                                                       SHEA   621
C  SAVE CASE DATA                                                       SHEA   622
      MDUM(2)=IAB                                                       SHEA   623
      IG20 = IDAPT(ICSQ)                                                SHEA   624
C     WRITE (20'IG20) MDUM                                              SHEA   625I
      CALL WRITMS (20,MDUM,108,IG20,0)                                  SHEA   626C
      CALL WRITMS (20,MDUM,108,IG20)                                    SHEA   627C
C                                                                       SHEA   628
C  RESET DIRECTORIES                                                    SHEA   629
      IG20 = 1                                                          SHEA   630
```

```
C     WRITE (20'IG20) NDUM                                          SHEA  631I
      CALL WRITMS (20,NDUM,2,IG20,1)                                SHEA  632C
      IG20 = 3                                                      SHEA  633
C     WRITE(20'IG20)  IRUNID                                        SHEA  634I
      CALL WRITMS (20,IRUNID,216,IG20,1)                            SHEA  635C
      IG20 = 5                                                      SHEA  636
C     WRITE(20'IG20)  NCORR                                         SHEA  637I
      CALL WRITMS (20,NCORR,216,IG20,1)                             SHEA  638C
C     WRITE(20'IREC)  ICORID                                        SHEA  639I
      CALL WRITMS (20,ICORID,108,IREC,1)                            SHEA  640C
C     WRITE(20'IDPT)  IDAPT                                         SHEA  641I
      CALL WRITMS (20,IDAPT,108,IDPT,1)                             SHEA  642C
C                                                                   SHEA  643
      WRITE(6,8200)                                                 SHEA  644
      WRITE(22,8200)                                                SHEA  645
 8200 FORMAT(1H0, 37HCASE DATA HAVE BEEN SAVED ON UNIT 20.)         SHEA  646
 9000 CONTINUE                                                      SHEA  647
      WRITE (6,6500)                                                SHEA  648
      WRITE(22,6500)                                                SHEA  649
 6500 FORMAT (1H , 19HEND OF SHEAR OPTION)                          SHEA  650
C                                                                   SHEA  651
C     INITIALIZE COEFF. TO ZERO                                     SHEA  652
      DO 950 I=1,94                                                 SHEA  653
  950 ZERO(I) = 0.0                                                 SHEA  654
      DO 955 I=1,4                                                  SHEA  655
      HME(I) = 0.0                                                  SHEA  656
  955 CONTINUE                                                      SHEA  657
      GO TO 1010                                                    SHEA  658
 1000 CONTINUE                                                      SHEA  659
      IERR = 1                                                      SHEA  660
C1010 RETURN                                                        SHEA  661I
 1010 CONTINUE                                                      SHEA  662C
      END                                                           SHEA  663
```

```
      SUBROUTINE  SOLVIT(A,ND,MD,KD,NI,MM,NO,NW,BB,NNB,NMB,NERR)
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     WRITTEN BY J. L. HESS * PROGRAMMED BY T. M. RIDDELL
C
C  THIS VERSION MODIFIED TO DO  PIVOTAL SEARCH BY ROWS.
C  REFER TO PAGE 35 OF DAC-7000 FOR DESCRIPTION OF METHOD.
C  CODED BY  D. N. SMYTH
C
C
      DIMENSION  A ( KD ), IRC(500), ISC(500)
C  AA IS USED WHEN SOLUTIONS ARE WRITTEN ON TAPE NW.
C     DIMENSION  AA(500)
C BB IS USED WHEN SOLUTIONS ARE RETURNED THROUGH CALL STATEMENT.
      DIMENSION BB(NNB,NMB)
C
      LOGICAL  LAST
C
      NERR = 0
      N = ND
      M = MD
      KORE = KD
      NPM = N + M
      IF (MAX0(3 * NPM, M * N) .GT. KORE) STOP 1
      MT = MM
      REWIND MT
      NIN = NI
      REWIND NIN
      NOUT = NO
      REWIND NOUT
      MP1 = M + 1
      NN = N
      NEL = NPM
C
C  INITIALIZE INTERCHANGE COUNTERS
      DO  5  I = 1,N
      IRC(I) = 0
```

```
      ISC(I) = I
    5 CONTINUE
      IP = 0
      INT = 0
      NM1 = N - 1
C
C - - CALCULATE THE MAXIMUM NO. OF ROWS, 'K'
C
   10 K = (KORE - NEL) / NEL
C
C - - TEST TO SEE IF THE REST OF THE MATRIX WILL FIT IN CORE
C
      LAST = K .GE. NN
      IF (LAST) K = NN
      IF ((NN-K) .EQ. 1)  K = K - 1
      IRO = IR
C
C - - READ 'K' ROWS OF THE AUGMENTED 'A' MATRIX
C
   20 NT = 0
      DO 30 IB = 1, K
      NS = NT + 1
      NT = NT + NEL
   30 CALL  GETT(NTN, 1, NEL, A(NS), 1, AA2)
C
C - - CHECK TO SEE IF WE WERE UNLUCKY ENOUGH TO END UP WITH ONLY ONE ROW
C
      IF (K .EQ. 1) GO TO 90
C
C - - 'K' IS GREATER THAN '1' SO WE CAN START THE TRIANGULARIZATION
C
      NELP1 = NEL + 1
      NS = - NEL
      NELP2 = NELP1 + 1
C
C - - FORM THE 'TRAPEZOIDAL' ARRAY  (8)
C
      K1 = K
      IF (K .LT. NN)  K1 = K1 + 1
      DO 40 IB = 2,K1
      NP = NELP2 - IB
      NS = NS + NELP1
      NT = NS
C
C  SEARCH FOR PIVOTAL ELEMENT, AMAX
```

```
      IMR = 0                                                           SOLV  091
      AMAX = 0.0                                                        SOLV  092
      I = IR                                                            SOLV  093
      IR = IR + 1                                                       SOLV  094
      NN = NS + N - IR                                                  SOLV  095
      DO 35 I2 = NS,NN                                                  SOLV  096
      I = I + 1                                                         SOLV  097
      IF (ABS(A(I2)) .LT. ABS(AMAX))  GO TO 35                          SOLV  098
      AMAX = A(I2)                                                      SOLV  099
      IMA = I2                                                          SOLV  100
      IMR = I                                                           SOLV  101
   35 CONTINUE                                                          SOLV  102
C  SET INTERCHANGE COUNTERS AND INTERCHANGE ELEMENTS IN BASE ROW        SOLV  103
      IF (IMR .EQ. 0)  GO TO 36                                         SOLV  104
      IRC (IR) = IMR                                                    SOLV  105
      A(IMA) = A(NS)                                                    SOLV  106
      A(NS) = AMAX                                                      SOLV  107
C                                                                       SOLV  108
      IS = ISC(IR)                                                      SOLV  109
      ISC(IR) = ISC(IMR)                                                SOLV  110
      ISC(IMR) = IS                                                     SOLV  111
C                                                                       SOLV  112
   36 CONTINUE                                                          SOLV  113
      IF (IR .GT. K)  GO TO 45                                          SOLV  114
      DO 40 IO = IR, K                                                  SOLV  115
      NT = NT + NEL                                                     SOLV  116
      MN = NT                                                           SOLV  117
      NB = NS                                                           SOLV  118
C                                                                       SOLV  119
C  INTERCHANGE ELEMENTS IN OBJECT ROW                                   SOLV  120
      IMA = IMA + NEL                                                   SOLV  121
      AMAX = A(IMA)                                                     SOLV  122
      A(IMA) = A(NT)                                                    SOLV  123
      A(NT) = -AMAX/A(NS)                                               SOLV  124
      DO 40 NF = 2, NP                                                  SOLV  125
      MN = MN + 1                                                       SOLV  126
      NB = NB + 1                                                       SOLV  127
   40 A(MN) = A(MN) + A(NT) * A(NB)                                     SOLV  128
   45 CONTINUE                                                          SOLV  129
      IF (LAST) GO TO 90                                                SOLV  130
C                                                                       SOLV  131
C - - WRITE THE 'TRAPEZOIDAL' MATRIX ON TAPE                            SOLV  132
C                                                                       SOLV  133
      NT = 0                                                            SOLV  134
      NP = NEL                                                          SOLV  135
```

```
      NS = - NEL
      DO 50 IO = 1, K
      NS = NS + NELP1
      NT = NT + NEL
      NN = NS + NP - 1
      CALL  SAVE(MT, 2, NP, NP, A(NS), 1, AA2)
   50 NP = NP - 1
      NP = NP - M
      NS = KORE - NEL + 1
C
C - - READ ANOTHER ROW
C
      DO 80 IO = 1, NP
      CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)
C
C - - MODIFY THIS ROW BY THE 'TRAPEZOIDAL' ARRAY
C
      IR = IRO
      NT = 1
      MN = NS
      DO 70 IR = 1, K
      NB = NT
      NF = MN + 1
      IR = IR + 1
      I2 = NS + IRO(IR) - 1 - IRO
      AMAX = A(I2)
      A(I2) = A(MN)
      A(MN) = -AMAX/A(NT)
      DO 60 NN = NF, KORE
      NB = NB + 1
   60 A(NN) = A(NN) + A(MN) * A(NB)
      MN = NF
   70 NT = NT + NELP1
C
C - - WRITE THE MODIFIED ROW ON TAPE
C
      NN1 = KORE - MN + 1
   80 CALL  SAVE(NOUT, 1, NN1, NN1, A(MN), 1, AA2)
      REWIND NOUT
      REWIND NIN
C
C - - SWITCH THE TAPES
C
      NT = NIN
      NIN = NOUT
```

```
      NOUT = NT
C
C - - RE-CALCULATE ROW LENGTH AND LOOP BACK
C
      NEL = NEL - K
      NN = NEL - M
      GO TO 10
C
C - - REWIND ALL TAPES
C
   90 REWIND MT
      REWIND NIN
      REWIND NOUT
C
C  HAVE K ROWS OF TRAPEZOIDAL ARRAY IN CORE.
C  PERFORM COLUMN INTERCHANGES ON PRECEEDING ROWS
C  DUE TO FOLLOWING ROWS
C
      IF (K .LT. 3)  GO TO 96
      NN = K - 2
      NS = -NEL
      DO 95 IB = 1,NN
      NS = NS + NELP1
      I1 = IB + IRO + 1
      DO 94 IR = I1,NM1
      IF (IRC(IR) .EQ. IR)  GO TO  94
      I2 = IRC(IR) - IRO
      IF (I2 .LE. C)  GO TO 94
      I2 = I2 + NS - IB
      I3 = IR - IRO + NS - IB
      AMAX = A(I2)
      A(I2) = A(I3)
      A(I3) = AMAX
   94 CONTINUE
   95 CONTINUE
C
C - - CONDENSE THE MATRIX
C
   96 NN = NEL
      NL = NEL + 1
      IF (K .EQ. 1) GO TO 110
      NS = 1
      NT = NEL
      DO 100 IB = 2, K
      NS = NS + NELP1
```

```
      NT = NT + NEL                                                     SOLV  226
      DO 100 IO = NS, NT                                                SOLV  227
      A(NL) = A(IO)                                                     SOLV  228
  100 NL = NL + 1                                                       SOLV  229
  110 N1 = KORE - K * M + 1                                             SOLV  230
C                                                                       SOLV  231
C - - THERE, NOW WE CAN START THE BACK-SOLUTION                         SOLV  232
C * * NOTE..THE FIRST AVAILABLE LOCATION FOR THE SOLUTIONS IS A(N1)     SOLV  233
C                                                                       SOLV  234
      NPEM = N                                                          SOLV  235
      NEL = NPM                                                         SOLV  236
      LAST = K .EQ. N                                                   SOLV  237
      NPASS = 0                                                         SOLV  238
C                                                                       SOLV  239
C - - SOLVE FOR THE ANSWERS CORRESPONDING TO 'K' ROWS                   SOLV  240
C                                                                       SOLV  241
  120 KM1 = K - 1                                                       SOLV  242
      KP1 = K + 1                                                       SOLV  243
      NS = NL - MP1                                                     SOLV  244
      NPASS = NPASS + 1                                                 SOLV  245
      DO 150 MN = 1, M                                                  SOLV  246
      NF = NS + MN                                                      SOLV  247
      A(NF) = A(NF) / A(NS)                                             SOLV  248
      NT = NS                                                           SOLV  249
      IF (KM1 .EQ. 0) GO  TO 150                                        SOLV  250
      DO 140 IB = 1, KM1                                                SOLV  251
      NF = NF - IB - M                                                  SOLV  252
      NT = NT - MP1 - IB                                                SOLV  253
      SUM = 0.0                                                         SOLV  254
      NP = NF                                                           SOLV  255
      N2 = MP1 + IB                                                     SOLV  256
      DO 130 IO = 1, IB                                                 SOLV  257
      NN = NT + IC                                                      SOLV  258
      NP = NP + N2 - IO                                                 SOLV  259
  130 SUM = SUM + A(NN) * A(NP)                                         SOLV  260
  140 A(NF) = (A(NF) - SUM) / A(NT)                                     SOLV  261
  150 CONTINUE                                                          SOLV  262
C                                                                       SOLV  263
C - - MOVE THE SOLUTIONS TO CONTIGUOUS LOCATIONS STARTING AT A(N1)      SOLV  264
C                                                                       SOLV  265
      N1 = KORE + 1                                                     SOLV  266
      DO 170 NN = 1, K                                                  SOLV  267
      DO 160 MN = 1, M                                                  SOLV  268
      NL = NL - 1                                                       SOLV  269
      N1 = N1 - 1                                                       SOLV  270
```

```
  160 A(N1) = A(NL)
  170 NL = NL - NN
C
C - - WRITE THE SOLUTIONS ON TAPE
C
      WRITE (NIN) K
      NS = N1 - 1
      DO 180 MN = 1, M
      NT = NS + MN
  180 WRITE (NIN ) (A(IO), IO = NT, KORE, M)
C
C - - TEST IF THIS IS THE LAST PASS
C
      IF (LAST) GO TO 260
C
C - - WE MUST NOW MODIFY THE TRIANGULAR MATRIX TO REFLECT THE EFFECT OF
C     THE SOLUTIONS OBTAINED SO FAR  (EW 21)
C * * NOTE..LOCATIONS A(1) TO A(N1-1) ARE NOW FREE TO USE
C
C - - CALCULATE THE NEXT VALUES OF 'NEL' AND 'NREM'
C
      NELOLD = NEL
      KOLD = K
      NEL = NEL - K
      NREM = NREM - K
C
C$$$$ CALCULATE NEW K.  B AND C (REAL) WILL ALWAYS BE INTEGERS.
C     K WILL BE CALCULATED REAL AND TRUNCATED - - GOOD.
C
      B = 1 + 2*M
      C = 2*(KOLD*(M+1) - KORE)
      K = (-B + SQRT(B**2 - 4*C))/2.0
      NROW = NREM - K + 1
      IF (K .LT. NREM) GO TO 190
      LAST = .TRUE.
      NROW = 1
      K = NREM
  190 NS = 1
      NT = NELOLD + 1
      NR2 = N - 1
C
C - - READ IN THE ROWS TO BE MODIFIED
C
C
      DO 250 IB = 1, NREM
```

```
      NT = NT - 1
      IF (IB .LE. NROW) GO TO 200
      NS = NS + NN
      NT = NT + NN
  200 NN = NT - NS + 1
      CALL  GETT(MT, 2, NN, A(NS), 1, AA2)
C
C  THE FIRST TIME THE TRAPEZOIDAL ARRAY IS READ BACK,
C  THE COLUMN ORDER IS MADE CONSISTENT -
C  THAT IS, THE COLUMN INTER CHANGES OF FOLLOWING
C  ROWS ARE MADE
C
      IF (INT .NE. 0)  GO TO 205
      NR1 = IB + 1
      DO 204 IR = NR1,NR2
      IF (IRC(IR) .EQ. IR)  GO TO 204
      I2 = IRC(IR) + NS - IB
      I3 = IR - IB + NS
      AMAX = A(I2)
      A(I2) = A(I3)
      A(I3) = AMAX
  204 CONTINUE
  205 CONTINUE
      NP = N1 - 1
      NF = NT - M - KM1
      NN = NN - KOLD
      DO 220 MN = 1,M
      N2 = NF
      NA = NP + MN
      NB = NA
      SUM = 0.0
      DO 210 IO = 1, KOLD
      SUM = SUM + A(N2) * A(NA)
      N2 = N2 + 1
  210 NA = NA + M
      N2 = N2 + MN - 1
  220 A(N2) = A(N2) - SUM
C
C - - WRITE THE MODIFIED ROW ON TAPE OR CONDENSE THE ROW
C
      NL = NT - M + 1
      IF (IB .GE. NROW) GO TO 230
      NF = NL - KP1
      NN1 = NF - NS + 1
      NN2 = NT - NL + 1
```

```
      CALL  SAVE(NOUT, 4, NN, NN1, A(NS), NN2, A(NL))
      GO TO 250
  230 NF = NL - KOLD
      DO 240 MN = NL, NT
      A(NF) = A(MN)
  240 NF = NF + 1
  250 CONTINUE
      INT = 1
      REWIND MT
      REWIND NOUT
C
C - - SWITCH THE TAPES
C
      NT = MT
      MT = NOUT
      NOUT = NT
C
C - - LOOP BACK THRU THE SOLUTION
C
      NL = NF
      GO TO 120
C
C - - START TO WRAP IT UP
C
  260 REWIND NIN
      N2 = N
C
C * * NOTE.. AT THIS POINT ALL LOCATIONS A(1) THRU A(KORE) ARE FREE
C
      DO 280 IR = 1, NPASS
      READ (NIN) K
      N1 = N2 - K + 1
      NS = N1
      NT = N2
C
C - - READ IN THE SOLUTIONS
C
      DO 270 IO = 1, M
      NM = NT - NS + 1
      CALL  GETT(NIN, 1,NM, A(NS), 1, AA2)
      NT = NT + N
  270 NS = NS + N
  280 N2 = N1 - 1
C
C - - WRITE THE SOLUTIONS ON TAPE
```

```
C
      NT = 0                                                            SOLV  406
      DO 290 IO = 1, M                                                  SOLV  407
      NS = NT + 1                                                       SOLV  408
      NT = NT + N                                                       SOLV  409
C  ALL N*M SOLVTIONS ARE IN CORE.  CORRECT ORDER                        SOLV  410
C  DUE TO INTERCHANGES MADE DURING SOLUTION.                            SOLV  411
C                                                                       SOLV  412
      DO 289 I = 1,N                                                    SOLV  413
      I1 = ISC(I)                                                       SOLV  414
      I2 = NS + I - 1                                                   SOLV  415
C     AA(I1) = A(I2)                                                    SOLV  416
      BB(I1,IO)= A(I2)                                                  SOLV  417
  289 CONTINUE                                                          SOLV  418
C     CALL SAVE(NW, 1, N, N, AA(1), 1, AA2)                             SOLV  419
  290 CONTINUE                                                          SOLV  420
C                                                                       SOLV  421
      RETURN                                                            SOLV  422
      END                                                               SOLV  423
                                                                        SOLV  424
```

```
      OVERLAY (MARK4,2,5)                                               SPEC  001C
      PROGRAM SPEC                                                      SPEC  002C
C     SUBROUTINE SPEC                                                   SPEC  003I
C                                                                       SPEC  004
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR                                SPEC  005
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   SPEC  006
     1            TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                         SPEC  007
      INTEGER ERROR,PAGE                                                SPEC  008
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,       SPEC  009
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                            SPEC  010
      DIMENSION TITLE(15),IPG(20)                                       SPEC  011
C  READ SUB-OPTION CONTROL CARD                                         SPEC  012
      READ (TAPEIN,10) IPG                                              SPEC  013
   10  FORMAT (20I1)                                                    SPEC  014
C                                                                       SPEC  015
      DO 80 I=1,20                                                      SPEC  016
       IPROG = IPG(I)                                                   SPEC  017
       IF (IPROG .EQ. 0) GO TO 90                                       SPEC  018
      GO TO (30,40), IPROG                                              SPEC  019
C                                                                       SPEC  020
C  FORCE DATA SUMMATION OPTION                                          SPEC  021
   30 CALL SUM                                                          SPEC  022
      GO TO 80                                                          SPEC  023
C                                                                       SPEC  024
C  TRIM OPTION (TO BE ADDED LATER)                                      SPEC  025
   40 CONTINUE                                                          SPEC  026
      GO TO 80                                                          SPEC  027
C                                                                       SPEC  028
   80 CONTINUE                                                          SPEC  029
C                                                                       SPEC  030
C  90 RETURN                                                            SPEC  031I
   90 CONTINUE                                                          SPEC  032C
      END                                                               SPEC  033
```

```
      SUBROUTINE STATS3
C
C   THIS ROUTINE PREPARES A DUMMY, ZERO-AREA ELEMENT TO PROVIDE THE
C   FINAL STATUS FLAG OF 3
C
C
      COMMON /ACFT/IPRINT,ITAPE,ITYPE,P1
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK
C
C
      WRITE (ITAPE,10)
   10 FORMAT (51H     0.0        0.0        0.0    2     0.0        0.0    ,
     1 29H     0.00010          3          )
      WRITE (ITAPE,20)
   20 FORMAT (51H    -0.0001     0.0        0.0    1    -0.0001     0.0    ,
     1 29H     0.00013          3          )
      IF (IPRINT .EQ. 1) WRITE (TAPEOT,30)
      IF (IPRINT .EQ. 1) WRITE (TAPEOT,40)
   30 FORMAT(52H      0.0        0.0        0.0    2     0.0        0.0    ,
     1 29H     0.00010          3          )
   40 FORMAT(52H     -0.0001     0.0        0.0    1    -0.0001     0.0    ,
     1 29H     0.00013          3          )
C
      RETURN
      END
```

```
      SUBROUTINE SUM                                                    SUM    001
C                                                                       SUM    002
C   THIS OPTION IS USED TO SUM UP SELECTED FORCE DATA SETS.  IT ALSO    SUM    003
C   MAY BE USED TO PUNCH SELECTED SETS OF FCRCE DATA.                   SUM    004
C                                                                       SUM    005
      COMMON /EXEC/CASE,TITLE,PAGE,ERROR                                SUM    006
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   SUM    007
     1            TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                         SUM    008
      COMMON /FLIGHT/MACH,ALT,PSTAG,TSTAG,V,RENO,PFS,TFS,AFS,RHOFS,VIS  SUM    009
      COMMON /GASD/GAM,GASCP,PRAN,IGAS,AV1,AV2,AV3,GTYPE(2)             SUM    010
      COMMON /REF/SREF,MAC,SPAN,XCG,YCG,ZCG                             SUM    011
      COMMON/INDATA/IRUNID,ICORID,IPLTCP,IBETA,BETAIN,IWRIT             SUM    012
C                                                                       SUM    013
      DIMENSION E1(8),E2(11),E4(41),F(11),FF(11),E5(41),E6(41),         SUM    014
     1    IPANL(10),IPAN9(10,50),LSUM(41),LCOM(81),E7(11),ICOMP(50),    SUM    015
     2  E8(14)                                                          SUM    016
      DIMENSION TITLE(15)                                               SUM    017
C                                                                       SUM    018
      INTEGER ERROR,PAGE,CASE                                           SUM    019
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,       SUM    020
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                            SUM    021
      REAL  MACH,MAC                                                    SUM    022
C     DATA TITLE9/4HFORC/                                               SUM    023I
      DATA TITLE9/10HFORCE DATA/                                        SUM    024C
      DATA BLANK/4H    /                                                SUM    025
C                                                                       SUM    026
      DO 5 I = 1,15                                                     SUM    027
    5 TITLE(I) = BLANK                                                  SUM    028
      CASE = 0                                                          SUM    029
C                                                                       SUM    030
C   READ MASTER TABLE OF CONTENTS FOR UNIT 9                            SUM    031
      IG9 = 1                                                           SUM    032
      CALL READMS (9,E1,8,IG9)                                          SUM    033C
C     READ (9'IG9) E1                                                   SUM    034I
C    NUMBER OF COMPONENTS ON UNIT                                       SUM    035
      NTOT = E1(1)                                                      SUM    036
C    NUMBER OF SUMMATIONS ON UNIT                                       SUM    037
      NSUM = E1(2)                                                      SUM    038
C    NEXT EMPTY RECORD                                                  SUM    039
      NEXT = E1(3)                                                      SUM    040
C   CHECK IF UNIT HAS VALID DATA                                        SUM    041
      IF (E1(4) .EQ. TITLE9) GO TO 20                                   SUM    042
      WRITE (TAPEOT,10)                                                 SUM    043
   10  FORMAT (1H0,47H***UNIT 9 CALLED BY ROUTINE SUM HAS NEVER BEEN ,  SUM    044
     1   27HINITIALIZED.  PROGRAM STOP.    )                            SUM    045
```

```
      STOP
C
   20 CONTINUE
C
C
C     POINTERS TO FORCE DATA
      IG9 = 2
      CALL READMS (9,E5,41,IG9)
C     READ (9'IG9) E5
      DO 30 I=2,41
   30  LCOM(I-1) = E5(I)
      IFTYPE = E5(1)
C
      IG9 = 3
      CALL READMS (9,E6,41,IG9)
C     READ (9'IG9) E6
      DO 40 I=2,41
   40  LSUM(I-1) = E6(I)
      ISTYPE = E6(1)
      DO 41 I=1,11
   41  E7(I) = 0.0
C
      IG9 = 4
      CALL READMS (9,E8,14,IG9)
C     READ  (9'IG9) E8
      MACH = E8(1)
      V    = E8(2)
      RENO = E8(3)
      ALT  = E8(4)
      PSTAG= E8(5)
      TSTAG= E8(6)
      GTYPE(1) = E8(7)
      GTYPE(2) = E8(8)
      SREF = E8(9)
      SPAN = E8(10)
      MAC  = E8(11)
      XCG  = E8(12)
      YCG  = E8(13)
      ZCG = E8(14)
      IG9 = 5
      CALL READMS(9,E5,41,IG9)
C     READ (9'IG9) E5
      DO 43 I = 1,41
   43 LCOM(40+I) = E5(I)
C
```

```
C   READ SUMMATION CONTROL CARD                                          SUM    091
C        ISUM = 0 (DO NOT SUM)                                           SUM    092
C             = 1 (SUM COMPONENTS)                                       SUM    093
C      IPUNCH= 0 (DO NOT PUNCH)                                          SUM    094
C             =1 (PUNCH SUMMATIONS ONLY)                                 SUM    095
C             = 2 (PUNCH COMPONENTS PLUS SUMMATIONS)                     SUM    096
C      IPRINT= 0 (DO NOT PRINT COMPONENTS, ONLY PRINT FINAL SUMMATION)   SUM    097
C             = 1 (PRINT COMPONENTS PLUS SUMMATION)                      SUM    098
   50 READ (TAPEIN,55) LAST,ISUM,ISAVE,IPRINT,IPUNCH,IRUN,ICOMP         SUM    099
   55  FORMAT (5I1,I4,20I2/ 30I2)                                        SUM    100
C                                                                       SUM    101
      IF (ISUM .GT. 0) NSUM = NSUM + 1                                  SUM    102
      IF (ISUM .GT. 0) LSUM(NSUM) = NEXT                                SUM    103
      NEXTS = NEXT                                                      SUM    104
      NSUMS = NSUM                                                      SUM    105
      NEXT = NEXT + 2                                                   SUM    106
C                                                                       SUM    107
C   OBTAIN ALPHA-BETA ARRAY FOR FIRST COMPONENT TO BE READ              SUM    108
      L = ICOMP(1)                                                      SUM    109
      IG9 = LCOM(L) + 1                                                 SUM    110
      IF (ISUM .EQ. 2)  IG9 = LSUM(L) + 1                               SUM    111
      CALL READMS (9,E4,41,IG9)                                         SUM    112C
C     READ (9'IG9) E4                                                   SUM    113I
      NAB = E4(1)                                                       SUM    114
C                                                                       SUM    115
C   DO LOOP FOR COMPONENTS SELECTED                                     SUM    116
      K = 0                                                             SUM    117
   58 K = K + 1                                                         SUM    118
C     COMPONENT NUMBER                                                  SUM    119
       L =  ICOMP(K)                                                    SUM    120
      IF (L .EQ. 0) GO TO 155                                           SUM    121
      NCT = K                                                           SUM    122
      IG9 = LCOM(L)                                                     SUM    123
      IF (ISUM .EQ. 2)  IG9 = LSUM(L)                                   SUM    124
      CALL READMS (9,E2,11,IG9)                                         SUM    125C
C     READ (9'IG9) E2                                                   SUM    126I
       LL = E2(1)                                                       SUM    127
      IF (LL .EQ. L) GO TO 100                                          SUM    128
      WRITE (TAPEOT,60)                                                 SUM    129
   60  FORMAT (1H0,45H***COMPONENT NUMBER REQUESTED DOES NOT MATCH ,    SUM    130
     1          47HNUMBER ON UNIT 9 IN ROUTINE SUM. PROGRAM STOP.  )    SUM    131
      STOP                                                              SUM    132
C    RECALL PANEL NUMBERS                                               SUM    133
  100 DO 110 I=1,10                                                     SUM    134
       IPAN9(I,K) = E2(I+1)                                             SUM    135
```

```
  110 CONTINUE                                                          SUM    136
C                                                                       SUM    137
C  PRINT OR PUNCH COMPONENTS AS REQUIRED                                SUM    138
      IPR = 0                                                           SUM    139
      IF (IPRINT.EQ.0 .AND. IPUNCH.NE.2) GC TC 150                      SUM    140
      IF (IPUNCH .EQ. 2) WRITE (TAPEG,121) L                            SUM    141
  121  FORMAT (26HCOMPONENT DATA SET NUMBER ,I3)                        SUM    142
      IF (IPRINT .EQ. 1) GO TC 195                                      SUM    143
  122 DO 124 I=1,NAB                                                    SUM    144
      IG9 = LCOM(L) + I*2                                               SUM    145
      IF (ISUM .EQ. 2)  IG9 = LSUM(L) + I*2                             SUM    146
      CALL READMS (9,F,11,IG9)                                          SUM    147C
C     READ (9'IG9) F                                                    SUM    148I
      IF (IPRINT .EQ. 1) WRITE (TAPEOT,290) F                           SUM    149
      IF (IPUNCH .EQ. 2) WRITE (TAPEG,123) F(1),F(6),F(4),F(9),F(5),    SUM    150
     1  F(11),F(10),F(7),IRUN                                           SUM    151
  123  FORMAT (7F9.4,F7.2,1X,I4)                                        SUM    152
  124 CONTINUE                                                          SUM    153
      IF (IPUNCH .EQ. 2) WRITE (TAPEOT,125)                             SUM    154
  125  FORMAT (1H ,5X,33HCOMPCNENT DATA HAVE BEEN PUNCHED.  )           SUM    155
  150 IF (K .LT. 50) GO TO 58                                           SUM    156
  155 CONTINUE                                                          SUM    157
      IPR = 1                                                           SUM    158
      IF (ISUM .EQ. 0) GO TC 440                                        SUM    159
C                                                                       SUM    160
C                                                                       SUM    161
C  DO LOOP FOR ALL ALPHA-BETA VALUES                                    SUM    162
      DO 190 J=1,NAB                                                    SUM    163
      ALPHAR =  E4(J+1)*1.745329252E-2                                  SUM    164
      BETAR  = E4(J+21)*1.745329252E-2                                  SUM    165
      SINA = SIN(ALPHAR)                                                SUM    166
      COSA = COS(ALPHAR)                                                SUM    167
      SINB = SIN(BETAR)                                                 SUM    168
      COSB = COS(BETAR)                                                 SUM    169
C   ZERO OUT FORCE ARRAY                                                SUM    170
      DO 160 I=1,11                                                     SUM    171
  160  F(I) = 0.0                                                       SUM    172
C                                                                       SUM    173
      DO 180 K=1,NCT                                                    SUM    174
       L = ICOMP(K)                                                     SUM    175
      IG9 = LCOM(L) + J*2                                               SUM    176
      IF (ISUM .EQ. 2)  IG9 = LSUM(L) + J*2                             SUM    177
      CALL READMS (9,FF,11,IG9)                                         SUM    178C
C     READ (9'IG9) FF                                                   SUM    179I
      IF (IBETA .NE. 0)  GO TO 165                                      SUM    180
```

```
      IF (ISUM .EQ. 2)  GO TO 165                                       SUM    181
C                                                                       SUM    182
      FF(4) = 2.*FF(4)                                                  SUM    183
      FF(5) = 0.0                                                       SUM    184
      FF(6) = 2.*FF(6)                                                  SUM    185
      FF(9) = 2.*FF(9)                                                  SUM    186
      FF(10)  = -FF(6)*YCG/SPAN                                         SUM    187
      FF(11)  =  FF(4)*YCG/SPAN                                         SUM    188
      FF(2) = FF(4)*COSA*COSB + FF(6)*SINA*COSB                         SUM    189
      FF(3) =-FF(4)*SINA      + FF(6)*COSA                              SUM    190
  165 CONTINUE                                                          SUM    191
C                                                                       SUM    192
      DO 170 I=2,11                                                     SUM    193
       F(I) = F(I) + FF(I)                                              SUM    194
  170  CONTINUE                                                         SUM    195
  180  CONTINUE                                                         SUM    196
C                                                                       SUM    197
      IF (F(2) .EQ. 0.0) F(2) = 0.000001                                SUM    198
      F(8) = F(3) / F(2)                                                SUM    199
      F(1) = FF(1)                                                      SUM    200
      F(7) = FF(7)                                                      SUM    201
C                                                                       SUM    202
      CALL WRITMS (9,F,11,NEXT)                                         SUM    203C
C     WRITE (9'NEXT) F                                                  SUM    204I
      NEXT = NEXT + 1                                                   SUM    205
      CALL WRITMS (9,E7,11,NEXT)                                        SUM    206C
C     WRITE (9'NEXT) E7                                                 SUM    207I
      NEXT = NEXT + 1                                                   SUM    208
  190 CONTINUE                                                          SUM    209
C                                                                       SUM    210
C                                                                       SUM    211
C                                                                       SUM    212
C***OUTPUT DATA ******                                                  SUM    213
C  WRITE HEADER INFORMATION FOR FORCE OUTPUT PAGE                       SUM    214
  195 CALL HEADER                                                       SUM    215
      IF (IPR .NE. 0)  GO TO 201                                        SUM    216
      IF (ISUM .EQ. 2)  GO TO 197                                       SUM    217
      WRITE(TAPEOT,198)  L                                              SUM    218
  198  FORMAT (1H0,26HCOMPONENT DATA SET NUMBER ,I3)                    SUM    219
      GO TO 204                                                         SUM    220
  197 WRITE(TAPEOT,199)  L                                              SUM    221
  199 FORMAT(1H0,26HPREVIOUS SUMMATION NUMBER , I3)                     SUM    222
      GO TO 204                                                         SUM    223
  201 CONTINUE                                                          SUM    224
      WRITE (TAPEOT,200) NSUM                                           SUM    225
```

```
  200  FORMAT (1H0,26HSUMMATICN DATA SET NUMBER ,I3)                    SUM    226
       DO 202 I=1,NCT                                                   SUM    227
       WRITE (TAPEOT,203) (IPAN9(II,I),II=1,10)                         SUM    228
  202 CONTINUE                                                          SUM    229
  203  FORMAT (1H ,5X,10I4)                                             SUM    230
  204 WRITE (TAPEOT,230) MACH,V,RENO                                    SUM    231
  230 FORMAT (1H0,7H  MACH=F8.3,6H  VEL=F9.1,16H FT/SEC  RE/FT =E13.5 ) SUM    232
       IF (PSTAG .LT. 0.00001) WRITE (6,240) ALT                        SUM    233
  240 FORMAT (1H ,7H  ALT =F8.0 )                                       SUM    234
       IF (PSTAG .GT. 0.00001) WRITE (6,250) PSTAG,TSTAG                SUM    235
  250 FORMAT (1H ,16X7HP STAG=F7.1,16H ATMCS   T STAG=F7.1,6H DEG F )   SUM    236
       WRITE (TAPEOT,260) GTYPE                                         SUM    237
  260  FORMAT (1H ,2X,2A4)                                              SUM    238
       WRITE (TAPEOT,270) SREF,SPAN,MAC,XCG,YCG,ZCG                     SUM    239
  270 FORMAT (1H0,9H  S REF =F9.2,8H  SPAN =F8.2,8H  MAC  =F8.2,/1H ,   SUM    240
     1  9H  X CG  =F9.2,8H  Y CG =F8.2,8H  Z CG =F8.2 )                 SUM    241
       WRITE (TAPEOT,280)                                               SUM    242
  280  FORMAT (1H0,10HFORCE DATA,/1H ,7H  ALPHA,4X,3HC D,7X,3HC L,7X,   SUM    243
     1 3HC A,7X,3HC Y,7X,3HC N,/1H ,7H  BETA ,4X,3HL/D,7X,3HC M,7X,     SUM    244
     2 4HC LL,6X,4HC LN)                                                SUM    245
       IF (IPR .EQ. 0) GO TO 122                                        SUM    246
       IF (IPUNCH .NE. 0) WRITE (TAPEG,285) NSUM                        SUM    247
  285  FORMAT (27HSUMMATION COMPONENT NUMBER ,I3)                       SUM    248
C                                                                       SUM    249
C  RETRIEVE OUTPUT DATA FROM UNIT 9 AND PRINT                           SUM    250
       DO 300 I=1,NAB                                                   SUM    251
        IG9 = LSUM(NSUM) + I*2                                          SUM    252
       CALL READMS (9,F,11,IG9)                                         SUM    253C
C      READ (9'IG9) F                                                   SUM    254I
C  WRITE OUTPUT FORCE DATA                                              SUM    255
       WRITE (TAPEOT,290) F                                             SUM    256
  290  FORMAT (1H0,F7.2,5F10.5,/1H ,F7.2,4F10.5)                        SUM    257
       IF (IPUNCH .NE. 0) WRITE (TAPEG,123) F(1),F(6),F(4),F(9),F(5),   SUM    258
     1  F(11),F(10),F(7),IRUN                                           SUM    259
  300 CONTINUE                                                          SUM    260
       IF (IPUNCH .NE. 0) WRITE (TAPEOT,302)                            SUM    261
  302  FORMAT (1H ,5X,25HTHESE DATA WHERE PUNCHED.  )                   SUM    262
C**OUTPUT DATA COMPLETED FOR THIS COMPONENT                             SUM    263
C                                                                       SUM    264
       IF (ISAVE .NE. 0) NEXT = NEXTS                                   SUM    265
       IF (ISAVE .NE. 0) NSUM = NSUMS                                   SUM    266
C                                                                       SUM    267
C  SAVE COMPONENT INFORMATION IN TABLE OF CONTENTS                      SUM    268
       IF (ISAVE .NE. 0) GO TO 440                                      SUM    269
       DO 191 I=2,11                                                    SUM    270
```

```
  191  E2(I) = ICOMP(I-1)
       E2(1) = NSUM
       IG9 = LSUM(NSUM)
       CALL WRITMS (9,E2,11,IG9)
C      WRITE (9'IG9) E2
       IG9 = IG9 + 1
       CALL WRITMS (9,E4,41,IG9)
C      WRITE (9'IG9) E4
C
C  UPDATE MAIN TABLE OF CONTENTS
       E1(1) = NTOT
       E1(2) = NSUM
       E1(3) = NEXT
       IG9 = 1
       CALL WRITMS (9,E1,8,IG9)
C      WRITE (9'IG9) E1
       DO 310 I=2,41
  310  E6(I) = LSUM(I-1)
       E6(1) = 0.0
       IG9 = 3
       CALL WRITMS (9,E6,41,IG9)
C      WRITE (9'IG9) E6
C
C
       WRITE (TAPEOT,430) NSUM
  430  FORMAT (1H0,10X,37HDATA SAVED ON UNIT 9.  SUMMATION SET ,
     1    8HNUMBER =,I4)
C
C
C  CHECK IF LAST CASE HAS BEEN COMPLETED
  440 IF (LAST .EQ. 0) GO TO 50
C
       RETURN
       END
```

```
      SUBROUTINE  SURF                                                  SURF  001
C                                                                       SURF  002
C*****THIS SUBROUTINE DEFINES A GENERAL AIRFOIL SURFACE                 SURF  003
C     IN THE BODY-AXIS COORDINATE SYSTEM.  THE SURFACE IS               SURF  004
C     GENERATED ACCORDING TO THE NUMBER OF AIRFOIL SECTIONS, NAF,       SURF  005
C     THE ORIENTATION OF THE CHORD LINE (YAW-PITCH-ROLL SEQUENCE),      SURF  006
C     THE MEAN LINE DISTRIBUTION, AND THE THICKNESS DISTRIBUTION.       SURF  007
C     THE SECTION COORDINATES MAY DEFINED AS AN NACA TYPE               SURF  008
C     (THICKNESS DISTRIBUTION NORMAL TO MEAN LINE) OR AS A              SURF  009
C     SHEARED TYPE (THICKNESS NORMAL TO THE CHORD LINE).                SURF  010
C                                                                       SURF  011
C     AN OPTION TO CALCULATE THE TIP-PLANE SURFACE OR THE ROOT-         SURF  012
C     PLANE SURFACE, OR BOTH, IS ALSO AVAILABLE.                        SURF  013
C     ALL INPUT QUANTITIES ENTER THROUGH THE LABELED COMMON/AIRFOIL/.   SURF  014
C                                                                       SURF  015
C     NAF      - THE NUMBER OF AIRFOIL SECTIONS                         SURF  016
C     NAFORD   - THE NUMBER OF SECTION ORDINATES                        SURF  017
C     NCAM     - =1, CAMBER DATA INPUT                                  SURF  018
C     NACA     - =1, NACA TYPE  SECTION                                 SURF  019
C     XOC      - ARRAY OF CHORD STATIONS, PERCENT CHORD                 SURF  020
C     AFORG    - ARRAY OF CHORD ORIGIN AND ORIENTATION DATA             SURF  021
C     AFCAM    - ARRAY OF CAMBER DISTRIBUTION, PERCENT CHORD            SURF  022
C     AFORD    - ARRAY OF THICKNESS DISTRIBUTION, PERCENT CHORD         SURF  023
C     SURFID   - SURFACE IDENTIFICATION (CC'S 73-76 ON OUTPUT  CARDS)   SURF  024
C     ITIP     - = 1, CALCULATE TIP SURFACE.                            SURF  025
C     IROOT    - = 1, CALCULATE ROOT SURFACE.                           SURF  026
C*****                                                                  SURF  027
      DIMENSION XOC(30),AFORG(20,7),AFCAM(20,30),AFORD(20,30),          SURF  028
     1          X(2),Y(2),Z(2),CDELTA(30),SDELTA(30)                    SURF  029
C                                                                       SURF  030
      COMMON /ACFT/ IPRINT,ITAPE,ITYPE,PI,DX,CASE                       SURF  031
C                                                                       SURF  032
      COMMON /AIRFOIL/ NAF,NAFORD,NCAM,NACA,ITIP,IROOT,                 SURF  033
     1                 XOC,AFORG,AFCAM,AFORD,SURFID                     SURF  034
     2                ,ISURF,IFLAP,NHINGE                               SURF  035
      COMMON /TAPE/TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,   SURF  036
     1             TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                        SURF  037
C                                                                       SURF  038
      INTEGER STAT1,STAT2                                               SURF  039
      INTEGER  TAPEIN,TAPEOT,TAPEA,TAPEB,TAPEC,TAPED,TAPEE,TAPEF,       SURF  040
     1         TAPEG,TAPEH,TAPEI,TAPEJ,TAPEK                            SURF  041
      INTEGER CASE                                                      SURF  042
C                                                                       SURF  043
C  DESCRIPTION OF COUNTERS AND CONTROLS                                 SURF  044
C     I IS THE AIRFOIL SECTION NUMBER                                   SURF  045
```

```
C          I = 1  IS THE ROOT                                          SURF  046
C          I = NAF IS THE TIP                                          SURF  047
C     J IS THE CHORDWISE STATION NUMBER                                SURF  048
C          J = 1  IS THE LEADING EDGE                                  SURF  049
C          J = NAFCRD IS THE TRAILING EDGE                             SURF  050
C     K IS THE POINT COUNTER FOR TYPE 3 CARDS (TWO POINTS PER CARD).   SURF  051
C     IEND IS CONTROL FLAG FOR TIP OR ROOT SURFACE                     SURF  052
C     INEXT IS  SECTION SEQUENCE INCREMENT                             SURF  053
C          INEXT = +1 FOR UPPER SURFACE                                SURF  054
C          INEXT = -1 FOR LOWER SURFACE                                SURF  055
C     ISEQ IS COUNTER FOR OUTPUT DATA (CC'S 77-80)                     SURF  056
C                                                                      SURF  057
C                                                                      SURF  058
C*****THE SURFACE IS FIRST TESTED FOR POSSIBLE SYMMETRY                SURF  059
C     ABOUT THE X-Z ORIGIN PLANE.  IF SO, THE SURFACE                  SURF  060
C     SYMMETRY FLAG (ISYMM) IS SET TO +1 OR -1 AS PER SIGN OF PHI.     SURF  061
C     TO BE SYMMETRICAL THREE CONDITIONS MUST BE MET                   SURF  062
C     1.  NO CAMBER DISTRIBUTION                                       SURF  063
C     2.  THE CHORD LINE IS IN THE X-Z ORIGIN PLANE                    SURF  064
C     3.  THE ROLL ANGLE PHI IS + OR - 90.0 DEGREES                    SURF  065
C     THE TEST IS MADE ON BOTH THE ROOT AND TIP SECTIONS               SURF  066
C                                                                      SURF  067
      I = 0                                                            SURF  068
      ISEQ = 0                                                         SURF  069
      IEND = 0                                                         SURF  070
      ISYMM = 0                                                        SURF  071
      IHNGE = 1000                                                     SURF  072
      IF (ISURF .NE. 0) GO TO 230                                      SURF  073
      IF (NCAM .EQ. 1) GO TO 10                                        SURF  074
      IF (AFORG(1,2) .GT. 0.0001) GO TO 10                             SURF  075
      IF (AFORG(NAF,2) .GT. 0.0001) GO TO 10                           SURF  076
      IF (COS(AFORG(1,7)*PI/180.) .GT. 0.0001) GO TO 10                SURF  077
      IF (COS(AFORG(NAF,7)*PI/180.) .GT. 0.0001) GO TO 10              SURF  078
C                                                                      SURF  079
C     SURFACE IS SYMMETRICAL ABOUT X-Z ORIGIN PLANE.                   SURF  080
      ISYMM =-1                                                        SURF  081
C     CHECK IF PHI IS POSITIVE OR NEGATIVE.                            SURF  082
      IF (AFORG(1,7) .LT. 0.0) GO TO 10                                SURF  083
C                                                                      SURF  084
C     POSITIVE VERTICAL, DO LOWER SURFACE FIRST.                       SURF  085
      ISYMM = 1                                                        SURF  086
      I = NAF + 1                                                      SURF  087
      GO TO 200                                                        SURF  088
C                                                                      SURF  089
C                                                                      SURF  090
```

```
   10 CONTINUE
      INEXT = 1
      DZ = 1.0
      DETA = 1.0
C
   20 K = 0
      STAT1 = 2
      STAT2 = 0
   30 I = I + INEXT
      JP = 1
      IF (I .GT. NAF) GO TO 150
      IF (I .EQ. 0)   GO TO 150
C
C*****CALCULATE CHORD LENGTH C  AND AIRFOIL ORIENTATION,
C     SINE AND COSINE VALUES OF
C         YAW ANGLE   -  SPSI, CPSI
C         PITCH ANGLE -  STHETA, CTHETA
C         ROLL ANGLE  -  SPHI, CPHI
C
      DXX =-(AFORG(I,4) - AFORG(I,1))*DX
      DYY =  AFORG(I,2) - AFORG(I,5)
      DZZ =  AFORG(I,6) - AFORG(I,3)
      DS1 = DXX**2 + DYY**2
      DS2 = SQRT(DS1 + DZZ**2)
      DS1 = SQRT(DS1)
C
      SPSI = DYY/DS1
      CPSI = DXX/DS1
C
      STHETA = DZZ/DS2
      CTHETA = DS1/DS2
C
      SPHI = SIN(AFORG(I,7)*PI/180.)
      CPHI = COS(AFORG(I,7)*PI/180.)
C
      C = 0.01*DS2
C
C***  PRESET MEAN LINE SLOPES
      DO 40 J = 1,NAFORD
      CDELTA(J) = 1.0
   40 SDELTA(J) = 0.0
C
C***  CHECK IF CAMBER DATA INPUT
      IF (NCAM .NE. 1) GO TO 70
C
```

```
      IF (NACA .NE. 2)  GO TO 46
      IF (C .EQ. 0.0)   GO TO 46
      CI = 1.0/C
      DO 45 J = 1,NAFORD
   45 AFCAM(I,J) = AFCAM(I,J)*CI
   46 CONTINUE
C
C***  CHECK AIRFOIL SECTION TYPE
      IF (NACA .NE. 1) GO TO 70
C
C***  NACA TYPE, THICKNESS NORMAL TO MEAN LINE.
C     MUST FIRST CALCULATE SLOPE ALONG MEAN LINE.
C     USE NEWTONIAN SINGLE QUADRATIC
C
      J1 = 0
      DO 60 IP = 1,NAFORD
C
      IF (IP .EQ. 2 .OR. IP .EQ. NAFORD) GO TO 50
C
      J1 = J1 + 1
      J2 = J1 + 1
      J3 = J1 + 2
      X1 = XOC(J1)
      X2 = XOC(J2)
      X3 = XOC(J3)
      Y1 = AFCAM(I,J1)
      Y2 = AFCAM(I,J2)
      Y3 = AFCAM(I,J3)
C
   50 DYDX = (Y2-Y1)/(X2-X1) + (2.0*XOC(IP) - X2 - X1)*
     1       ((Y3-Y2)/(X3-X2) - (Y2-Y1)/(X2-X1))/(X3-X1)
C
      CDELTA(IP) = 1.0/SQRT(1.0 + DYDX**2)
      SDELTA(IP) = DYDX*CDELTA(IP)
C
   60 CONTINUE
C
C
C***  BEGIN SURFACE CALCULATION
   70 J = 0
      IF (K .EQ. 1) GO TO 90
   80 K = 0
   90 J = J + 1
      IF (J .LE. NAFORD) GO TO 100
C
```

```
C***  SET DUMMY POINT AND WRITE DATA.
      K = 2
      X(K) = X(1)
      Y(K) = Y(1)
      Z(K) = Z(1)
      STAT2 = STAT1
      GO TO (110,260), JP
C
C***  CALCULATE COORDINATES
  100 K = K + 1
      XI = (XOC(J) - AFORD(I,J)*SDELTA(J)*DZ)*C
      ETA = (AFCAM(I,J) + DZ*AFORD(I,J)*CDELTA(J))*C
      X(K) = DX*AFORG(I,1) - XI*CTHETA*CPSI
     1       + DETA*ETA*(STHETA*CPSI*CPHI + SPSI*SPHI)
      Y(K) = AFORG(I,2) - XI*CTHETA*SPSI
     1       + DETA*ETA*(STHETA*SPSI*CPHI - CPSI*SPHI)
      Z(K) = AFORG(I,3) + XI*STHETA + DETA*ETA*CTHETA*CPHI
C
      IF (K .EQ. 1) GO TO 90
C
      GO TO (110,260), JP
C***  WRITE DATA
  110 ISEQ = ISEQ + 1
      WRITE(ITAPE,120) X(1),Y(1),Z(1),STAT1,
     1                 X(2),Y(2),Z(2),STAT2,CASE,ITYPE,SURFID,ISEQ
C
  120 FORMAT(3F10.4,I1,3F10.4,I1,2XI2,5XI1,A4,I4)
C
      IF (IPRINT .EQ. 1)
     1WRITE (TAPEOT,130) X(1),Y(1),Z(1),STAT1,
     2                X(2),Y(2),Z(2),STAT2, CASE,ITYPE,SURFID,ISEQ
C
  130 FORMAT(1H ,3F10.4,I1,3F10.4,I1,2XI2,5XI1,A4,I4)
C
C***  CHECK IF THIS IS TO HAVE A CONTROL SURFACE.
      IF ( J    .EQ. IHNGE) GO TO 250
C***  CHECK STATUS FLAGS
      IF (STAT1 .GT. 0) STAT1 = 0
      IF (STAT2 .GT. 0) STAT2 = 0
C
C***  CHECK FOR PROPER CYCLING
      IF (J .LT. NAFORD) GO TO 80
C
C
C***  HAVE COMPLETED AIRFOIL SECTION, SET STATUS AND GO TO NEXT ONE.
```

```
  140 STATL = 1                                                         SURF  226
      GO TO 30                                                          SURF  227
C                                                                       SURF  228
C***  HAVE COMPLETED SURFACE, CHECK WHICH ONE.                          SURF  229
  150 CONTINUE                                                          SURF  230
      IF (NACA .EQ. 2)  NACA = 0                                        SURF  231
      IF (IFLAP .EQ. 1) GO TO 270                                       SURF  232
      IF (INEXT .LT. 0) GO TO 160                                       SURF  233
C                                                                       SURF  234
C***  JUST COMPLETED UPPER SURFACE.                                     SURF  235
      IF (IEND .EQ. 1) GO TO 190                                        SURF  236
      IF (ITIP .EQ. 1) GO TO 180                                        SURF  237
      IF (ISYMM .EQ.  0) GO TO 200                                      SURF  238
C                                                                       SURF  239
C***  JUST COMPLETED LOWER SURFACE.                                     SURF  240
  160 IF (IEND - 1 )170,190,220                                         SURF  241
  170 IF (ITIP .EQ. 1) GO TO 180                                        SURF  242
      IF (IROOT .EQ. 1) GO TO 210                                       SURF  243
      RETURN                                                            SURF  244
C                                                                       SURF  245
C                                                                       SURF  246
C                                                                       SURF  247
C***  TIP SURFACE                                                       SURF  248
  180 STAT1 = 2                                                         SURF  249
      STAT2 = 0                                                         SURF  250
      IEND = 1                                                          SURF  251
      INEXT = +1                                                        SURF  252
      I = NAF                                                           SURF  253
      IF (ISYMM .NE. 1) GO TO 70                                        SURF  254
      DETA = 0.0                                                        SURF  255
      I = I -1                                                          SURF  256
      GO TO 30                                                          SURF  257
  190 DETA = 1.0                                                        SURF  258
      IF (ISYMM .EQ. -1) DETA = 0.0                                     SURF  259
      IF (ISYMM .EQ. 0) DZ = -DZ                                        SURF  260
      ITIP = 0                                                          SURF  261
      IEND = 0                                                          SURF  262
      I = NAF                                                           SURF  263
      J = 0                                                             SURF  264
      GO TO 80                                                          SURF  265
C                                                                       SURF  266
C                                                                       SURF  267
C***  LOWER SURFACE.                                                    SURF  268
  200 INEXT = -1                                                        SURF  269
      DZ = -1.0                                                         SURF  270
```

```
      DETA = 1.0
      GO TO 20
C
C
C***  ROOT SURFACE
  210 STAT1 = 2
      IEND = 2
      INEXT = -1
      I = 1
      DETA = 1.0
      IF (ISYMM .EQ.  0) GO TO 70
      I = 2
      IF (ISYMM .EQ. 1) GO TO 30
      DETA = 0.0
      GO TO 30
  220 DETA = 1.0
      IF (ISYMM .EQ. 1) DETA = 0.0
      IF (ISYMM .EQ. 0) DZ = -DZ
      IROOT = 0
      IEND = 0
      I = 1
      J = 0
      GO TO 80
C
C
C
C
  230 ISYMM = -ISURF
C
C***  ONLY UPPER OR LOWER SURFACE, ZERO OUT TIP AND ROOT SECTIONS.
      ITIP = 0
      IROOT = 0
C
C***  CHECK IF THIS SURFACE TO HAVE A FLAP (OR CONTROL SURFACE).
      IF (IFLAP .EQ. 0)  GO TO 240
C
C***  CONTROL SURFACE.  SET HINGE LINE LOCATION.
      IHNGE = NHNGE
      REWIND TAPEB
  240 IF (ISYMM .LT. 0) GO TO 10
      I = NAF + 1
      GO TO 200
C
C
C
```

```
  250 J = J - 1                                                         SURF  316
      JP = 2                                                            SURF  317
      STAT1 = 1                                                         SURF  318
      GO TO 80                                                          SURF  319
C                                                                       SURF  320
C                                                                       SURF  321
  260 WRITE (TAPEB) X,Y,Z,STAT1,STAT2                                   SURF  322
      STAT1 = 0                                                         SURF  323
      STAT2 = 0                                                         SURF  324
      IF (J .LT. NAFORD) GO TO 80                                       SURF  325
C                                                                       SURF  326
C***  HAVE COMPLETED AIFOIL SECTION                                     SURF  327
      STAT1 = 1                                                         SURF  328
      GO TO 30                                                          SURF  329
C                                                                       SURF  330
C***  HAVE COMPLETED AIRFOIL. TRANSFER FLAP ELEMENTS FROM               SURF  331
C     UNIT 3  TO UNIT ITAPE. GIVE FIRST POINT STATUS 2                  SURF  332
C     AND LAST POINT STATUS 3.                                          SURF  333
C                                                                       SURF  334
  270 CONTINUE                                                          SURF  335
      REWIND TAPEB                                                      SURF  336
      ITOTAL = NAF*(NAFORD + 1)/2                                       SURF  337
      READ (11) X,Y,Z,STAT1,STAT2                                       SURF  338
      STAT1 = 2                                                         SURF  339
  280 ISEQ = ISEQ + 1                                                   SURF  340
      IF (ISEQ .EQ. ITOTAL) STAT2 = 3                                   SURF  341
      WRITE (ITAPE,120) X(1),Y(1),Z(1),STAT1,                           SURF  342
     1                  X(2),Y(2),Z(2),STAT2,CASE,ITYPE,SURFID,ISEQ     SURF  343
C                                                                       SURF  344
      IF (IPRINT .EQ. 1)                                                SURF  345
     1WRITE (TAPEOT,130) X(1),Y(1),Z(1),STAT1,                          SURF  346
     2                  X(2),Y(2),Z(2),STAT2,CASE,ITYPE,SURFID,ISEQ     SURF  347
C                                                                       SURF  348
      IF (ISEQ .EQ. ITOTAL) RETURN                                      SURF  349
      READ (TAPEB) X,Y,Z,STAT1,STAT2                                    SURF  350
      GO TO 280                                                         SURF  351
C                                                                       SURF  352
C                                                                       SURF  353
      END                                                               SURF  354
```

```
      SUBROUTINE VALU2(N,X,Y,Z,XN,YN,ZN,IC,IT,INT,EMT,EMX,EMY,EMZ,POP,
     1     TOT)
C
      COMMON/SURFN2/INORM,ISURF,IX,IY,IZ,U1,UL,V1,VL,CR,CT,CHX,N2,N3,
     1     XO,YO,ZO,AP11,AP12,AP13,AP21,AP22,AP23,AP31,AP32,AP33,
     2  FLOWC(7),XI(500),YI(500),B(503,7),XKF(503),AA(5999),
     A  XB(4), YB(4), ZB(4),
     3  NDSET,IABSET,IR,LORG(20),ISR(20),NS,ISFR,IVIS, IFTYP,IFLOW,
     4     BX(2,20), BY(2,20), NB
C
      DIMENSION X(N),Y(N),Z(N),INT(N),EMT(N),EMX(N),EMY(N),EMZ(N),
     1     POP(N),TOT(N),VT(6),XT(6),FI(7), BP(2)
C
C
      DO 500 I = 1,N
      IF (IC .EQ. 1)  GO TO 70
C
C  TRANSFORM COORDS
      XX = X(I) - XO
      YY = Y(I) - YO
      ZZ = Z(I) - ZO
C
      XT(1) = XX*AP11 + YY*AP12 + ZZ*AP13
      XT(2) = XX*AP21 + YY*AP22 + ZZ*AP23
      XT(3) = XX*AP31 + YY*AP32 + ZZ*AP33
      XT(4)  =  XT(1)
      XT(5)  =  SQRT(XT(2)**2 + XT(3)**2)
      IF (XT(2) .EQ. 0.0)  GO TO 40
      XT(6)  =  ATAN2(XT(2),-XT(3))
      IF(XT(6).LT.0.0) XT(6) = 6.283185308 + XT(6)
      GO TO 50
   40 XT(6) = 0.0
      IF (XT(3) .GT. 0.0)  XT(6) = 3.141592654
   50 CONTINUE
C
C  NORMALIZE COORDS
      YN = (XT(IY) - V1)/VL
      IF (ISURF .EQ. 1)  GO TO 60
      XN = (XT(IX) - U1)/UL
      GO TO 70
   60 CHX = CR + (CT-CR)*YN
      XLE = U1 + (UL-U1)*YN
      XN = (XT(IX) - XLE)/CHX
C
C  CHECK IF WITHIN SURFACE BOUNDARIES
```

```
   70 IF (XN .LT. C.C)  GO TO 80                                    VAL2  046
      IF (XN .GT. 1.C)  GO TO 80                                    VAL2  047
      IF (YN .LT. C.C)  GO TO 80                                    VAL2  048
      IF (YN .GT. 1.C)  GO TO 80                                    VAL2  049
C                                                                   VAL2  050
C  CHECK IF WITHIN FLOW BOUNDARIES                                  VAL2  051
      IB = 0                                                        VAL2  052
  800 IB = IB + 1                                                   VAL2  053
      DO 810 I1 = 1,NB                                              VAL2  054
      J = I1                                                        VAL2  055
      IF (YN .LT. BY(IB,I1))  GO TO 820                             VAL2  056
  810 CONTINUE                                                      VAL2  057
C                                                                   VAL2  058
  820 IF (J .EQ. 1)  J = 2                                          VAL2  059
      DY = BY(IB,J) - BY(IB,J-1)                                    VAL2  060
      IF (DY .EQ. C.0)  GO TO 80                                    VAL2  061
      DX = BX(IB,J) - BX(IB,J-1)                                    VAL2  062
      BP(IB) = BX(IB,J-1) + (YN - BY(IB,J-1))*DX/DY                 VAL2  063
      GO TO (830,840), IB                                           VAL2  064
  830 IF (XN .LT. BP(IB))  GO TO 80                                 VAL2  065
      GO TO 800                                                     VAL2  066
  840 IF (XN .GT. BP(IB))  GO TO 80                                 VAL2  067
C                                                                   VAL2  068
C  POINT WITHIN FLOW BOUNDARIES                                     VAL2  069
      XN = XN - BP(1)                                               VAL2  070
      IF (IVIS .EQ. 0)  GO TO 850                                   VAL2  071
      XN = XN/(BX(2,1) - BX(1,1))                                   VAL2  072
      GO TO 860                                                     VAL2  073
  850 XN = XN/(BP(2) - BP(1))                                       VAL2  074
  860 CONTINUE                                                      VAL2  075
C                                                                   VAL2  076
C  POINT WITHIN BOUNDARIES.  CALCULATE VALUES.                      VAL2  077
      XKF(1) = 1.0                                                  VAL2  078
      XKF(2) = XN                                                   VAL2  079
      XKF(3) = YN                                                   VAL2  080
      J3 = 3                                                        VAL2  081
C                                                                   VAL2  082
      DO 250 J = 1,N2                                               VAL2  083
      J3 = J3 + 1                                                   VAL2  084
      UIJ = 0.0                                                     VAL2  085
      TR1 = 0.0                                                     VAL2  086
      TR2 = 0.0                                                     VAL2  087
      IF (XN .EQ. XI(J))  GO TO 210                                 VAL2  088
      TR1 = (XN - XI(J))**2                                         VAL2  089
  210 IF (YN .EQ. YI(J))  GO TO 220                                 VAL2  090
```

```
      TR2 = (YN - YI(J))**2                                             VAL2  091
  220 RBIJ = TR1 + TR2                                                  VAL2  092
      IF (RBIJ .EQ. 0.0)  GO TO 230                                     VAL2  093
      UIJ = RBIJ*ALOG(RBIJ)                                             VAL2  094
  230 XKF(J3) = UIJ                                                     VAL2  095
  250 CONTINUE                                                          VAL2  096
C                                                                       VAL2  097
C                                                                       VAL2  098
      DO 300 K = 1,7                                                    VAL2  099
      FI(K) = FLOWC(K)                                                  VAL2  100
      DO 290 J = 1,N3                                                   VAL2  101
      FI(K) = FI(K) + XKF(J)*B(J,K)                                     VAL2  102
  290 CONTINUE                                                          VAL2  103
  300 CONTINUE                                                          VAL2  104
C                                                                       VAL2  105
      INT(I) = 1                                                        VAL2  106
      EMT(I) = FI(1)                                                    VAL2  107
      FT = SQRT(FI(2)**2 + FI(3)**2 + FI(4)**2)                         VAL2  108
      IF (FT .EQ. 0.0)  FT = 1.0                                        VAL2  109
      FI(2) = FI(2)/FT                                                  VAL2  110
      FI(3) = FI(3)/FT                                                  VAL2  111
      FI(4) = FI(4)/FT                                                  VAL2  112
      EMX(I) = FI(2)/CR                                                 VAL2  113
      EMY(I) = FI(3)/VL                                                 VAL2  114
      POP(I) = FI(5)                                                    VAL2  115
      TOT(I) = FI(6)                                                    VAL2  116
      ZN = FI(7)                                                        VAL2  117
C                                                                       VAL2  118
C  TRANSFORM XN, YN, ZN  COORDS BACK TO (X, Y, Z)                       VAL2  119
      IF (IVIS .EQ. 0)  GO TO 310                                       VAL2  120
      XN = XN*(BX(2,1) - BX(1,1))                                       VAL2  121
      GO TO 320                                                         VAL2  122
  310 XN = XN*(BP(2) - BP(1))                                           VAL2  123
  320 XN = XN + BP(1)                                                   VAL2  124
      XT(IY) = YN*VL + V1                                               VAL2  125
      XT(IZ) = ZN                                                       VAL2  126
      IF (ISURF .EQ. 1)  GO TO 75                                       VAL2  127
      XT(IX) = XN*UL + U1                                               VAL2  128
      GO TO 76                                                          VAL2  129
   75 XT(IX) = XN*(CR+(CT-CR)*YN) + U1 + (UL-U1)*YN                     VAL2  130
   76 CONTINUE                                                          VAL2  131
      IF (INORM .LE. 0)  GO TO 79                                       VAL2  132
      GO TO (81, 82, 83, 84),  INORM                                    VAL2  133
C                                                                       VAL2  134
C  INORM = 0      IX =4,  IY = 5,  IZ = 6                               VAL2  135
```

```
   79 XX = XT(IX)
      SINPH = SIN(XT(IZ))
      COSPH = COS(XT(IZ))
      YY =  XT(IY)*SINPH
      ZZ = -XT(IY)*COSPH
      VT(1) = FI(2)
      VT(2) = FI(3)*SINPH + FI(4)*COSPH
      VT(3) = FI(4)*SINPH - FI(3)*COSPH
      GO TO 90
C
C  INORM = 1      IX = 1,  IY = 2,  IZ = 3
   81 XX = XT(IX)
      YY = XT(IY)
      ZZ = XT(IZ)
  110 VT(1) = FI(2)
      VT(2) = FI(3)
      VT(3) = FI(4)
      GO TO 90
C
C  INORM = 2      IX = 1,  IY = 3,  IZ = 2
   82 XX = XT(IX)
      YY = XT(IZ)
      ZZ = XT(IY)
  111 VT(1) = FI(2)
      VT(2) = FI(4)
      VT(3) = FI(3)
      GO TO 90
C
C  INORM = 3      IX = 2,  IY = 3,  IZ = 1
   83 XX = XT(IZ)
      YY = XT(IX)
      ZZ = XT(IY)
  112 VT(1) = FI(4)
      VT(2) = FI(2)
      VT(3) = FI(3)
      GO TO 90
C
C  INORM = 4      IX = 4,  IY = 6,  IZ = 5
   84 XX = XT(IX)
      SINPH = SIN(XT(IY))
      COSPH = COS(XT(IY))
      YY = XT(IZ)*SINPH
      ZZ =-XT(IZ)*COSPH
      EMY(I) = EMY(I)/ZN
  113 VT(1) = FI(2)
```

```
      VT(2) = FI(4)*SINPH + FI(3)*COSPH                                 VAL2  181
      VT(3) =-FI(4)*COSPH + FI(3)*SINPH                                 VAL2  182
C                                                                       VAL2  183
   90 IF (IC .EQ. C)  GO TO 100                                         VAL2  184
      X(I) = XO + XX*AP11 + YY*AP21 + ZZ*AP31                           VAL2  185
      Y(I) = YO + XX*AP12 + YY*AP22 + ZZ*AP32                           VAL2  186
      Z(I) = ZO + XX*AP13 + YY*AP23 + ZZ*AP33                           VAL2  187
C                                                                       VAL2  188
C  CHECK COORDINATE SYSTEM FOR DIRECTION COSINES                        VAL2  189
  100 IF (IT .EQ. 1)  GO TO 500                                         VAL2  190
  120 FMX(I) = VT(1)*AP11 + VT(2)*AP21 + VT(3)*AP31                     VAL2  191
      FMY(I) = VT(1)*AP12 + VT(2)*AP22 + VT(3)*AP32                     VAL2  192
      FMZ(I) = VT(1)*AP13 + VT(2)*AP23 + VT(3)*AP33                     VAL2  193
      GO TO 500                                                         VAL2  194
C                                                                       VAL2  195
C  POINT NOT WITHIN BOUNDARIES.  SET FLAG AND GO TO NEXT POINT.         VAL2  196
   90 INT(I) = 0                                                        VAL2  197
C                                                                       VAL2  198
C                                                                       VAL2  199
  500 CONTINUE                                                          VAL2  200
C                                                                       VAL2  201
C                                                                       VAL2  202
      RETURN                                                            VAL2  203
      END                                                               VAL2  204
```

REFERENCES

1. Gentry, A.E., Smyth, D.N., and Oliver, W.R.: The MARK IV Supersonic-Hypersonic Arbitrary-Body Program. Volume 1 - Users Manual. AFFDL-TR-73-159 Volume 1, November 1973.

2. Gentry, A.E., Smyth, D.N., and Oliver, W.R.: The MARK IV Supersonic-Hypersonic Arbitrary-Body Program. Volume II - Program Formulation. AFFDL-TR-73-159 Volume II, November 1973.

3. Gentry, A.E., Smyth, D.N., and Oliver, W.R.: The MARK IV Supersonic-Hypersonic Arbitrary-Body Program. Volume III - Program Listing. AFFDL-TR-73-159 Volume III, November 1973.

4. Lloyd, M.L., Loo, H.C., Smyth, D.N., and Heely, S.D.: Analysis of Static Pressure Data From 1/12-Scale Model of the YF-12A. Volume I - Data Analysis and Program Description. NASA-CR-137753, October 1981 (C) Title Unclassified.

5. Jackson, C.M., and Sawyer, W.C.: A Method for Calculating the Aerodynamic Loading on Wing-Body Combinations at Small Angles of Attack in Supersonic Flow. NASA TN D-6441, October 1971.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-151940		

4. Title and Subtitle	5. Report Date
ANALYSIS OF STATIC PRESSURE DATA FROM 1/12 SCALE MODEL OF YF-12A AIRPLANE VOLUME III - THE MARK IVS HABP - USERS MANUAL	OCTOBER 1981
	6. Performing Organization Code
7. Author(s) D. N. Smyth and H. C. Loo	8. Performing Organization Report No.
	10. Work Unit No.
9. Performing Organization Name and Address Douglas Aircraft Company McDonnell Douglas Corporation Long Beach, California 90846	11. Contract or Grant No. NAS2-7585
	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

This is one of three final reports.

16. Abstract

In 1972 a series of wind-tunnel tests were conducted in the Ames Unitary Wind-Tunnel System to obtain pressure data on a 1/12-scale model of the YF-12A airplane. The basic wind-tunnel-test static-pressure coefficients have been corrected and then interpolated by using three-dimensional surface-spline techniques. The resulting pressure coefficients were then converted to three-component forces orthogonal to a specified integration coordinate system and integrated by using finite-element integration techniques.

Volume III provides instructions for the use of a digital computer program which was prepared to integrate the pressures to obtain the vehicle components of force and moments and the shears, bending moments, and torques at various stations on the fuselage, wing, and vertical surfaces as well as elevon hinge moments.

The computer program described is identified as the MARKS IVS Mod 0 version of the Supersonic-Hypersonic Arbitrary Body Program.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
YF-12A Research Vehicle Computers (Digital) Data Processing Aerodynamic Loads	STAR Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	436	

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