AD-752 559

V/STOL AIRCRAFT AERODYNAMIC PREDICTION METHODS INVESTIGATION. VOLUME III. MANUAL FOR COMPUTER PROGRAMS

Peter T. Wooler, et al

Northrop Corporation

Prepared for:

Air Force Flight Dynamics Laboratory

January 1972

AFFDL-TR-72-26
Volume III

AD752559

V/STOL AIRCRAFT AERODYNAMIC PREDICTION METHODS INVESTIGATION

Volume III. Manual for Computer Programs

P.T. Wooler
H.C. Kao
M.F. Schwendemann
H.R. Wasson
H. Ziegler

Northrop Corporation
Aircraft Division

TECHNICAL REPORT AFFDL-TR-72-26, Volume III
January 1972

Approved for public release; distribution unlimited.

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

305

NOTICE

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

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

AIR FORCE/56780/30 November 1972 — 300

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Northrop Corporation Aircract Division Hawthorne, California	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

V/STOL AIRCRAFT AERODYNAMIC PREDICTION METHODS INVESTIGATION
Volume III

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report - 1 May 1969 to 31 January 1972

5. AUTHOR(S) (First name, middle initial, last name)

Peter T. Wooler
Hsiao C. Kao
Myles F. Schwendemann
Howard R. Wasson
Henry Ziegler

6. REPORT DATE January 1972	7a. TOTAL NO. OF PAGES 198	7b. NO. OF REFS None
8a. CONTRACT OR GRANT NO F33615-69-C-1602 b. PROJECT NO. 698 BT c. Task No. 698 BT 01 d.	9a. ORIGINATOR'S REPORT NUMBER(S) NOR 72-9, Volume III 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFFDL-TR-72-26, Volume III	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Air Force Flight Dynamics Laboratory Wright-Patterson Air Force Base Ohio 45433

13. ABSTRACT

Analytical engineering methods are developed for use in predicting the static and dynamic stability and control derivatives and force and moment coefficients of lift-jet, lift-fan, and vectored thrust V/STOL aircraft in the hover and transition flight regimes. The methods take into account the strong power effects, large variations in angle of attack and sideslip, and changes in aircraft geometry that are associated with high disk loaded V/STOL aircraft operating in the aforementioned flight regimes. The aircraft configurations studied have a conventional wing, fuselage and empennage. The prediction methods are suitable for use by design personnel during the preliminary design and evaluation of V/STOL aircraft of the type previously mentioned.

This report consists of four volumes. Details of the computer programs associated with the prediction methods are given in this volume. The theoretical development of the prediction methods may be found in Volume I. The methods are applied to a number of V/STOL configurations in Volume II. The results of a literature survey are presented in Volume IV.

DD FORM 1 NOV 65 1473

Unclassified
Security Classification

ia

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

☆U S Government Printing Office: 1972 — 759-488/196

Security Classification

V/STOL AIRCRAFT AERODYNAMIC PREDICTION METHODS INVESTIGATION

Volume III
Manual for Computer Programs

P.T. Wooler
H.C. Kao
M.F. Schwendemann
H.R. Wasson
H. Ziegler

ib

FOREWORD

This report summarizes the work accomplished by the Aircraft Division of Northrop Corporation, Hawthorne, California, for the Air Force Flight Dynamics Laboratory, AFSC, Wright Patterson Air Force Base, Ohio, under USAF Contract No. F33615-69-C-1602 (Project 698 BT). This document consitutes the Final Report under the contract.

This work was accomplished during the period 1 May 1969 to 31 January 1972, and this report was released by the authors in January 1972. The Air Force Project Engineers were Mr. Robert Nicholson and Mr. Henry W. Woolard of the Control Criteria Branch, Flight Control Division, AFFDL. Their assistance in monitoring the work and providing data is greatly appreciated.

This technical report has been reviewed and is approved.

C. B. Westbrook
Chief, Control Criteria Branch
Flight Control Division
Air Force Flight Dynamics Laboratory

ABSTRACT

Analytical engineering methods are developed for use in predicting the static and dynamic stability and control derivatives and force and moment coefficients of lift-jet, lift-fan, and vectored thrust V/STOL aircraft in the hover and transition flight regimes. The methods take into account the strong power effects, large variations in angle of attack and sideslip, and changes in aircraft geometry that are associated with high disk loaded V/STOL aircraft operating in the aforementioned flight regimes. The aircraft configurations studied have a conventional wing, fuselage and empennage. The prediction methods are suitable for use by design personnel during the preliminary design and evaluation of V/STOL aircraft of the type previously mentioned.

This report consists of four volumes. Details of the computer programs associated with the prediction methods are given in this volume. The theoretical development of the prediction methods may be found in Volume I. The methods are applied to a number of V/STOL configurations in Volume II. The results of a literature survey are presented in Volume IV.

TABLE OF CONTENTS

Preceding page blank

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

The purpose of this investigation was to develop analytical engineering methods for predicting the static and dynamic longitudinal and lateral-directional aerodynamic stability and control derivatives and coefficients of lift jet, lift fan, and vectored thrust V/STOL aircraft in the hover and transition flight regimes during unaccelerated flight conditions. The methods developed under the investigation were to be suitable for use by design personnel during the preliminary design and evaluation of lift jet, lift fan and vectored thrust V/STOL aircraft. Where appropriate, the methods developed might use high speed computers to permit solutions to be obtained within reasonable time periods. The aircraft configurations studied were to have a conventional wing, fuselage and empennage.

In Volume I the aerodynamic prediction methods are developed in a form suitable for application to each aircraft component. The theoretical basis or semi-empirical analysis is presented. Empirical coefficients are determined, where necessary, and extensive comparisons of calculations with test data are made.

Volume II gives detailed examples of the application of the prediction methods to the determination of the aerodynamic forces, moments, and, in some cases, surface pressure distributions, on the aircraft wing, fuselage and empennage. In each case a sample problem is given with method applicability and limitations discussed.

This volume is intended to serve as a User's Manual for the computer programs developed as part of the investigation. Information dealing with both the operating and programming aspects is presented for each computer program developed as part of the effort. An abbreviated section is included on the Lifting Surface program, which is utilized in the application of the prediction methods presented in Volume II, but is itself a modified version of an existing program. A complete listing of all the programs is appended.

SECTION II

JET FLOW FIELD PROGRAM

1. DESCRIPTION

The Jet Flow Field program evaluates the induced velocity field due to single or multiple jets exhausting into an arbitrarily directed mainstream.

The equations of motion governing the development of each jet are integrated numerically for the position of the jet centerline, the nondimensionalized mean jet speed and the nondimensionalized major diameter of the ellipse which represents the jet cross section in the mathematical model. The set of first order equations is integrated by means of a fourth order Adams predictor/corrector routine with a Runge-Kutta starting solution.

The induced velocity components due to each jet at a given control point are then calculated by replacing each jet with a representative singularity distribution of sinks and doublets along the jet centerline. The contributions to the induced velocity components from the singularity distribution are summed over the length of each jet centerline. The velocity components due to each of the singularity distributions are additive at every control point.

For multiple jet configurations, distances between jet centerlines are tested and when intersection of two jets is indicated, a coalesced jet is established from continuity and momentum considerations. The coalesced jet is treated as another independent jet in the computations for the induced velocity field.

a. Restrictions

Jets must exhaust at some angle into the mainstream, i. e. the jet exhaust direction may not coincide with the freestream direction.

For a two-jet configuration the jet exits must both lie in the same XY plane and the jet exhaust plane, defined by the freestream vector and the initial jet exhaust vector must be the same for both jets (see Figure 1 for definition of coordinate system).

The same restrictions apply to a three-jet configuration. Additionally three-jet configurations must be colinear and negative angles of attack cannot be treated.

Control points at which jet-induced velocity components are to be evaluated may not lie within the jet exhaust itself, as the formulation of the mathematical model is not valid in this region. Generally, control points positioned less than 2 jet exit diameters from the center of the jet exit should be avoided.

b. Options

- Wing Option: The program computes the control points from the mapping coefficients and radii generated by the Mapping Function program.
- Fuselage Option: The program computes the control points from the mapping coefficients and radii generated by the Mapping Function program.
- Tabulation Option: Coordinates of the control points are provided as part of the input to the program.

The first two options assure compatibility with the Transformation Method program, when the Jet Flow Field program is to be used in conjunction with that program. The punch control option is exercised to generate data for the Transformation Method program in card form.

The third option may be utilized to generate input to the Lifting Surface program, by again exercising the punch control option.

2. OPERATING INFORMATION

Core and Time Requirements:

Computer:	CDC 6600
Core:	$100K_8$ to load
	$62K_8$ to execute
Time:	Approximately 0.6 minutes for a typical run using 250 control points.

Additional Requirements: None

3. INPUT DATA

Figure 1 shows a typical wing configuration relative to the input/output coordinate system. Figure 2 shows a typical fuselage configuration relative to this coordinate system.

The input cards required by the program are shown in Figure 3. The cards of Group A are always required. They are followed by the cards of Group B or Group C or Group D depending on which of the geometry options discussed above is being executed. The input cards are grouped in this manner and discussed in detail below.

Card No.	Variable	Format	Description
GROUP A: Required for all runs			
①	MULT	I6	Specifies number of jets in configuration MULT = 1, 2 or 3
	IGEØM	I6	Specifies option of program being exercised If IGEØM = 1 control points computed on wing = 2 control points computed on fuselage = 3 control points are provided as input = 4 same as 3, but flat plate pressure coefficient is also computed at every control point
	IPUNCH	I6	Punch control If IPUNCH = 0 no punched output = 1 punched output
②	ALFA	F12.0	Angle of attack α (defined in Figure 2) } in degrees
	BETA	F12.0	Angle of sideslip β (defined in Figure 2) } in degrees
③	N	I6	Number of steps to be used in numerical integration of jet centerline Limit: $N \leq 100$
	G	F12.0	Step size in numerical integration of jet centerline, in fraction of jet exit diameter
④	XJET	F12.0	X-coordinate of center of jet exit
	YJET	F12.0	Y-coordinate of center of jet exit
	ZJET	F12.0	Z-coordinate of center of jet exit
	PHI	F12.0	Jet exhaust angle ϕ (defined in Figure 1) } in degrees
	PSI	F12.0	Jet exhaust angle ψ (defined in Figure 1) } in degrees
	DJET	F12.0	Jet diameter
⑤	VELJ	F12.0	Freestream to jet exhaust velocity ratio

Card No.	Variable	Format	Description

- Cards of the type 4 and 5, describing the other jets, follow at this point if MULT>1. For multiple jet configurations, upstream jets are listed ahead of downstream jets.

Card No.	Variable	Format	Description
⑥	DIA	F12.0	Empirical factor controlling initial cross section of a coalesced jet. Function of jet orientation angle Ω. (See Vol I, p. 56 for definition) If Ω | <20° DIA = 1.0 | >70° DIA = 0.5 May be left blank for a single-jet configuration.

GROUP B: Cards provide data to generate control points on wing

Card No.	Variable	Format	Description
①	NTHT	I6	Number of control points at each spanwise station or number of equal increments $\Delta\theta$ into which the mapping circle is divided
	NS	I6	Number of spanwise locations where control points are located Limit: NS ≤ 25
	NCØEF	I6	Number of terms used in the mapping expansion Limit: NCØEF ≤ 15
	IRECT	I6	Indicates whether or not wing is rectangular If IRECT | = 0 wing is rectangular | = 1 wing is not rectangular
②	Y(I)	F12.0	Spanwise location of control station
	R(I)	F12.0	Radius of mapping circle
	DRDY(I)	F12.0	Rate of change of R with Y
③	A(J, I)	E12.5	Real part of mapping coefficient. } J=1, NCØEF
	B(J, I)	E12.5	Imaginary part of mapping coefficient } J=1, NCØEF

- Sets of cards now follow to describe the other wing stations, I = 2, NS.
- If IRECT = 0, cards listing the real and imaginary parts of the coefficients are omitted.

Card No.	Variable	Format	Description

GROUP C: Cards provide data to generate control points on fuselage

Card No.	Variable	Format	Description
①	NTHT	I6	Number of control points at each station, if NSYM = 1. If NSYM = 0, number of control points generated will be NTHT + 1.
	NS	I6	Number of fuselage stations where control points are located Limit: NS ≤ 25
	NCØEF	I6	See definition, card 1, Group B
	NSYM	I6	Flow symmetry indicator If NSYM { = 0 compute only starboard side = 1 compute entire cross section
②	X(I)	F12.0	X-coordinate of control station
	R(I)		See definition, card 2, Group B
	DRDX(I)		Rate of change of R with X
③	A(J, I)	E12.5	Real part of mapping coefficient J = 1, NCØEF

- Sets of cards now follow to describe the other fuselage stations, I = 2, NS

Note: For procedure of obtaining mapping coefficients and radii, refer to Volume II, Section I and to Section III of this volume.

GROUP D: Cards provide control points as direct input

Card No.	Variable	Format	Description
①	NS	I6	Number of spanwise control stations
	NC	I6	Number of control points at each station
②	X0(I)	F12.0	X-coordinate of control point } I = 1, NC x NS
	Y0(I)	F12.0	Y-coordinate of control point } I = 1, NC x NS
	Z0(I)	F12.0	Z-coordinate of control point } I = 1, NC x NS

Combined Limits: **Group B:**   **NS x NTHT ≤ 600**

Group C: **If NSYM** { = 0 NS x (NTHT + 1) ≤ 600 ; = 1 NS x NTHT ≤ 600 }

Group D: **NC x NS ≤ 600**

3. OUTPUT

Both printed and punched output may be obtained

a. Printed Output

The jet configuration being treated is identified both by appropriate heading and by printout of pertinent input information. Jet centerline information on all the jets in the configuration includes the centerline coordinates, the nondimensionalized mean jet speed and the nondimensionalized major diameter of the ellipse representing the jet cross section. Points of intersection of jets are identified.

The induced velocity components U, V, W, all nondimensionalized by U_∞ are printed out at each control point. Additionally, if IGEØM = 4 was specified, the flat plate pressure coefficient, computed by using an image system, is printed out at each control point.

b. Punched Output

For the first two options discussed in subsection 1.b, punched cards may be generated which form a continuous input data block for the Transformation Method program. Data are punched in sets for X- or Y-stations. Data consist of station, radius of mapping circle, rate of change of the radius, mapping coefficients and induced velocities at the control points. For convenience, the punched cards are identified and sequenced in cols 73-80.

For the third option discussed in subsection 1.b, punched cards may be generated which can be utilized as part of the input to the Lifting Surface program. The non-dimensionalized velocity component W is punched out for every control point. This can serve as an approximation for the tangent of the jet-induced downwash angle for small angles of attack. Thus the punched output from this option can serve as the downwash matrix [W] in the input to the Lifting Surface program.

4. PROGRAMMING INFORMATION

a. Logical Structure

The logical flow chart for the program is shown in Figure 4.

b. Purpose of Subroutines

BITEST	- Tests for blockage and intersection of jets for multiple-jet configurations
INTEG	- Integrates equations of motion for the jet path
CØMP	- Computes extent of overlap between the jets in a multiple-jet configuration
BALANC	- Establishes initial conditions for a coalesced jet from a momentum balance
ØUTPT	- Transforms local coordinates to program coordinates
VELØC	- Evaluates induced velocities at one control point
DERIV	- Computes derivatives for ADAMS
TRWING	- Computes control points on wing
TRBØDY	- Computes control points on fuselage
ADAPT	- Punches output for Transformation Method program
PRTØUT	- Prints out computed answers
TRANS1 TRANS2	- Transforms input coordinates to program coordinates
VEL1	- Computes effective velocity ratios for downstream jets in a multiple jet configuration
TRANS3	- Transforms program coordinates to output coordinates
PLANE	- Computes point of intersection between a given plane and a given line
ADAMS	- Adams predictor/corrector routine
CFCAL	- Computes direction cosines for the jet-centered coordinate system

RØTATE	-	Transforms program coordinates to jet-centered coordinates
XPRØD	-	Computes cross product of two vectors
SØL	-	Solves a system of three simultaneous equations

c. Interdependence of Subroutines

The Calling-Called matrix for the program is shown in Figure 5.

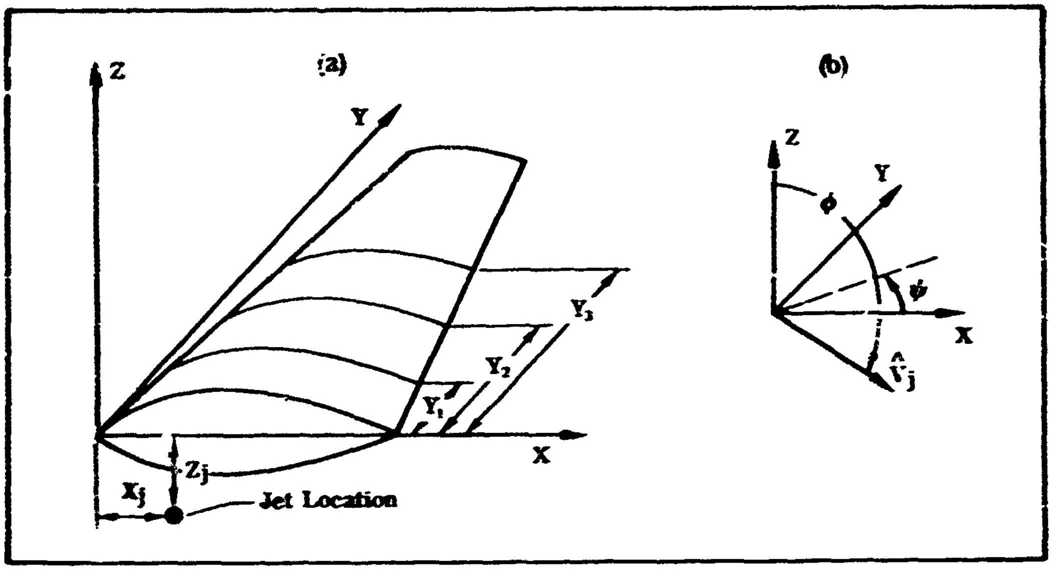

FIGURE 1. COORDINATE SYSTEM FOR TYPICAL WING

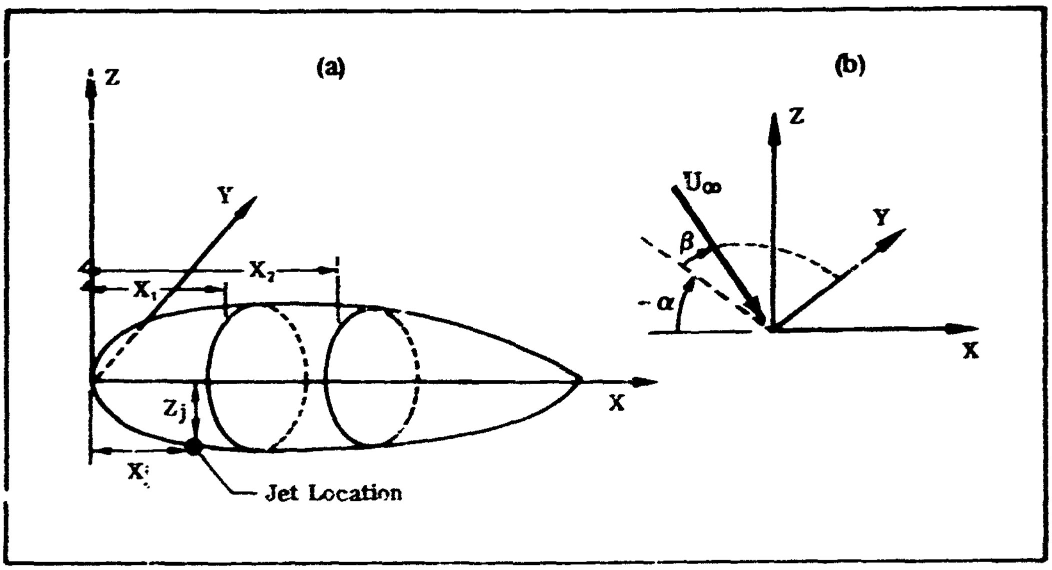

FIGURE 2. COORDINATE SYSTEM FOR TYPICAL FUSELAGE

A	①	MULT	IGEØM	IPUNCH					
A	②	ALFA		BETA					
A	③	N	G						
A	④	XJET		YJET		ZJET	PHI	PSI	DJET
A	⑤	VELJ							
		● Additional data as indicated							
A	⑥	DIA							
B	①	NTHT	NS	NCØEF	IRECT				
B	②	Y (I)		R (I)		DRDY (I)			
B	③	A (J, I)		B (J, I)					
		● Additional data as indicated							
C	①	NTHT	NS	NCØEF	NSYM				
C	②	X (I)		R (I)		DRDX (I)			
C	③	A (J, I)							
		● Additional data as indicated							
D	①	NS	NC						
D	②	X0 (I)		Y0 (I)		Z0 (I)			

FIGURE 3. INPUT DATA FOR JET FLOW FIELD PROGRAM

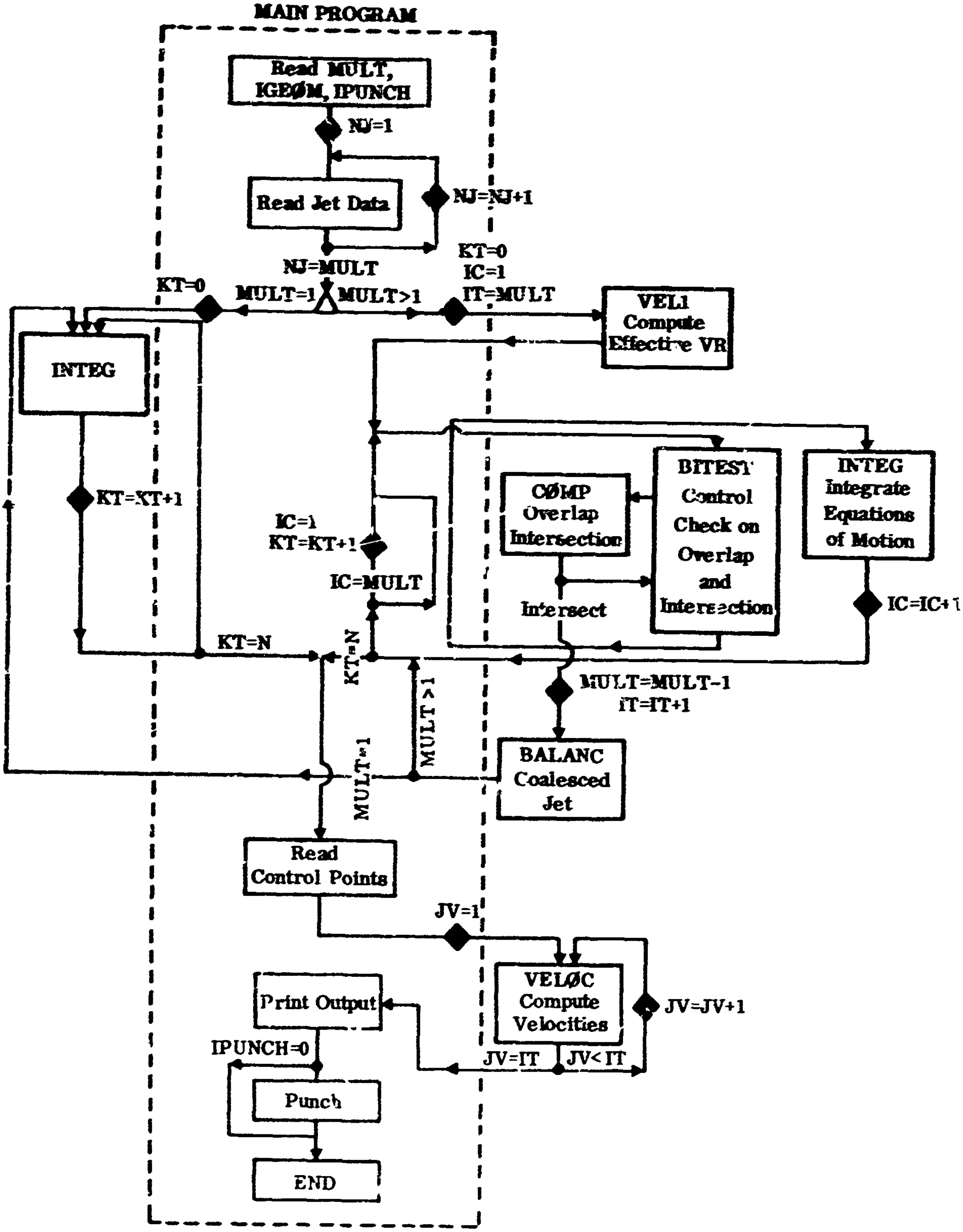

FIGURE 4. LOGICAL FLOW CHART FOR JET FLOW FIELD PROGRAM

Calling \ Called	ADAPT	BITEST	CFCAL	INTEG	PRTØUT	RØTATE	TRANS1	TRANS2	TRANS3	TRBØDY	TRWING	VELØC	VEL1	XPRØD	BALANC	CØMP	PLANE	ADAMS	ØUTPT	SØL	DERIV
MAIN	●	●	●	●	●	●	●	●	●	●	●	●	●	●							
BITEST			●			●								●	●	●	●				
INTEG																		●	●		
CØMP														●			●				
ØUTPT						●															
VELØC						●															
VEL1														●			●				
PLANE																				●	
ADAMS																					●
CFCAL														●							

FIGURE 5. CALLING-CALLED MATRIX FOR JET FLOW FIELD PROGRAM

SECTION III

MAPPING FUNCTION PROGRAM

1. DESCRIPTION

The mapping function program provides a method of obtaining a mapping of an arbitrary cross section into a unit circle. This mapping is obtained by first developing a potential for a vortex flow about the section and comparing this potential with the known potential for a vortex flow about the circle. Points where the two potentials are equal are known to map into each other in a conformal transformation. Knowing the point-to-point correspondence between points on the section and points on the mapping circle, it is then possible to obtain the derivative of the mapping function with any corners on the section explicitly specified. This derivative of the mapping function is integrated numerically about the mapping circle and the mapped section obtained is printed out.

The program also takes the derivative of the mapping and removes the corners which are contained explicitly by expanding the expressions specifying the corners. The expression thus obtained can be integrated analytically to obtain the mapping function. The mapping function is obtained in this manner and the coefficients of the mapping function obtained are printed out. The program then prints out section coordinates for the section as obtained from this mapping function. This mapped section can then be compared with the original section to determine the accuracy of the mapping.

a. Restrictions

Cross sections must describe a discrete cross-sectional area.

Corner points must be separated by an element of distance Δs.

2. OPERATING INFORMATION

Core and Time Requirements:

Computer: CDC 6600

Core. 56,[illegible] to load

4[illegible] to execute

Time: Approximately .25 minutes for a typical symmetric section with NTERM = 10. Sections with corners and asymmetric sections would require more time.

Additional Requirements: None

3. INPUT DATA

Figure 6 defines the coordinates in the section and circle planes.

The input data cards required are shown in Figure 7. They are described in detail below.

Card No.	Variable	Format	Description
①	NPT	I3	Number of coordinate points describing the section to be read Limit: NPT $\leq$ 90
	KØRN	I3	Number of corners or pseudocorners on section Limit: KØRN $\leq$ 20
	NTERM	I3	Number of terms in potential expansion and mapping series to be computed Limit: NTERM $\leq$ 50
	NSYM	I3	Symmetry indicator If NSYM = 0 symmetric section If NSYM = 1 asymmetric section
②	X(I)	F9.5	X-coordinates of points describing the section, listed in sequential order starting at the positive X-axis and going counterclockwise. I = 1,NPT If NSYM = 0 last point is on negative X-axis If NSYM = 1 last point is same as first point
③	Y(I)	F9.5	Y-coordinates of points describing the section. I = 1,NPT
④	DX	F9.5	Shift of coordinate system along X-axis desired to center section.

- If KØRN = 0, cards 5, 6 are omitted.

Card No.	Variable	Format	Description
⑤	NCØR(I)	I3	For a true corner, this is the sequence number of the corner point in the X(I) tabulation. For a pseudocorner, NCØR(I) = 0. I = 1,KØRN Limit: Second point in tabulation may not be a corner point. Adjacent points in tabulation may not be corner points.
⑥	XCØR(I)	F9.5	X-coordinate of corner point or pseudocorner point.
	YCØR(I)	F9.5	Y-coordinate of corner point or pseudocorner point.
	DALPHA(I)	F9.5	Angle $\Delta\alpha$ turned through at the corner, specified in radians. ($\mid$DALPHA(I)$\mid \leq \pi$, sign convention is shown in Figure 6 ; see also Figure 47, Vol I, p. 79)

- There would now follow cards for I = 2, KØRN.
- If NSYM = 0, card 7 is omitted.

Card No.	Variable	Format	Description
⑦	ALPHA(1)	F9.5	Angle α which the tangent to the section makes with the X-axis at the first point. If the first point is a corner point the angle between the X-axis and the normal to the bisector of $\Delta\alpha$ is utilized.
⑧	X1	F6.2	X-coordinate for first point of numerical integration of mapping
	Y1	F6.2	Y-coordinate for first point of numerical integration of mapping
	TH0	F6.2	Angle θ about mapping circle, corresponding to the first point to be mapped (in degrees).
	THF	F6.2	Angle θ about mapping circle, corresponding to the last point to be mapped (in degrees).
	DTH	F6.2	Approximate spacing of mapping in increments about the mapping circle (in degrees).

Note: Card 8 gives parameters for numerical integration of the derivative of the mapping function. Card 9 gives the parameters for the analytically integrated mapping function . (See Eqs. 58, 59 Vol I, p.83)

Card No.	Variable	Format	Description
⑨	N	I3	Number of points at which mapping is to be computed.
	DTH	F6.2	Angular spacing about mapping circle at which mapped points are to be located, specified in degrees.
	TH0	F6.2	See definition, card 8.

Note: The optimum value of NTERM is to some extent dependent on the section to be mapped. NTERM = 10 normally gives a satisfactory mapping. Too large a number of terms may cause a divergence of the series, especially for thin sections such as airfoils.

4. OUTPUT

Figure 8 shows an example of the output obtained from the mapping program. This example is for a symmetrical body section.

Figure 8(a) shows some of the parameters calculated in computing the potential about the given section and comparing the results with the unit circle potential. Columns 1 and 2 reproduce the input X and Y coordinates of the section outline, except that the X value has been shifted by an amount DX which was specified in the input data. Column 3 gives the radial distance R_b from each point to the new origin. Column 4 gives the section distance s to each point. Column 5 gives the velocity computed at each point. Velocities written out at corner points are meaningless. Column 6 gives the angle α which the section tangent makes with the X-axis. Column 7 gives the position angle ω for each point in degrees. Column 8 gives the angle θ around the mapping circle in degrees.

Figure 8(b) gives the mapping obtained for the input section by numerical integration. The first and second columns are the X and Y coordinates on the mapped section, and the third column gives the angular distance around the mapping circle for each point in radians. The extent of the section printed out here and the number of points is specified by card 8 of the input data.

Figure 8(c) shows the mapping circle radius and the coefficients of the mapping function with the corners removed. The real parts of the coefficients are written first and then the imaginary parts, which in this example are zero. The number of coefficients calculated is one less than the NTERM specified in the input.

Figure 8(d) tabulates the X and Y coordinates of the mapped section with the corners removed from the mapping. The number of points and spacing between points were specified by input card 9.

5. PROGRAMMING INFORMATION

a. Logical Structure

The logical flow chart for the program is shown in Figure 9.

b. Purpose of Subroutines

MAPP1 — This subroutine computes the coefficients of the derivative of the mapping function without the corners explicitly expressed. The subroutine then computes the corner parameters and obtains the derivative of the mapping with the corners explicitly expressed. The subroutine then sets up a series of increments around the mapping circle at which points of the mapping are to be computed. It then calls MAPP which computes points on the section. The points on the section are then printed out.

MAPP5 — This subroutine removes the corners from the derivatives of the mapping function and evaluates the coefficients for this form of the derivative. The analytical integration is then performed. The program then computes points on the section using the mapping function obtained at points requested by the inputs. The program prints out the radius of the mapping circle, the coefficients of the mapping function and the points computed from the mapping representing the section.

MAPP — This subroutine is used to compute a point on the section after an incremental distance about the mapping circle has been traveled. Three options are provided for this routine. The first option (KODE = 1) specifies that the end points of the increment are both on the circle and the integration is carried out on the unit circle. This option is used when no corner point is in the interval. The second option (KODE = 2) integrates the derivative of the mapping function along a radial line. This option is not used by the program. The third option (KODE = 3) integrates about a corner point. A semicircular path about the corner point is followed external to the mapping circle and a point on the section past the corner is computed.

MATINV — Inverts a matrix

QATAN — Computes $\tan^{-1}(y/x)$ given y and x. The angle computed is not the principal angle but ranges from 0 to 360 degrees, depending on the signs of x and y.

c. Interdependence of Subroutines

The Calling-Called matrix for the program is shown in Figure 10.

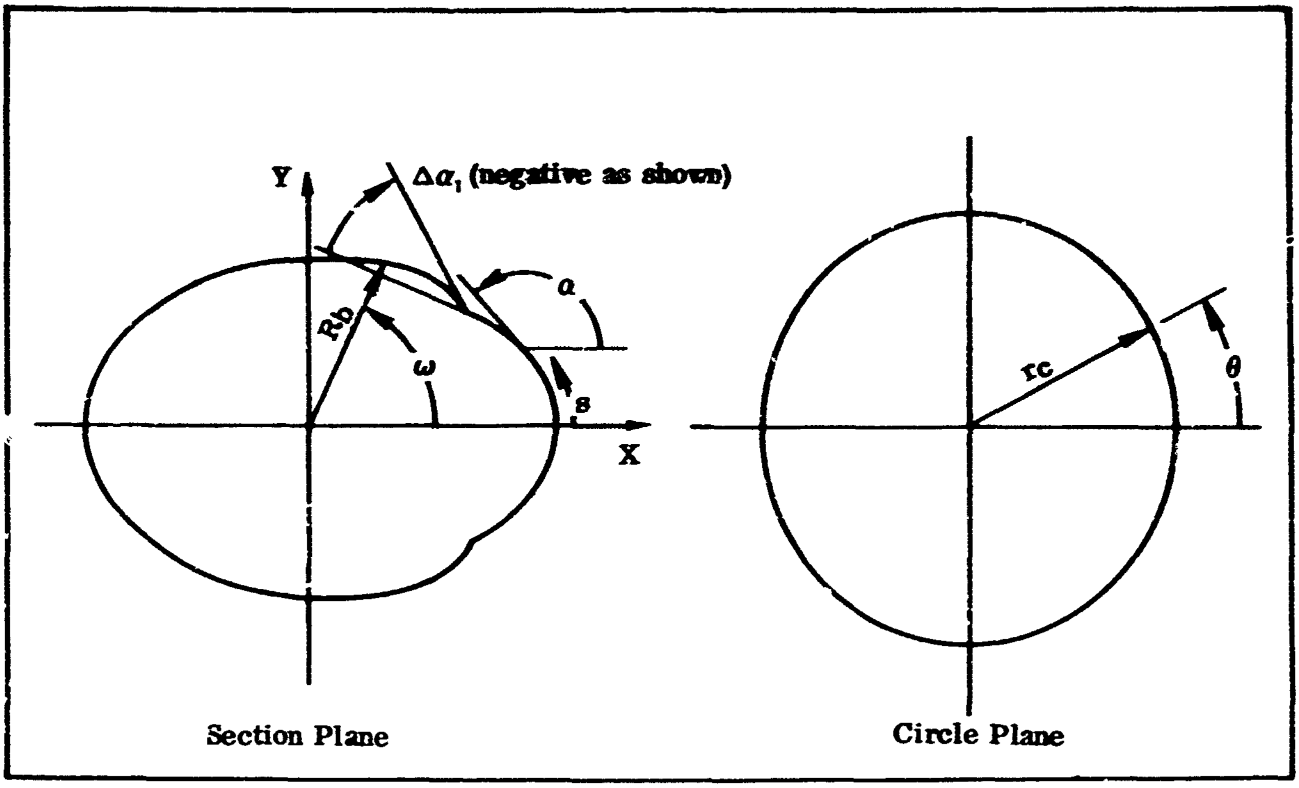

FIGURE 6. COORDINATE SYSTEM FOR SECTION AND CIRCLE PLANES

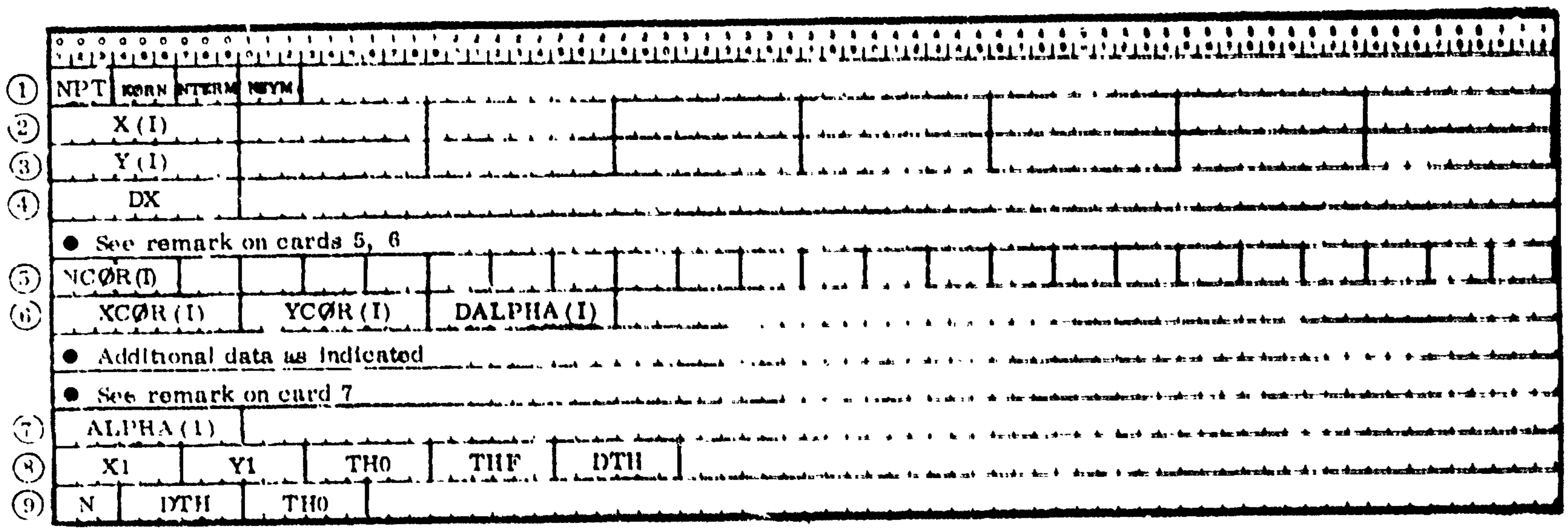

FIGURE 7. INPUT DATA FOR MAPPING FUNCTION PROGRAM

COMPUTATIONS FOR S AND ALPHA VERSUS THETA.

X	Y	R	S	V	ALPHA	OMEGA	THETA
0.29300E 02	0.0	0.29300E 02	0.0	0.23437E-01	0.90000E 02	0.0	0.0
0.29300E 02	0.29951E 01	0.29453E 02	0.29951E 01	0.23481E-01	0.90000E 02	0.58386E 01	0.40000E 01
0.29300E 02	0.59899E 01	0.29906E 02	0.59899E 01	0.24882E-01	0.90000E 02	0.11594E 02	0.80750E 01
0.29300E 02	0.89848E 01	0.30647E 02	0.89848E 01	0.27908E-01	0.90000E 02	0.17046E 02	0.12923E 02
0.29300E 02	0.11980E 02	0.31655E 02	0.11980E 02	0.32014E-01	0.90000E 02	0.22238E 02	0.17686E 02
0.29300E 02	0.14975E 02	0.32905E 02	0.14975E 02	0.36064E-01	0.90473E 02	0.27071E 02	0.23902E 02
0.29212E 02	0.17964E 02	0.34293E 02	0.17969E 02	0.36681E-01	0.99159E 02	0.31984E 02	0.29017E 02
0.27955E 02	0.20648E 02	0.34754E 02	0.20964E 02	0.37472E-01	0.12987E 03	0.36450E 02	0.36137E 02
0.25066E 02	0.22557E 02	0.34170E 02	0.23964E 02	0.38991E-01	0.14923E 03	0.41311E 02	0.42061E 02
0.22938E 02	0.23784E 02	0.33043E 02	0.26961E 02	0.39790E-01	0.16147E 03	0.46037E 02	0.49410E 02
0.20041E 02	0.24535E 02	0.31680E 02	0.29950E 02	0.38162E-01	0.16914E 03	0.50757E 02	0.56108E 02
0.17075E 02	0.24945E 02	0.30229E 02	0.32951E 02	0.33779E-01	0.17487E 03	0.55308E 02	0.62244E 02
0.14085E 02	0.25096E 02	0.28778E 02	0.35946E 02	0.26503E-01	0.17893E 03	0.60097E 02	0.67344E 02
0.11090E 02	0.25100E 02	0.27441E 02	0.38941E 02	0.21492E-01	0.18044E 03	0.66162E 02	0.71389E 02
0.80952E 01	0.25100E 02	0.26373E 02	0.41936E 02	0.20983E-01	0.17970E 03	0.72124E 02	0.74949E 02
0.51004E 01	0.25100E 02	0.25613E 02	0.44930E 02	0.23460E-01	0.18011E 03	0.78512E 02	0.78730E 02
0.21055E 01	0.25100E 02	0.25188E 02	0.47925E 02	0.22589E-01	0.17997E 03	0.85203E 02	0.82742E 02
0.88936E 00	0.25100E 02	0.25116E 02	0.49142E 02	0.21112E-01	0.18003E 03	0.87496E 02	0.86257E 02
-0.38042E 01	0.25100E 02	0.25399E 02	0.53919E 02	0.20147E-01	0.17999E 03	0.98799E 02	0.89613E 02
-0.68791E 01	0.25100E 02	0.26026E 02	0.56910E 02	0.22882E-01	0.18000E 03	0.10532E 03	0.93304E 02
-0.98739E 01	0.25100E 02	0.26972E 02	0.59905E 02	0.22280E-01	0.18000E 03	0.11147E 03	0.97210E 02
-0.12869E 02	0.25100E 02	0.28207E 02	0.62900E 02	0.20449E-01	0.18000E 03	0.11714E 03	0.10084E 03
-0.15863E 02	0.25100E 02	0.29692E 02	0.65894E 02	0.21111E-01	0.18029E 03	0.12229E 03	0.10434E 03
-0.18858E 02	0.25048E 02	0.31353E 02	0.68889E 02	0.24374E-01	0.18199E 03	0.12697E 03	0.10819E 03
-0.21847E 02	0.24874E 02	0.33106E 02	0.71884E 02	0.28682E-01	0.18681E 03	0.13129E 03	0.11272E 03
-0.24823E 02	0.24535E 02	0.34902E 02	0.74879E 02	0.32219E-01	0.18848E 03	0.13534E 03	0.11799E 03
-0.27761E 02	0.23963E 02	0.36673E 02	0.77874E 02	0.34407E-01	0.19392E 03	0.13920E 03	0.12306E 03
-0.30617E 02	0.23064E 02	0.38332E 02	0.80870E 02	0.35066E-01	0.20161E 03	0.14330E 03	0.12961E 03
-0.33291E 02	0.21719E 02	0.39749E 02	0.83868E 02	0.34901E-01	0.21261E 03	0.14687E 03	0.13559E 03
-0.35620E 02	0.19839E 02	0.40772E 02	0.86868E 02	0.34480E-01	0.22969E 03	0.15048E 03	0.14152E 03
-0.37446E 02	0.17468E 02	0.41320E 02	0.89867E 02	0.34072E-01	0.23879E 03	0.15499E 03	0.14734E 03
-0.38744E 02	0.14770E 02	0.41464E 02	0.92865E 02	0.33584E-01	0.24929E 03	0.15713E 03	0.15316E 03
-0.39609E 02	0.11904E 02	0.41359E 02	0.95861E 02	0.32862E-01	0.25671E 03	0.16327E 03	0.15886E 03
-0.40159E 02	0.89611E 01	0.41147E 02	0.98859E 02	0.31887E-01	0.26185E 03	0.16741E 03	0.16437E 03
-0.40485E 02	0.59845E 01	0.40925E 02	0.10185E 03	0.30843E-01	0.26946E 03	0.17194E 03	0.16972E 03
-0.40652E 02	0.29944E 01	0.40762E 02	0.10485E 03	0.30031E-01	0.26804E 03	0.17574E 03	0.17491E 03
-0.40700E 02	0.0	0.40700E 02	0.10784E 03	0.29725E-01	0.27004E 03	0.17999E 03	0.18000E 03

FIGURE 8(a). SAMPLE OUTPUT FOR MAPPING FUNCTION PROGRAM

SECTION MAPPING BY NUMERICAL INTEGRATION.

X	Y	THETA
0.29352E 02	0.36200E 01	0.84900E-01
0.29477E 02	0.70875E 01	0.16982E 00
0.29596E 02	0.10282E 02	0.25472E 00
0.29600E 02	0.13133E 02	0.33963E 00
0.29391E 02	0.15626E 02	0.42454E 00
0.28895E 02	0.17781E 02	0.50945E 00
0.28077E 02	0.19635E 02	0.59436E 00
0.26934E 02	0.21216E 02	0.67926E 00
0.25475E 02	0.22534E 02	0.76417E 00
0.23787E 02	0.23566E 02	0.84908E 00
0.21628E 02	0.24364E 02	0.93399E 00
0.19196E 02	0.24876E 02	0.10189E 01
0.16414E 02	0.25153E 02	0.11038E 01
0.13270E 02	0.25250E 02	0.11887E 01
0.97872E 01	0.25233E 02	0.12736E 01
0.60171E 01	0.25169E 02	0.13585E 01
0.20378E 01	0.25109E 02	0.14434E 01
-0.20582E 01	0.25052E 02	0.15283E 01
-0.61745E 01	0.25094E 02	0.16132E 01
-0.10226E 02	0.25130E 02	0.16982E 01
-0.14112E 02	0.25160E 02	0.17831E 01
-0.17782E 02	0.25141E 02	0.18680E 01
-0.21179E 02	0.25023E 02	0.19529E 01
-0.24269E 02	0.24742E 02	0.20378E 01
-0.27041E 02	0.24253E 02	0.21227E 01
-0.29503E 02	0.23512E 02	0.22076E 01
-0.31676E 02	0.22494E 02	0.22925E 01
-0.33589E 02	0.21196E 02	0.23774E 01
-0.35261E 02	0.19633E 02	0.24623E 01
-0.36706E 02	0.17828E 02	0.25472E 01
-0.37917E 02	0.15807E 02	0.26321E 01
-0.38896E 02	0.13585E 02	0.27170E 01
-0.39643E 02	0.11173E 02	0.28020E 01
-0.40173E 02	0.85806E 01	0.28869E 01
-0.40515E 02	0.58260E 01	0.29718E 01
-0.40701E 02	0.29467E 01	0.30567E 01
-0.40760E 02	-0.10836E-02	0.31416E 01

FIGURE 8(b). (Continued)

```
RADIUS OF MAPPING CIRCLE =  0.33517E 02
REAL PARTS OF COEFFICIENTS.
  0.16505E 03 -0.80102E 03 -0.11475E 06 -0.54775E 06  0.17340E 07 -0.11495E 09 -0.70960E 10
 -0.60582E 11 -0.20672E 13
IMAGINARY PARTS OF COEFFICIENTS.
  0.0          0.0          0.0          0.0          0.0          0.0          0.0
  0.0          0.0
```

FIGURE 8(c). (Continued)

MAPPING OF SECTION WITH CORNERS REMOVED

X	Y
33.74065	0.0
33.79533	[illegible].71916
33.92583	[illegible].27334
34.04285	10.53344
34.03062	13.42824
33.78172	15.94495
33.22249	18.11038
32.32132	19.96355
31.07805	21.53214
29.50369	22.82333
27.60135	23.83009
25.35806	24.54831
22.75081	24.99301
19.76187	25.20651
16.39557	25.25409
12.68871	25.20993
8.71039	25.14067
4.55337	25.09227
0.32129	25.08478
-3.88293	25.11311
-7.96487	25.15170
-11.84319	25.15948
-15.45312	25.08385
-18.74998	24.86530
-21.71246	24.44403
-24.34244	23.76903
-26.66010	22.80710
-28.69418	21.54808
-30.47015	20.00362
-32.00211	18.19855
-33.29158	16.15981
-34.33445	13.90728
-35.13164	11.45243
-35.69736	8.83503
-36.06100	5.98449
-36.25879	3.02955
-36.32085	0.00042

FIGURE 8(d). (Concluded)

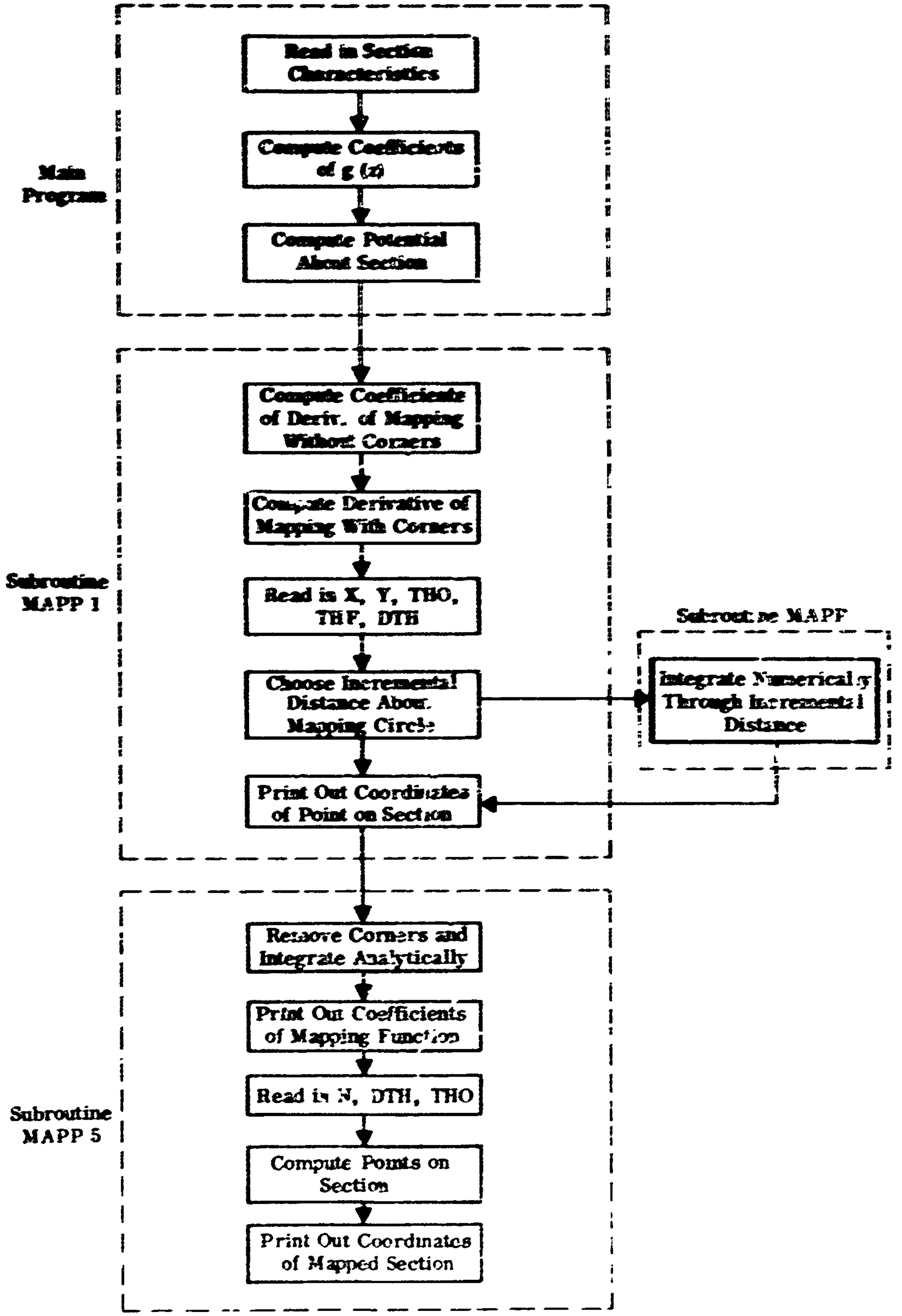

FIGURE 9. LOGICAL FLOW CHART FOR MAPPING FUNCTION PROGRAM

Calling \ Called	MAPP1	MAPP8	MATINV	QATAN	MAPP
MAIN	●	●	●	●	
MAPP1					●

FIGURE 19. CALLING-CALLED MATRIX
FOR MAPPING FUNCTION PROGRAM

SECTION IV

TRANSFORMATION METHOD PROGRAM

1. DESCRIPTION

This program computes the pressure distributions on a wing or a fuselage. By integrating the pressure on the surface, the force and moment can be obtained.

The principal input data are the induced velocity field and the mapping coefficients given by Sections II and III. The former is, however, calculated with no obstacle present in the flow. Thus, the main function of the transformation method is to insert a wing or a fuselage in this given field and to move the obstacle momentarily in such a manner that the boundary condition is satisfied. This induces a velocity potential from which, along with the potential caused by the exhausting jet, the surface pressure can be determined.

a. Restrictions

Some implicit assumptions made in the program to describe a wing or fuselage must be satisfied. The following restrictions do not apply when only the segment method is used and no force and moments are computed. The coordinate system utilized is that of Figures 1 and 2 of Section II.

Wing Geometry:

Wing and jet configuration are symmetric about the midspan.

Midspan is located at Y = 0.

For zero sideslip, the first control station is located at Y = 0 and the last control station must be located at the starboard wingtip.

For sideslip other than zero, the first control station is located at the port wingtip and the last control station is located at the starboard wingtip.

Fuselage Geometry:

The fuselage nose must be located at X = 0.
The plane of symmetry of the fuselage must be situated at Y = 0.
No control stations may cut through an exhausting jet.

b. **Options**

- **Geometry:** Wing or fuselage
- **Power Configuration:** Power effect, power on or power off
- **Computational Method:** Segment method alone or segment method plus three-dimensional modification
- **Force and Moment:** Computation of integrated force and moment may be exercised or suppressed

2. OPERATING INFORMATION

Core and Time Requirements:

Computer: CDC 6600

Core: 215 K_8 to load
200 K_8 to execute

Time: Approximately 3 minutes for a typical run with NSTA = 11 and MTHET = 36

Additional Requirements: The program requires one intermediate storage tape unit.

3. INPUT DATA

The program requires the input data cards shown in Figure 11. Cards 1 and 2 are required for all computations. Some of the cards of Group A may be omitted depending on the Power Configuration option specified. Additional cards from Group B may be required according to other options specified. Either the w-type or f-type cards are added from Group B depending on the Geometry option.

Card No.	Variable	Format	Description
①	IGEØM	I6	Geometry index if IGEØM = 1 wing; = 2 fuselage
	MØDIN	I6	Modification index if MØDIN = 0 segment method only; = 1 segment method plus 3-D modification

Card No.	Variable	Format	Description
①	JSTØP	I6	Number of iterations if JSTØP { = 0 segment method only; = n iterate n times }
	IDIS	I6	Number of layers in the parallelepiped network residual sources and sinks Limit: IDIS ≤ 4 if MØDIN = 0, IDIS = 1
	JPØWER	I6	Power index if JPØWER { = -1 power off; = 0 power effect; = 1 power on }
	IRECT	I6	Configuration index if IRECT { = 0 rectangular wing; = 1 nonrectangular wing or fuselage }
	IFØRCE	I6	Force index if IFØRCE { = 0 no force and moment computed; = 1 force and moment computed }
②	NSTA	I3	Number of control stations Limit: 8 ≤ NSTA ≤ 16 for fuselage 8 ≤ NSTA ≤ 12 for wing with no sideslip 8 ≤ NSTA ≤ 16 for wing with sideslip
	N	I3	Number of terms used in mapping expansion Limit: N ≤ 12
	NFØUR	I3	Number of terms used in Fourier analysis for boundary functions in segment method and also for downwash correction in 3-D wing modification Limit: NFØUR ≤ 20
	NSYM	I3	Computation index if IGLØM { = 1 NSYM = 1; = 2 NSYM = 0 }

Card No.	Variable	Format	Description
②	MTHET	I3	When NSYM = 0 and BETA = 0, MTHET is the number of equal increments $\Delta\theta$ on the mapping semi-circle. When NSYM = 1 or BETA $\neq$ 0, MTHET is the number of equal increments $\Delta\theta$ on the full mapping circle. Limit: MTHET$\leq$18 when NSYM = 0 and BETA = 0; MTHET$\leq$36 when NSYM = 1 or BETA $\neq$ 0
	UJ	F7.3	Freestream to jet exit velocity ratio
	ALPHA	F7.3	Angle of attack in degrees
	BETA	F7.3	Angle of sideslip in degrees

GROUP A:

Card No.	Variable	Format	Description
①	APART (I)	F12.6	Coordinate of control station. APART (I) = Y (I) for wing; APART (I) = X (I) for fuselage
	R (I)	F12.6	Radius of mapping circle
	DRDX (I)	F12.6	Gradient of R

- If NSYM = 0, only A's appear on the next card

Card No.	Variable	Format	Description
②	A (J,I)	E12.5	Real part of mapping coefficient (J = 1,N)
	B (J,I)	E12.5	Imaginary part of mapping coefficient (J = 1,N)

- If JPØWER = -1, omit cards 3, 4, 5

Card No.	Variable	Format	Description
③	U (I,J)	E12.5	Induced velocity component in X-direction. J = 1, NTHET
④	V (I,J)	E12.5	Induced velocity component in Y-direction. J = 1, NTHET
⑤	W (I,J)	E12.5	Induced velocity component in Z-direction. J = 1, NTHET

where NTHET = MTET+1 if NSYM = 0 and BETA = 0
NTHET = MTHET if NSYM = 1 or BETA $\neq$ 0

- There would now follow sets of cards for I = 2, NSTA

Note: For all Power Configuration options other than JPØWER = -1, all the data cards of Group A are generated for stations I = 1, NSTA by the Jet Flow Field program.

For the Power-Off Configuration, Cards 1 and 2 must be provided at each station. These mapping coefficients, radii and gradients required are obtained from the Mapping Function program.

GROUP B: Additional data cards for further computations

Geometry Option: IGEØM = 1

If MØDIN:
- = 0 and IFØRCE
  - = 0 no further computations
  - = 1 card w3 required
- = 1 and IFØRCE
  - = 0 cards w1 and w2 required
  - = 1 cards w1→w3 required

Card No.	Variable	Format	Description
w1	NBØØL	I6	NBØØL = 0, no modification is imposed on any of the computed velocity components. NBØØL = 1, velocity components, due to residual sources and sinks at the station nearest to the jet are the average values of the computed and interpreted components.
	MEXIT	I6	If BETA = 0, MEXIT = 1. If BETA ≠ 0, MEXIT = station number where jet is located.
w2	MØD	I6	Number of stations where downwash modification is to be effected. Generally: MØD = NSTA-3 if BETA = 0 MØD = NSTA/2-3 if BETA ≠ 0
w3	NDJ	I3	Number of exhausting jets
	DJET	F12.6	Jet exit diameter
	XCG	F12.6	X-coordinate of moment center
	YCG	F12.6	Z-coordinate of moment center
	CHØRD	F12.6	Reference length for nondimensionalizing moment

Geometry Option: IGEØM = 2

if MØDIN = 0 and IFØRCE = 0 no further computations; = 1 cards f3 and f4 are required

if MØDIN = 1 and IFØRCE = 0 cards f1 and f2 are required; = 1 cards f1→f4 are required

Card No.	Variable	Format	Description
(f1)	NJET	I6	NJET = I when the upstream jet is located between stations I and I+1
(f2)	APART(NSTA+1)	F12.6	X-coordinate of fuselage tail
(f3)	NDJ	I3	See definition, card w3
	DJET	F12.6	See definition, card w3
	XCG	F12.6	See definition, card w3
	CHØRD	F12.6	See definition, card w3
(f4)	YTIP	F12.6	Y-coordinate of fuselage nose
	ZTIP	F12.6	Z-coordinate of fuselage nose
	APART(NSTA+1)	F12.6	X-coordinate of fuselage tail
	YTAIL	F12.6	Y-coordinate of fuselage tail
	ZTAIL	F12 6	Z-coordinate of fuselage tail

The optimum manner of choosing control stations along the fuselage or across the wing span is at equally spaced intervals. When this is not possible, it is desirable to avoid large variation in adjacent intervals and cluster of stations at one location.

4. OUTPUT

There are, in general, four groups of output data:

a. Control indices and other input variables: Control indices and other pertinent input data are printed out and identified.

b. Table for geometry: The correspondence between the angular increments on the mapping circle and the rectangular coordinates of each station is listed.

c. **Tables for pressure distribution: The computed pressure coefficients on the surface are tabulated. The first table contains the results obtained by the segment method, which is then followed by table (or tables) to include the three-dimensional modifications.**

d. **Force and moment data: The calculated force and moment data are printed out. Preceding this, the parameters used in three-dimensional modification and for force and moment computations are also identified and listed.**

If options in the input data do not call for three-dimensional modification and the force and moment calculation, Group (c) will contain only one table and Group (d) will not appear.

5. PROGRAMMING INFORMATION

a. Logical Structure

The logical flow chart for the program is shown in Figure 12.

b. Purpose of Subroutines

STRIP — Establishes the appropriate induced velocity field for subroutines VLBØDY or VLWING, calculates pressure coefficients from the output arguments of VLWING or VLBØDY and prints out pressure distribution tables.

VLBØDY — Defines the boundary function, represents it in Fourier series and calculates the velocity components from the complex potential for the fuselage configuration.

VLWING — Similar to VLBØDY but for the wing configuration.

WMØD3 — Determines the strength of residual sources and sinks and modifies the original induced velocity field for the wing configuration.

BMØD3 — Similar to WMØD3, but for the fuselage configuration.

DNWASH — Uses lifting line theory to modify the downwash field.

FMWING — Integrates pressure distribution to give force and moment on a wing.

FMBØDY — Similar to FMWING, but for the fuselage configuration.

THEØ — Expands a given function into a Fourier series.

INTEG — Performs integration of a given function.

SVCØ — Fits a cubic curve through four points.

SVIN — Interpolates this cubic curve.

c. Interdependence of Subroutines

The Calling-Called matrix for the program is shown in Figure 13.

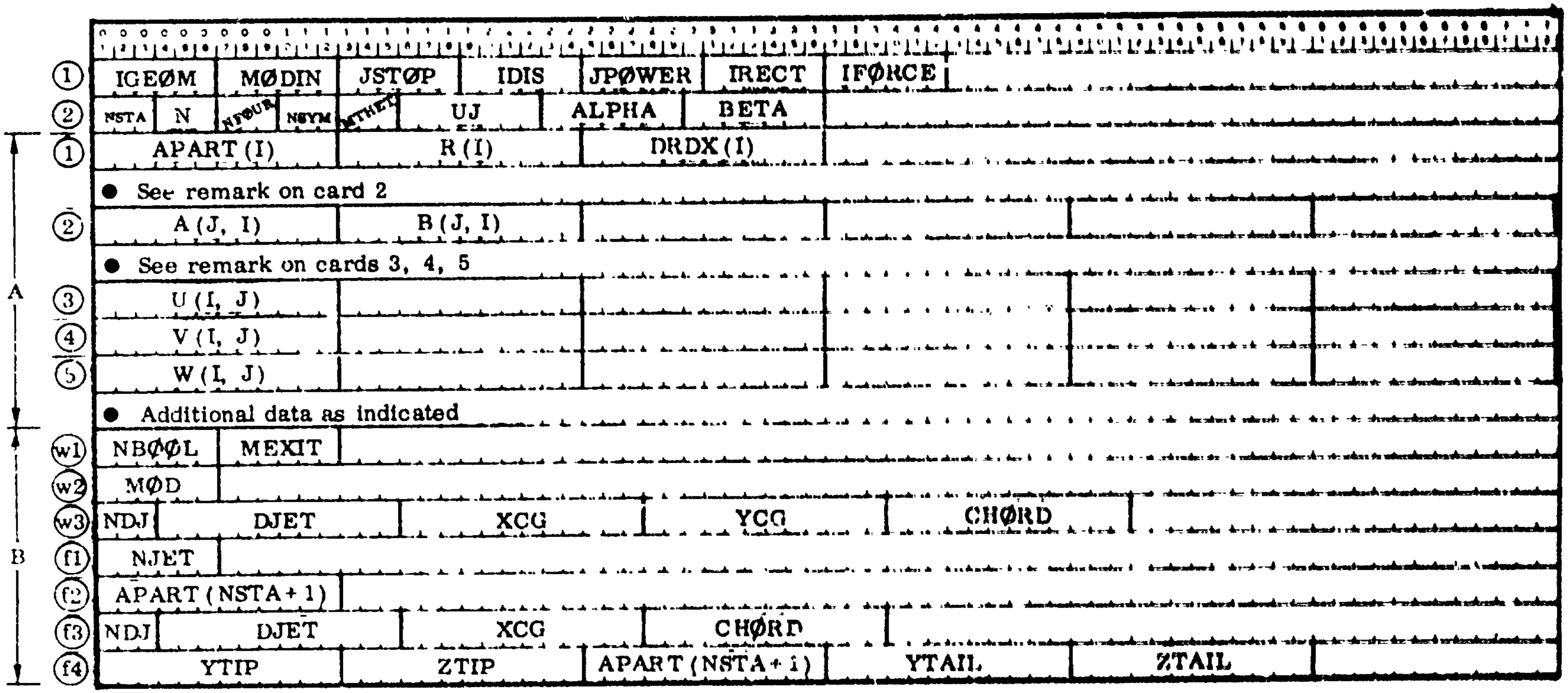

FIGURE 11. INPUT DATA FOR TRANSFORMATION METHOD PROGRAM

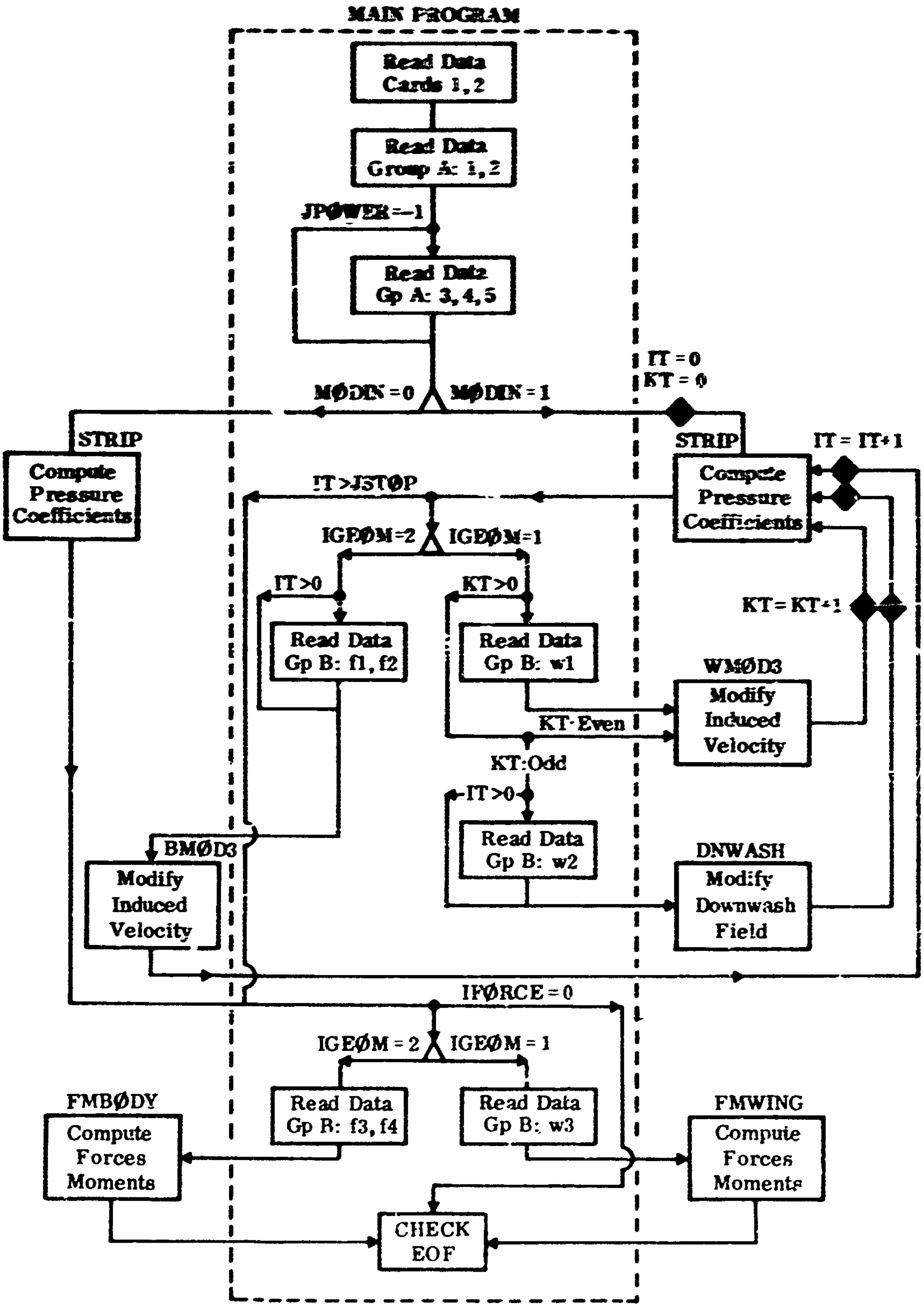

FIGURE 12. LOGICAL FLOW CHART FOR TRANSFORMATION METHOD PROGRAM

Calling \ Called	BMØDS	DNWASH	FMBØDY	FMWING	STRIP	WMØDS	SVCØ	SVIN	VLBØDY	VLWING	THEØ	INTEG
MAIN	●	●	●	●	●	●						
THEØ							●	●				
STRIP									●	●		
VLBØDY											●	
VLWING											●	
INTEG							●	●				
DNWASH												●

FIGURE 13. CALLING–CALLED MATRIX
FOR TRANSFORMATION METHOD PROGRAM

SECTION V

LIFTING SURFACE PROGRAM

The Lifting Surface program is a modified version of the computer program developed by Northrop Corporation under Bu Weps contract NOw-63-0726-C for designing and analyzing subsonic lifting surfaces. The design options have been eliminated and the capability to compute the downwash distribution due to a given camber distribution has been eliminated. The discussion in this section will be restricted to those areas affected by the modifications, primarily the sequence of input cards. While it is intended to provide adequate information to permit utilization of the Lifting Surface program, in conjunction with the Jet Flow Field program, to evaluate power effects on wings, the authoritative documentation on the program remains Northrop Technical Report NOR 64-195 prepared for Bureau of Naval Weapons, Department of Navy, April 1965.

1. DESCRIPTION

The program calculates the pressure loading on a wing due to a specified downwash distribution. It includes provisions for body effect. The program consists of three main components (CHAIN1, CHAIN6, CHAIN7) which may be used together in one continuous operation, or independently.

The first step in the analysis is the calculation of the downwash control point matrix $[D]$, in CHAIN1. The next step is to calculate the least squares inverse of the downwash control point matrix, $[D]^{\psi}$ in CHAIN6. This may be done in a continuous operation following the computation of $[D]$, in which case $[D]$ will be read off intermediate storage tape. CHAIN6 may also be used independently in which case the downwash control point matrix $[D]$ is supplied to the program on punched cards. However, it is preferable to compute $[D]$ and $[D]^{\psi}$ in a continuous operation, in order to maintain maximum accuracy.

The downwash control point matrix $[D]$ and its least squares inverse $[D]^{\psi}$, depend on the planform, the location of the downwash control points, and the number of terms in the loading series. Once calculated, $[D]^{\psi}$ forms an input to the third main component of the program, CHAIN7, which computes the pressure loading. The downwash control point matrix $[D]$ and its least squares inverse $[D]^{\psi}$ are not recomputed as long as the planform, control point locations and the size of the pressure loading series are not changed. The least squares inverse $[D]^{\psi}$ may be retained in punched card form to serve as input to CHAIN7 for additional studies of pressure loadings on the same wing.

Thus the third component of the program, CHAIN7, may be called directly by the inversion program or used separately. The principal information required is: the least squares inverted downwash control point matrix, the wing planform geometry and the downwash distribution. In a continuous operation, the least squares inverted downwash control point matrix will be read off intermediate storage tape. When CHAIN7 is used independently, $[D]^{\psi}$ is supplied to the program on punched cards. CHAIN7 calculates the overall and local aerodynamic coefficients and the pressure loading distribution at a set of specified pressure control points. The overall moment coefficients are referred to an axis located at one quarter of the mean aerodynamic chord. The program is designed to analyze an unlimited number of downwash distributions for the one downwash control point matrix $[D]$. The body effect on the downwash distribution will be included by the program if the spanwise location of the edge of the fuselage is specified. If the body effect is to be omitted, the spanwise location is made zero.

a. Restrictions

The program is applicable to continuous surfaces of arbitrary planform and no interference effects such as slots, ground effects, large dihedral angles or end plates are included.

Downwash control points must not be located at or near the leading edge, since the cotangent elements of $[D]$ would become excessively large and dominate in the solution for the pressure coefficient matrix $[A]$.

Due to the computing techniques utilized, downwash control points must not be located at discontinuities in the planform and at flap hinge lines.

b. Options

- Execute CHAIN1 to obtain the downwash control point matrix $[D]$
- Execute CHAIN6 independently to obtain the least square inverse of the downwash control point matrix, $[D]^{\psi}$

- Execute CHAIN7 independently to obtain the aerodynamic coefficients and the pressure loading distribution
- Execute CHAIN1 and CHAIN6 in a continuous manner to obtain $[D]^{\psi}$
- Execute CHAIN1, CHAIN6 and CHAIN7 in a continuous manner to obtain the aerodynamic coefficients and the pressure loading distribution

Punch controls to obtain $[D]$ or $[D]^{\psi}$ in card form, when execution is not in a continuous manner, are available and will be discussed as part of the input.

2. OPERATING INFORMATION

Core and Time Requirements:

Computer:	CDC 6600
Core:	$124K_8$ to load $107K_8$ to execute
Time:	Approximately 2.5 minutes for a typical run with a downwash control point matrix $[D] = [100 \times 36]$

Additional Requirements: The program requires two intermediate storage tape units.

3. INPUT DATA

A typical wing with two geometric regions is shown in Figure 14. The wing dimensions must be normalized by the wing semispan before specifying data. Only data for the starboard wing are specified since the wing is considered to be symmetric.

The input data required are shown in Figure 15. The first card controls which of the three main components are to be executed. The other cards, sequentially, form the input to CHAIN1, CHAIN6 and CHAIN7. They are grouped in this manner in Figure 15. They are described in detail below.

Card No.	Variable	Format	Description
①	ISTART	I5	Indicates where execution of the program is to begin If ISTART = 1 start with CHAIN1 = 2 start with CHAIN6 = 3 start with CHAIN7

Card No.	Variable	Format	Description
①	ISTØP	I5	Indicates where execution of the program is to stop If ISTOP = 1 stop after CHAIN1; = 2 stop after CHAIN6; = 3 stop after CHAIN7

CHAIN1 : Computation of downwash control point matrix

Card No.	Variable	Format	Description
①	ARRAY	12A6	Title card for CHAIN1
②	NS	I5	Number of stations on semispan where downwash control points are located
	M	I5	Number of spanwise modes to be used in pressure loading series
	N	I5	Number of chordwise modes, including the flap modes, to be used in pressure loading series Limitation: $M \times N \leq 36$
	NEED	I5	Indicates whether or not $\cot\theta/2$ mode is to be used If NEED = 0 don't use $\cot\theta/2$ mode; = 1 use $\cot\theta/2$ mode
	NFLAP	I5	Number of leading and trailing edge flaps
	NPR	I5	Print control for [D] If NPR = 0 don't print; = 1 print
	NPU	I5	Punch control for [D] If NPU = 0 don't punch; = 1 punch
	NAY	I5	Intermediate print control If NAY = 0, no intermediate printout; = 1, intermediate printout
	NØLED	I5	Number of leading edge discontinuities (including root and tip positions)
	NØTED	I5	Number of trailing edge discontinuities (including root and tip positions)

Card No.	Variable	Format	Description
③	SPACE	F10.0	Indicates how downwash control points are located chordwise at the spanwise control stations If SPACE { ≥.02 the value is used to space points equidistant; = 0 must specify chordwise locations }
	FMACH	F10.0	Mach number
	F	F10.0	Root semichord
④	YSTAT(I)	F10.0	Spanwise locations of downwash control points. I = 1, NS.
⑤	FLPØS(I)	F10.0	Chordwise location of the flap hinge line in percent of chord. I = 1, NFLAP
⑥	AMLE(I)	F10.0	Tangents of the sweepback angles of the leading edges of the geometric regions. I = 1, NØLED-1
⑦	AMTE(I)	F10.0	Tangents of the sweepback angles of the trailing edges of the geometric regions. I = 1, NØTED-1
⑧	YLEAD(I)	F10.0	Spanwise locations of leading edge discontinuities. I = 1, NØLED
⑨	YTRAIL(I)	F10.0	Spanwise locations of trailing edge discontinuities. I = 1, NØTED

- If SPACE ≠ 0, omit cards 10, 11

Card No.	Variable	Format	Description
⑩	NCP(I)	I5	Number of downwash control points at each spanwise station. I = 1, NS
⑪	XDWASH(J, I)	F6.0	Chordwise locations of downwash control points at each spanwise station, in fraction of chord. J = 1, NCP(I).

- There now follow sets, I = 2, NS.

Card No.	Variable	Format	Description
●	If NAY = 0, omit card 12		
12	NAY3	I5	Additional print controls
	NAY4	I5	If NAYI = 0 no additional printout; = 1 additional printout
	NAY5	I5	
	NAY6	I5	

CHAIN6: Computation of least squares inverse of downwash control point matrix

Card No.	Variable	Format	Description
1	ARRAY	12A6	Title card for CHAIN6
2	NROW	I5	Number of rows in downwash control point matrix, or number of control points contained in [D]
	NCØL	I5	Number of columns in downwash control point matrix [D]. This is the product of chordwise and spanwise pressure modes.
	NREAD	I5	Indicates if [D] is to be read from intermediate storage tape as in a continuous operation or from card input If NREAD = 0 read from tape; = 1 read card input
	NPR	I5	Print control for $[D]^{\psi}$ If NPR = 0 don't print; = 1 print
	NPU	I5	Punch control for $[D]^{\psi}$ If NPU = 0 don't punch; = 1 punch
	NAY	I5	See definition, card 2, CHAIN1

◆ [If NREAD = 1, the punched matrix [D] is inserted at this point. This is the output obtained from CHAIN1 when operating in a noncontinuous manner.]

CHAIN7: Computation of aerodynamic coefficients

Card No.	Variable	Format	Description
1	ARRAY	12A6	Title card for CHAIN7

Card No.	Variable	Format	Description
②	N	I5	See definition, card 2, CHAIN1
	M	I5	See definition, card 2, CHAIN1
	NS	I5	See definition, card 2, CHAIN1
	NRØW	I5	See definition, card 2, CHAIN6
	NETA	I5	Number of spanwise locations where chordwise pressure loadings are to be calculated
	NDISC	I5	Number of wing discontinuities (including root and tip points).
	NFLAP	I5	See definition, card 2, CHAIN1
	NAY	I5	See definition, card 2, CHAIN1
	NPSI	I5	Number of chordwise points at which pressure loading is computed Limit: $NPSI \leq 50$
③	NALFA	I5	Number of angles of attack treated Limit: $NALFA \leq 20$
	NEPSLN	I5	Indicates number of EPSLN's to be read on card
	NEED	I5	See definition, card 2, CHAIN1
	NREAD1	I5	Indicates if $[D]^{\psi}$ is to be read from intermediate storage tape as in a continuous operation or from card input If NREAD1 { = 0 read from tape; = 1 read from cards
	NREAD2	I5	Indicates if the downwash matrix $[W]$ is read from cards. Due to the modifications, eliminating the capability to compute the downwash distribution from the camber distribution, NREAD2 MUST BE>ZERO.
	NW	I5	Number of downwash distributions to be considered.

Card No.	Variable	Format	Description
(4)	F	F10.0	See definition, card 3, CHAIN1
	SPACE	F10.0	See definition, card 3, CHAIN1
	YF	F10.0	Spanwise location of edge of fuselage
	DPSI	F10.0	Indicates how points, where pressure loading is to be computed, are located chordwise at all the ETA's If DPSI { $\leq .02$ the value is used to space the points equidistant; < 0 must specify chordwise locations }
(5)	YSTAT(I)	F7.0	See definition, card 4, CHAIN1
(6)	ETA(I)	F7.0	Spanwise locations where pressure loading distributions are calculated I = 1, NETA
(7)	EPSLN(I)	F7.0	Angles of incidence between ℄ of fuselage and wing root chord in degrees. I = 1, NEPSLN
(8)	ALFA(I)	F7.0	Angles of attack of fuselage in degrees I = 1, NALFA
(9)	FLPØS(I)	F7.0	See definition, card 5, CHAIN1
(10)	CHORD(I)	F7.0	Chord at spanwise discontinuities. I = 1, NDISC
(11)	WHY(I)	F7.0	Location of spanwise discontinuities. I = 1, NDISC
(12)	DELTA(I)	F7.0	Chordwise distance from root leading edge to leading edge at spanwise discontinuities

- If SPACE $\neq 0$, omit card 13

Card No.	Variable	Format	Description
(13)	NCP(I)	I2	See definition, card 10, CHAIN1

- If DPSI > 0, omit card 14

Card No.		Variable	Format	Description
⑭	[	PSI(I)	F7.0	Chordwise locations of points where pressure loading is to be computed in fraction of chord
◆	[	If NREAD1 = 1, the punched matrix $[D]^{\psi}$ is inserted at this point. This is the output obtained from CHAIN6 when operating in a noncontinuous manner.		
⑮	[	W(I, J)	E14.7	Tangent of the downwash angle at the downwash control points. J = 1, NCP(I)

- There now follow sets, I = 2, NS.

4. OUTPUT

Depending on the options specified both printed and punched output may be obtained.

a. Printed Output

CHAIN1 prints pertinent input information to identify the problem. CHAIN6, which inverts the matrix [D] prints out the determinant of the unit matrix as a check on the numerical accuracy. CHAIN7 prints geometric parameters of the wing (mean aerodynamic chord, etc.). It also prints out the overall and local aerodynamic coefficients and the pressure loading at the spanwise and chordwise locations specified.

b. Punched Output

CHAIN1 may generate the downwash control point matrix [D] in punched form to serve as input to CHAIN6 when the components of the program are not executed in a continuous manner.

CHAIN6 may generate the least squares inverse of the downwash control matrix $[D]^{\psi}$ to serve as input to CHAIN7 when that component of the program is being executed independently.

5. PROGRAMMING INFORMATION

a. Logical Structure

The logical flow chart for the modified version of the program is shown in Figure 16.

b. Interdependence of Subroutines

The Calling-Called matrix for the program is shown in Figure 17.

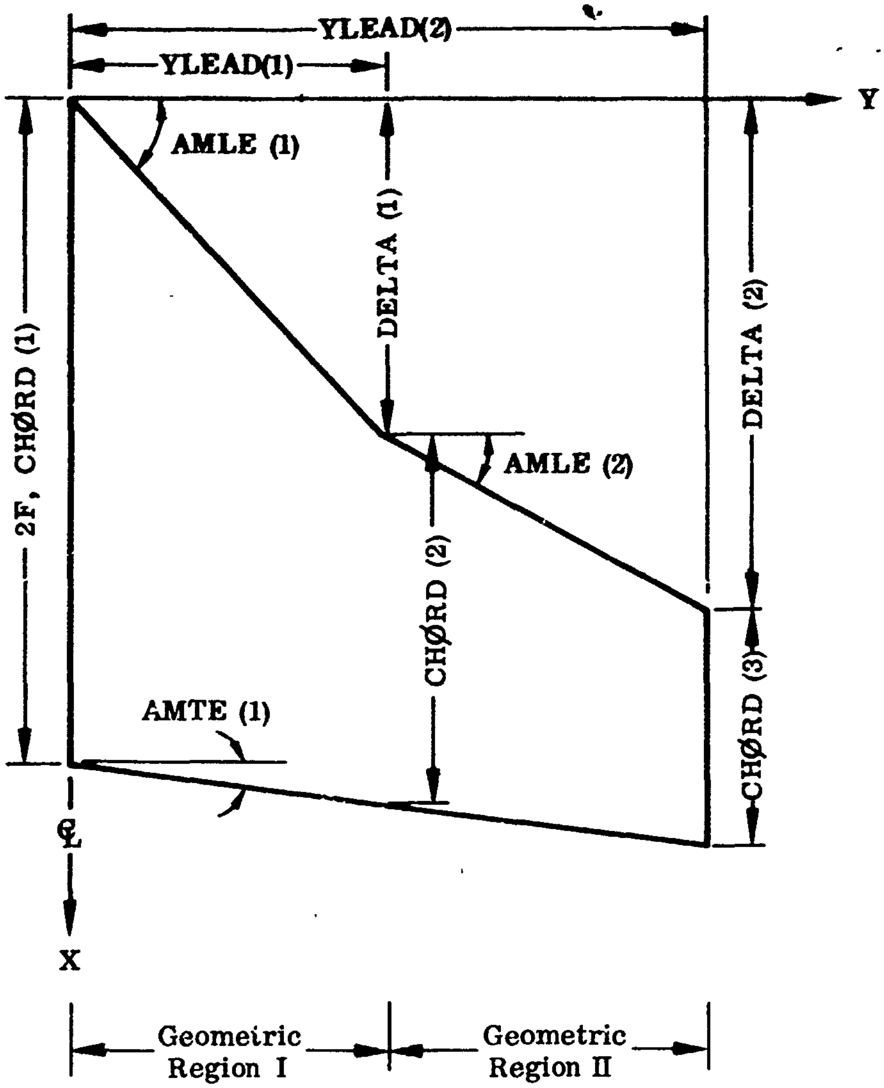

FIGURE 14. COORDINATE SYSTEM FOR LIFTING SURFACE PROGRAM

Chain	Card										
	1	ISTART	ISTØP								
CHAIN1	1	ARRAY									
CHAIN1	2	NS	M	N	NFED	NFLAP	NPR	NPU	NAY	NØLED	NØTED
CHAIN1	3	SPACE		FMACH		F					
CHAIN1	4	YSTAT (I)									
CHAIN1	5	FLPØS (I)									
CHAIN1	6	AMLE (I)									
CHAIN1	7	AMTE (I)									
CHAIN1	8	YLEAD (I)									
CHAIN1	9	YTRAIL (I)									
CHAIN1		● See remark on cards 10, 11									
CHAIN1	10	NCP(I)									
CHAIN1	11	XDWASH(J, I)									
CHAIN1		● Additional data as indicated									
CHAIN1		● See remark on card 12									
CHAIN1	12	NAY3	NAY4	NAY5	NAY6						
CHAIN6	1	ARRAY									
CHAIN6	2	NRØW	NCØL	NREAD	NPR	NPU	NAY				

◆ [If NREAD = 1, the punched matrix [D] is inserted at this point. This is the output obtained from CHAIN1 when operating in a noncontinuous manner.]

CONTINUED ON NEXT PAGE

FIGURE 15. INPUT DATA FOR LIFTING SURFACE PROGRAM

①	ARRAY								
②	N	M	NS	NRØW	NETA	NDISC	NFLAP	NAY	NPSI
③	NALFA	NEPSLN	NEED	NREAD1	NREAD2	NW			
④	F	SPACE	YF	DPSI					
⑤	YSTAT(I)								
⑥	ETA(I)								
⑦	EPSLN(I)								
⑧	ALFA(I)								
⑨	FLPØS(I)								
⑩	CHØRD(I)								
⑪	WHY(I)								
⑫	DELTA(I)								
	● See remark on card 13								
⑬	NCP(I)								
	● See remark on card 14								
⑭	PSI (I)								
	◆ If NREAD1 = 1, the punched matrix $[D]^{\psi}$ is inserted at this point. This is the output obtained from CHAIN6 when operating in a noncontinuous manner.								
⑮	W(I, J)								
	● Additional data as indicated								

CHAIN7

FIGURE 15. (Concluded)

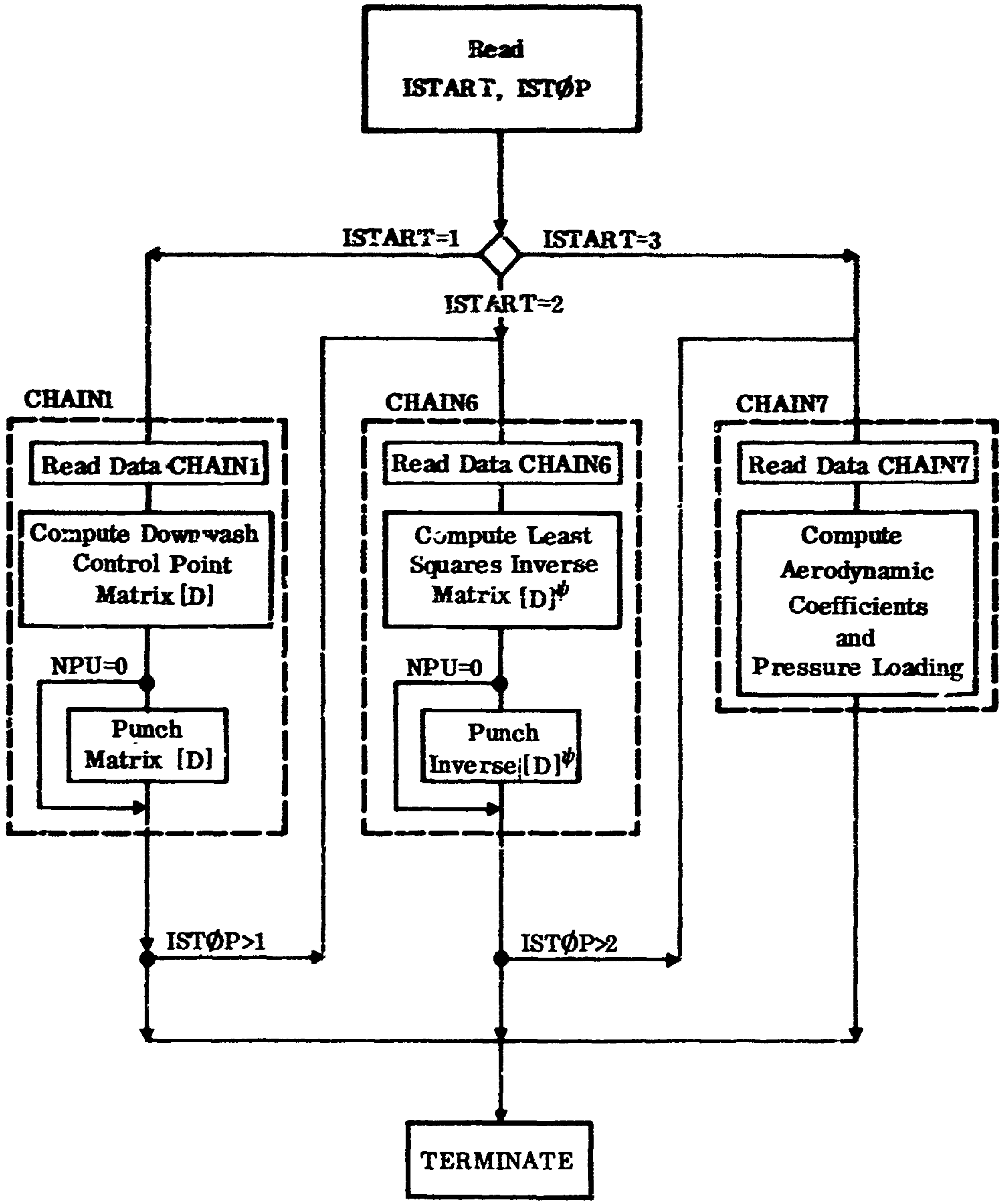

FIGURE 16. LOGICAL FLOW CHART FOR LIFTING SURFACE PROGRAM

Calling \ Called	CHAIN1	CHAIN6	CHAIN7	FKERNL	FNUD	MATRØW	MPRINT	PINVRS	AERØ	FPMI	FRMI	FSQM	MATINV	PRESSR
MAIN	●	●	●											
CHAIN1				●	●	●	●							
CHAIN6							●	●						
CHAIN7									●	●	●			
AERØ												●		
PINVRS													●	
MATRØW														●

FIGURE 17. CALLING–CALLED MATRIX FOR LIFTING SURFACE PROGRAM

SECTION VI

NONLINEAR BODY AERODYNAMICS PROGRAM

1. DESCRIPTION

The nonlinear body aerodynamics computer program combines slender body theory and viscous cross flow theory to obtain the aerodynamic coefficients for an arbitrary body. The program computes the coefficients C_N, C_m, C_Y, C_n, and C_ℓ in body axes as functions of resultant angle of attack α, roll angle ϕ, pitching velocity q and yawing velocity r. The coefficients are printed out with the slender body contribution and the viscous contribution listed separately. The rolling moment coefficient C_ℓ does not have a viscous contribution calculated for it, since it is not possible to formulate a satisfactory model for it. Zero is printed out for the viscous contribution.

It is assumed that a mapping is known for the sections along the body and that the coefficients of the mapping are continuous functions of axial distance along the body. The method of obtaining the mapping is described in Volumes I and II. An approximate method has also been described and is preferred where simplicity and ease of use are desired.

2. OPERATING INFORMATION

Core and Time Requirements:

Computer:	CDC 6600
Core:	35.5 K_8 to load
	22.1 K_8 to execute
Time:	Approximately .1 minutes for a run with nine angles of attack and one set of ϕ, q and r.

Additional Requirements: None

3. INPUT DATA

The coordinate system utilized by the program is that shown in Figure 2 of Section II.

The input data cards required by the program are shown in Figure 18. The input cards of Group A describe the body. The cards of Group B give the flight conditions and reference dimensions for the computation of the aerodynamic coefficients. The input cards are grouped in this manner and discussed in detail below.

Card No	Variable	Format	Description
			GROUP A: Input data describing the body.
①	MZT	I3	The maximum number of mapping coefficients of any station input to the program Limit: MZT ≤ 12
	NX	I3	Number of input data stations along body Limit: NX ≤ 40
②	X1 (I)	E12.5	Station along body. I = 1, NX
③	RB1 (I)	E12.5	Radius of mapping circle r_c at input station. I = 1, NX
④	DRDX1 (I)	E12.5	Derivative of the mapping circle radius with respect to X, at input station. I = 1, NX
⑤	S1 (I)	E12.5	Cross sectional area S at input station. I = 1, NX
⑥	DSDX1 (I)	E12.5	Derivative of cross sectional area with respect to X at input station. I = 1, NX
⑦	CDCY1 (I)	E12.5	Cross sectional drag area per unit length in the vertical direction, $C_D c_y$. I = 1, NX
⑧	CDCL1 (I)	E12.5	Cross sectional drag area per unit length in the lateral direction, $C_D c_z$. I = 1, NX
⑨	NZ	I3	Number of terms in mapping function at station I. If NZ = 0, MZT will be used. Limit: NZ ≤ 12

Card No.	Variable	Format	Description
(9)	ISM	I3	Symmetry indicator at station I. If ISM {=0, symmetrical cross section; =1, unsymmetrical cross section

- If MZT > 1 and if ISM {=0, include cards 10, 11; =1, include cards 10a, 11a

Card No.	Variable	Format	Description
(10)	REAL1 (J, I)	E12.5	Alternating real and imaginary coefficients of mapping function for symmetrical section. If NZ {=0, J=1, MZT-1; >1, J=1, NZ-1
(11)	REPR1 (J, I)	E12.5	Derivatives of mapping function coefficients with respect to X for symmetrical sections. If NZ {=0, J=1, MZT-1; >1, J=1, NZ-1
(10) a	REAL1 (J, I)	E12.5	Real component of coefficient of mapping function for unsymmetrical section. If NZ {=0, J=1, MZT-1; >1, J=1, NZ-1
	XMAG1 (J, I)	E12.5	Imaginary component of coefficient of mapping function for unsymmetrical section.
(11) a	REPR1 (J, I)	E12.5	Derivative of real component of coefficient of mapping function for unsymmetrical section. If NZ {=0, J=1, MZT-1; >1, J=1, NZ-1
	XMPR1 (J, I)	E12.5	Derivative of imaginary component of coefficient of mapping function for unsymmetrical section.

- There now follow sets of cards, I = 2, NX

Card No.	Variable	Format	Description

GROUP B: Input data specifying flight conditions and reference dimensions for the computation of the aerodynamic coefficients.

Card No.	Variable	Format	Description
①	CØMNT	18A4	Comment card
②	REF	F10.4	Reference length ℓ_r
	SREF	F10.4	Reference area
	CG	F10.4	X-coordinate of the center of gravity and moment center
	DX1	F10.4	Incremental step size for integrating along the X-axis
③	NA	I2	Number of angles of attack at which coefficients are to be computed Limit: NA ≤ 18
	NP	I2	Number of roll angles for which coefficients are to be computed. Limit: NP ≤ 9
	NQ	I2	Number of pitching velocities for which coefficients are to be computed Limit: NQ ≤ 9
	NR	I2	Number of yawing velocities for which coefficients are to be computed. Limit: NR ≤ 9
④	ALPHA1 (I)	F8.4	Angle of attack, in degrees. I = 1, NA
⑤	PHI1 (I)	F8.4	Roll angle, in degrees. I = 1, NP
⑥	Q1 (I)	F8.4	Pitching velocity, $\frac{q\ell_r}{2U_\infty}$, in radians. I = 1, NQ
⑦	R1 (I)	F8.4	Yawing velocity, $\frac{r\ell_r}{2U_\infty}$, in radians. I = 1, NR

4. OUTPUT

Figure 19 shows sample output for the nonlinear body aerodynamics program. The title card is reproduced on the first line. The second line shows the roll angle PHI (ϕ, in degrees), the pitching velocity Q ($\frac{q l_r}{2U_\infty}$, in rads) and yawing velocity R ($\frac{r l_r}{2U_\infty}$, in rads) at which the aerodynamic coefficients are to be computed.

The program then tabulates the computed coefficients. The table is headed to identify the angle of attack, ALFA, and the aerodynamic coefficients being computed, CN (C_N), CM (C_m), CY (C_y), CEM (C_n) and CRM (C_ℓ). For each angle of attack specified in degrees, a potential set of coefficients and a viscous set of coefficients is listed. The complete coefficients can be obtained by adding the two parts.

If more than one PHI, Q or R has been specified as part of the input, the program will repeat the tabulation.

5. PROGRAMMING INFORMATION

a. Logical Structure

The logical flow chart for the program is shown in Figure 20.

b. Purpose of Subroutines

DATA – Reads and stores the portion of the input data dealing with the description of the body

CØEFF – This routine sets a step size for integrating forces and moments along the body. It calls LØCVAL which returns body parameters at the desired station and then calls FØRCE which computes pieces of the coefficients up to the given station. When this routine reaches the rear end of the body, enough information is available for the main program to compute the potential coefficients.

LØCVAL – Obtains interpolated body data at the station required by CØEFF

AINTRP – Interpolation routine. Determines a body parameter as a function of the axial distance.

FØRCE – Computes parts of the potential force and moment coefficients up to the station at which it is called. When it is called at the rear end of the body, it determines the parameters needed for computing the rolling moment .

VISC – **Computes the viscous contributions to C_N, C_m, C_y and C_n by dividing the body into increments and integrating the viscous equations along the body.**

c. Interdependence of Subroutines

The Calling-Called matrix for the program is shown in Figure 21.

A	1	MZT	NX						
	2	X1 (I)							
	3	RB1 (I)							
	4	DRDX1 (I)							
	5	S1 (I)							
	6	DSDX1 (I)							
	7	CDCY1 (I)							
	8	CDCL1 (I)							
	9	NZ	ISM						
		● See remark on cards 10, 11, 10a, 11a							
	10	REAL1 (J, I)							
	11	REPR1 (J, I)							
	10a	REAL1 (J, I)	XMAG1 (J, I)						
	11a	REPR1 (J, I)	XMPR1 (J, I)						
		● Additional data as indicated							
B	1	CØMNT							
	2	REF	SREF	CG	DX1				
	3	NA	NP	NQ	NR				
	4	ALPHA1 (I)							
	5	PHI1 (I)							
	6	Q1 (I)							
	7	R1 (I)							

FIGURE 18. INPUT DATA FOR NONLINEAR BODY AERODYNAMICS PROGRAM

V/STOL TEST MODEL DATA. 12/2/70.

PHI= 90.000 Q= 0.0 R= 0.0

ALPHA		CN	CM	CY	CEM	CRM
0.0	POTENTIAL	3.4009E-04	-2.2614E-02	3.4001E-15	-2.1316E-14	1.9517E-17
	VISCOUS	0.0	0.0	0.0	0.0	0.0
5.0000	POTENTIAL	3.3751E-04	-2.2442E-02	-2.6378E-04	-7.1750E-02	2.3563E-03
	VISCOUS	2.5569E-16	-9.1357E-17	-6.9620E-03	1.7322E-03	0.0
10.0000	POTENTIAL	3.2983E-04	-2.1932E-02	-5.1954E-04	-1.4132E-01	4.5879E-03
	VISCOUS	1.0150E-15	-3.6265E-16	-2.7636E-02	6.8764E-03	0.0
15.0000	POTENTIAL	3.1731E-04	-2.1099E-02	-7.5952E-04	-2.0660E-01	6.5785E-03
	VISCOUS	2.2548E-15	-8.0565E-16	-6.1395E-02	1.5276E-02	0.0
20.0000	POTENTIAL	3.0031E-04	-1.9968E-02	-9.7642E-04	-2.6559E-01	8.2275E-03
	VISCOUS	3.9375E-15	-1.4069E-15	-1.0721E-01	2.6676E-02	0.0
25.0000	POTENTIAL	2.7935E-04	-1.8575E-02	-1.1636E-03	-3.1652E-01	9.4568E-03
	VISCOUS	6.0119E-15	-2.1481E-15	-1.6369E-01	4.0730E-02	0.0
30.0000	POTENTIAL	2.5507E-04	-1.6960E-02	-1.3155E-03	-3.5783E-01	1.0216E-02
	VISCOUS	8.4150E-15	-3.0067E-15	-2.2913E-01	5.7010E-02	0.0
35.0000	POTENTIAL	2.2820E-04	-1.5174E-02	-1.4274E-03	-3.8827E-01	1.0485E-02
	VISCOUS	1.1074E-14	-3.9567E-15	-3.0152E-01	7.5024E-02	0.0
40.0000	POTENTIAL	1.9957E-04	-1.3270E-02	-1.4960E-03	-4.0691E-01	1.0276E-02
	VISCOUS	1.3908E-14	-4.9692E-15	-3.7868E-01	9.4222E-02	0.0
45.0000	POTENTIAL	1.7004E-04	-1.1307E-02	-1.5190E-03	-4.1319E-01	9.6317E-03
	VISCOUS	1.6830E-14	-6.0134E-15	-4.5826E-01	1.1402E-01	0.0

FIGURE 19. SAMPLE OUTPUT FOR NONLINEAR BODY AERODYNAMICS PROGRAM

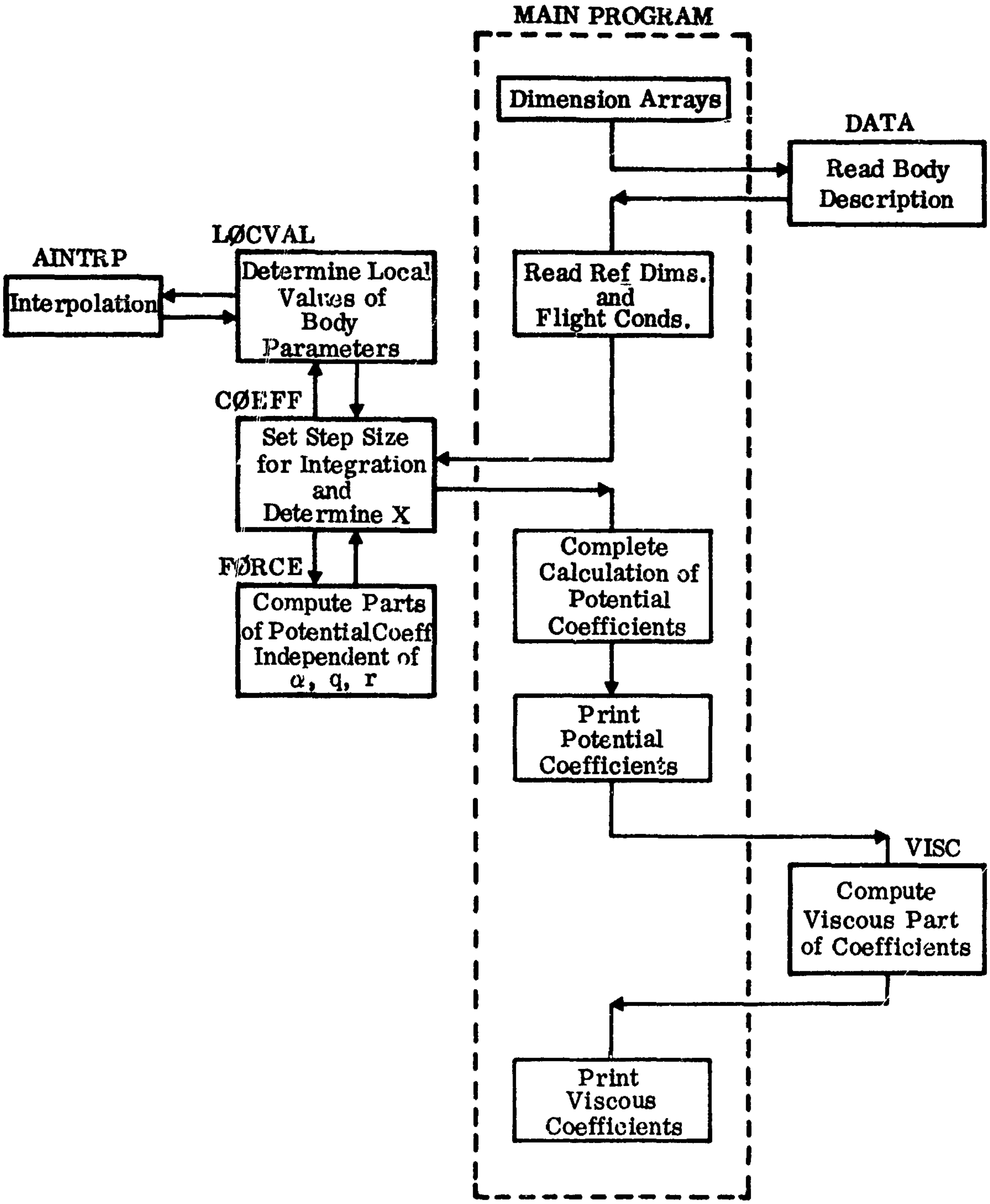

FIGURE 20. LOGICAL FLOW CHART FOR NONLINEAR BODY AERODYNAMICS PROGRAM

Calling \ Called	CØEFF	DATA	VISC	AINTRP	FØRCE	LØCVAL
MAIN	●	●	●			
LØCVAL				●		
CØEFF					●	●

FIGURE 21. CALLING–CALLED MATRIX FOR NONLINEAR BODY AERODYNAMICS PROGRAM

SECTION VII

NONLINEAR WING AERODYNAMICS PROGRAM

1. DESCRIPTION

The nonlinear wing aerodynamics program determines the aerodynamic coefficients C_N, C_m, and C_ℓ in a body axis coordinate system as functions of angle of attack α, sideslip angle β, pitching velocity q, rolling velocity p and yawing velocity r. The theoretical background for the method is described in Volume I and the application to a sample problem is given in Volume II.

2. OPERATING INFORMATION

Core and Time Requirements:

Computer:	CDC 6600
Core:	43.4 K_8 to load
	30.2 K_8 to execute
Time:	Approximately .3 minutes for a run with two angles of attack and two iterations per angle of attack

Additional Requirements: None

3. INPUT DATA

The coordinate system utilized to describe the input is that of Figure 14 of Section V. However, all dimensions are nondimensionalized with respect to the wing root chord. Only the data for the starboard panel of the wing are specified, since the wing is assumed to be geometrically symmetric.

The input data cards required by the program are shown in Figure 22 and are described in detail below.

Card No.	Variable	Format	Description
①	ALPHA	F9.5	Initial value for the wing angle of attack α, in degrees
	BETA	F9.5	Angle of sideslip β, in degrees
	DALPHA	F9.5	Step size of alpha, in degrees
②	ETA0	F9.5	Y-coordinate of wing root chord
	ETAB	F9.5	Y-coordinate of wing tip chord
	TR	F9.5	Wing taper ratio
	TNLE	F9.5	Tangent of sweepback angle of wing leading edge
③	P	F9.5	Rolling velocity, $\frac{p\ell_r}{2U_\infty}$, in radians
	Q	F9.5	Pitching velocity, $\frac{q\ell_r}{2U_\infty}$, in radians
	R	F9.5	Yawing Velocity, $\frac{r\ell_r}{2U_\infty}$, in radians
④	REFL	F9.5	Reference length, ℓ_r, in percent of root chord
	XCG	F9.5	X-coordinate of pitching velocity axis
	ZCG	F9.5	Z-coordinate of yawing velocity axis
⑤	CD	F9.5	Drag coefficient of wing section at $\alpha = 90^o$
	CDXPØS	F9.5	X-coordinate of line of action of section drag at $\alpha = 90^o$, in percent of root chord
⑥	NSTA	I6	Number of circulation control stations on one wing panel Limit: NSTA ≤ 10
	NDWSH	I6	Number of downwash control stations on one wing panel NDWASH must be set equal to NSTA-1.

Card No.	Variable	Format	Description
7	NALPHA	I6	**Number of angles of attack**
	NIT	I6	**Number of iterations on the effective angle of attack for each α**
8	NSYM	I6	**Symmetry indicator** If NSYM { =0, symmetrical wing loading; =1, asymmetrical wing loading }
9	ETA (I)	F9.5	Y-coordinate of circulation control station, in fraction of root chord. I = 1, NSTA
10	ETADW (I)	F9.5	Y-coordinate of downwash control station, in fraction of root chord. I = 1, NDWASH Use same values as ETA(I)
11	XI0 (1)	F9.5	X-coordinate of the inboard extremity of the leading lifting line, in fraction of root chord
	TN (1)	F9.5	Tangent of the sweepback angle of the leading lifting line
12	XI0 (2)	F9.5	X-coordinate of the inboard extremity of the aft lifting line
	TN (2)	F9.5	Tangent of the sweepback angle of the aft lifting line
13	XI0 (3)	F9.5	X-coordinate of the inboard extremity of the downwash control line
	TN (3)	F9.5	Tangent of the sweepback angle of the downwash control line
14	ALPHEF (I)	F9.5	Estimate of the effective angle of attack for each downwash control station. I = 1, NDWSH
15	AL (I)	F9.5	Angles of attack for which the weighting of the circulation between the two lifting lines is to be input. I = 1, 10 (See Vol II, p.167)
16	WGHT (I)	F9.5	Values of the weighting function at the α's given in card 15. I = 1, 10

4. OUTPUT

The angles of attack and sideslip are printed out, followed by P ($\frac{p\ell_r}{2U_\infty}$, in radians), Q ($\frac{q\ell_r}{2U_\infty}$, in radians), and R ($\frac{r\ell_r}{2U_\infty}$, in radians). The spanwise loading and effective angle of attack are then printed out.

The normal force coefficient (normalized by wing area and freestream dynamic pressure) and body axis moment coefficients (normalized by the reference length ℓ_r) are printed out.

This set of output (except for angles of attack and sideslip) is repeated for the number of iterations on effective angle of attack, specified in the input.

The above output is repeated for the number of angles of attack specified.

5. PROGRAMMING INFORMATION

a. Logical Structure

The logical flow chart for the program is shown in Figure 23.

b. Purpose of Subroutines

WGT – Determines weighting of circulation between the two lifting lines

GAUSS – Performs numerical integration, using 16 point Gaussian quadrature

LGRANG – Determines expression for the total circulation as a function of values at the circulation control points, using Lagrange's method.

LLINE – Determines the influence coefficients matrix for the downwash due to the bound vorticity

TRVØRT – Evaluates the influence coefficients matrix for the downwash due to the trailing vorticity

MATINV – Calculates the inverse of the influence coefficients matrix

FMINT – Integrates the span loading to determine the body axes force and moments

FØRM1 – Evaluates the integrand required in LLINE

FØRM2 – Evaluates integrand required in TRVØRT

FØRM3 – Evaluates integrand required in TRVØRT

c. Interdependence of Subroutines

The Calling–Called matrix for the program is shown in Figure 24

Card	Columns 1–9	Columns 10–18	Columns 19–27	Columns 28–36	Columns 37–45	Columns 46–54	Columns 55–63	Columns 64–72
1	ALPHA	BETA	DALPHA					
2	ETA0	ETAB	TR	TNLE				
3	P	Q	R					
4	REFL	XCG	ZCG					
5	CD	CDXPØS						
9	ETA(I)							
10	ETADW(I)							
11	XI0 (1)	TN (1)						
12	XI0 (2)	TN (2)						
13	XI0 (3)	TN (3)						
14	ALPHEF(I)							
15	AL (I)							
16	WGHT (I)							

6	NSTA	NDWSH
7	NALPHA	NIT
8	NSYM	

FIGURE 22. INPUT DATA FOR NONLINEAR WING AERODYNAMICS PROGRAM

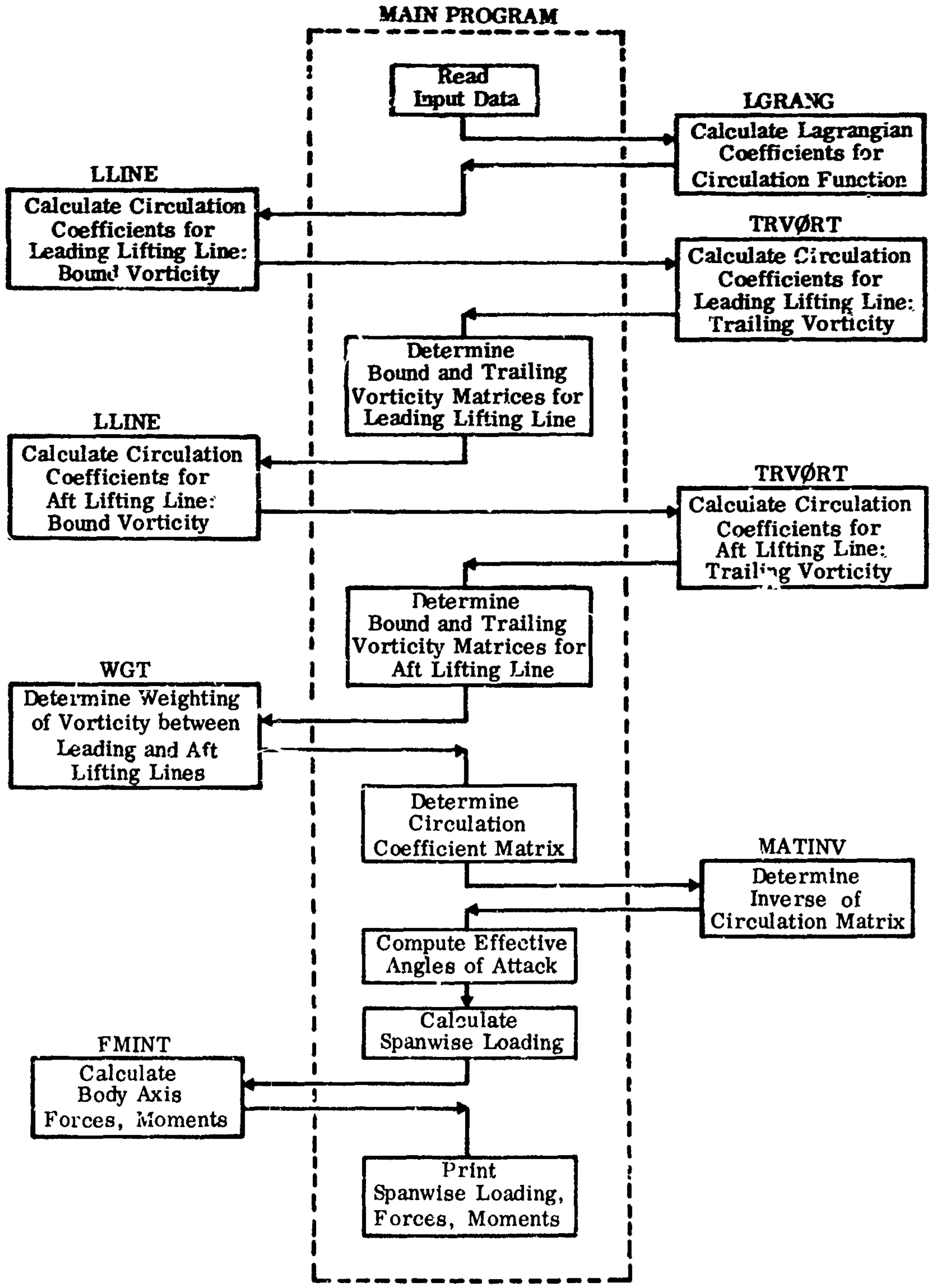

FIGURE 23. LOGICAL FLOW CHART FOR NONLINEAR WING AERODYNAMICS PROGRAM

Calling \ Called	WGT	GAUSS	LGRANG	LLINE	TRVØRT	MATINV	FMINT	FØRM1	FØRM2	FØRM3
MAIN	●		●	●	●	●	●			
LLINE		●						●		
TRVØRT		●							●	●

FIGURE 24. CALLING-CALLED MATRIX FOR NONLINEAR WING AERODYNAMICS PROGRAM

APPENDIX

COMPUTER PROGRAM LISTINGS

Preceding page blank

```
      PROGRAM JET3 (INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
     1TAPE7=PUNCH)
C
C     EVALUATION OF JET-INDUCED VELOCITY FIELD (MAXIMUM OF 3 JETS)
C     INITIAL JET EXHAUST DIRECTION MUST BE THE SAME FOR ALL THREE JETS
C     FOR 3-JET COMPUTATIONS, JET EXITS MUST ALL BE IN THE SAME XY PLANE
C
      DIMENSION COEFR(15,25),COEFI(15,25)
      DIMENSION STATN(25),RADIUS(25),SLP3D(25)
      DIMENSION X1(100),Z1(100),UJ1(100),D1(100),DXDZ1(100)
      DIMENSION X2(100),Z2(100),UJ2(100),D2(100),DXDZ2(100)
      DIMENSION X3(100),Z3(100),UJ3(100),D3(100),DXDZ3(100)
      DIMENSION X4(100),Z4(100),UJ4(100),D4(100),DXDZ4(100)
      DIMENSION X5(100),Z5(100),UJ5(100),D5(100),DXDZ5(100)
      DIMENSION XBAS1(100),YBAS1(100),ZBAS1(100)
      DIMENSION XBAS2(100),YBAS2(100),ZBAS2(100)
      DIMENSION XBAS3(100),YBAS3(100),ZBAS3(100)
      DIMENSION XBAS4(100),YBAS4(100),ZBAS4(100)
      DIMENSION XBAS5(100),YBAS5(100),ZBAS5(100)
      DIMENSION CF1(3,3),CF2(3,3),CF3(3,3),CF4(3,3),CF5(3,3)
      DIMENSION UUE1(100),UUE2(100),UUE3(100),UUE4(100),UUE5(100)
      DIMENSION PAR(10)
      DIMENSION SDXDZ1(100),SDXDZ2(100),SDXDZ3(100),SDXDZ4(100),
     1 SDXDZ5(100)
C
      COMMON/BLK1/STATN,RADIUS,SLP3D,COEFR,COEFI
      COMMON/BLK2/CF1,CF2,CF3,CF4,CF5,UUE1,UUE2,UUE3,UUE4,UUE5,PAR
      COMMON/BLK3/X1,Z1,UJ1,D1,DXDZ1,X2,Z2,UJ2,D2,DXDZ2
      COMMON/BLK4/X3,Z3,UJ3,D3,DXDZ3,X4,Z4,UJ4,D4,DXDZ4
      COMMON/BLK5/X5,Z5,UJ5,D5,DXDZ5
      COMMON/BLK6/XBAS1,YBAS1,ZBAS1,XBAS2,YBAS2,ZBAS2,XBAS3,YBAS3,ZBAS3
      COMMON/BLK7/XBAS4,YBAS4,ZBAS4,XBAS5,YBAS5,ZBAS5
      COMMON/BLK8/ALFQ,BETQ,GETQ,F1,F2,F3,F4,F5,VKONST
      COMMON/BLK9/MULT,IHOLD1,IHOLD2,IHOLD3,KOUNT1,KOUNT2
      COMMON/BLK10/IONE,ITWO,ITHR,IFOUR,IFIV,N1,N2,N3,N4,N5
      COMMON/BLK11/IFIX1,IFIX2,IFIX3
      COMMON/BLK12/XJ1,YJ1,ZJ1,DJET1,VELJ1,XJ2,YJ2,ZJ2,DJET2,VELJ2
      COMMON/BLK13/XJ3,YJ3,ZJ3,DJET3,VELJ3,XJ4,YJ4,ZJ4,DJET4,VELJ4
      COMMON/BLK14/XJ5,YJ5,ZJ5,DJET5,VELJ5
      COMMON/BLK15/G,G2,G3,G4,G5,STEPI,STEPI2,STEPI3,STEPI4,STEPI5
      COMMON/BLK16/V2X1,V2Y1,V2Z1,V2X2,V2Y2,V2Z2,V2X3,V2Y3,V2Z3
      COMMON/BLK17/V2X4,V2Y4,V2Z4
      COMMON/BLK18/DR3,DR4,DR5
      COMMON/BLK19/SDXDZ1,SDXDZ2,SDXDZ3,SDXDZ4,SDXDZ5
      COMMON/BLK20/DIARAT,DREF
C
      DIMENSION XO(600),YO(600),ZO(600),U(600),V(600),W(600)
      DIMENSION CP(600)
      DIMENSION PHID(3),PSID(3)
C
C     SET PARAMETERS
C
      E1 = .45
      E2 = .08
      E3 = 30.
      PI = 3.1416
      C1 = 2.24
```

```
C
C     READ IN JET DATA
C
      READ (5,501) MULT,IGEOM,IPUNCH
      READ (5,502) ALFA,BETA
      READ (5,503) N,G
  501 FORMAT (12I6)
  502 FORMAT (6F12.0)
  503 FORMAT (I6,F12.0)
      READ (5,502) XJ1,YJ1,ZJ1,PHID(1),PSID(1),DJET1,VELJ1
      IF (MULT-2) 4,2,2
    2 READ (5,502) XJ2,YJ2,ZJ2,PHID(2),PSID(2),DJET2,VELJ2
      IF (MULT-2) 4,4,3
    3 READ (5,502) XJ3,YJ3,ZJ3,PHID(3),PSID(3),DJET3,VELJ3
    4 CONTINUE
      READ (5,502) DIARAT
      WRITE (6,690)
      IF (MULT-2) 14,15,16
   14 WRITE (6,603)
  603 FORMAT (1H0,44X,32H*** SINGLE JET CONFIGURATION ***/)
      N1 = N+1
      GO TO 17
   15 WRITE (6,604)
  604 FORMAT (1H0,45X,29H*** TWO-JET CONFIGURATION ***/)
      GO TO 17
   16 WRITE (6,605)
  605 FORMAT (1H0,44X,31H*** THREE-JET CONFIGURATION ***/)
   17 CONTINUE
      WRITE (6,606) XJ1,YJ1,ZJ1,PHID(1),PSID(1),VELJ1
  606 FORMAT (1H0,22X,4HXJET,11X,4HYJET,11X,4HZJET,12X,3HPHI,12X,3HPSI,
     112X,5HU/UJO/15X,F15.4,1X,F14.4,1X,F14.4,1X,F14.4,1X,F14.4,1X,
     2F14.4)
      IF (MULT-2) 20,18,18
   18 WRITE (6,607) XJ2,YJ2,ZJ2,PHID(2),PSID(2),VELJ2
  607 FORMAT(15X,F15.4,1X,F14.4,1X,F14.4,1X,F14.4,1X,F14.4,1X,F14.4)
      IF (MULT-2) 20,20,19
   19 WRITE (6,607) XJ3,YJ3,ZJ3,PHID(3),PSID(3),VELJ3
   20 CONTINUE
      WRITE (6,608) ALFA,BETA
  608 FORMAT(1H0,/22X,19HANGLE OF ATTACK    =,1X,F7.2/22X,19HANGLE OF SID
     1ESLIP =,1X,F7.2)
      WRITE (6,609) N,G
  609 FORMAT(1H0,/22X,32HNUMBER OF STEPS IN INTEGRATION =,1X,I3,/22X,22H
     1INTEGRATION INTERVAL =,1X,F5.2,1X,18HJET EXIT DIAMETERS)
      CALL TRANS1 (MULT,ALFA,BETA,PSID)
      DO 8 I=1,MULT
      PHI = PHID(I)*.0174533
      PSI = PSID(I)*.0174533
      IF (I-2) 5,6,7
    5 CONTINUE
      CALL CFCAL (ALFQ,BETQ,GETQ,PHI,PSI,CF1)
      V2X1 = SIN(PHI)*COS(PSI)
      V2Y1 = COS(PHI)
      V2Z1 = SIN(PHI)*SIN(PSI)
      CALL ROTATE (V2X1,V2Y1,V2Z1,CF1,VXT,VYT,VZT,0)
      UJ1(1) = 1.
      D1(1) = 1.
      X1(1) = 0.
      Z1(1) = 0.
```

```
      DXDZ1(1) = VXT/VZT
      XBAS1(1) = XJ1
      YBAS1(1) = YJ1
      ZBAS1(1) = ZJ1
      STEPI = .2*G
      D = ATAN(VXT/VZT)
      IF (VXT) 901,902,902
  901 F1 = .3*COS(D)
      GO TO 903
  902 F1 = .3/COS(D)
  903 CONTINUE
      GO TO 8
    6 CONTINUE
      CALL CFCAL (ALFQ,BETQ,GETQ,PHI,PSI,CF2)
      V2X2 = SIN(PHI)*COS(PSI)
      V2Y2 = COS(PHI)
      V2Z2 = SIN(PHI)*SIN(PSI)
      CALL ROTATE (V2X2,V2Y2,V2Z2,CF2,VXT,VYT,VZT,0)
      UJ2(1) = 1.
      D2(1) = 1.
      X2(1) = 0.
      Z2(1) = 0.
      DXDZ2(1) = VXT/VZT
      XBAS2(1) = XJ2
      YBAS2(1) = YJ2
      ZBAS2(1) = ZJ2
      G2 = G*DJET1/DJET2
      STEPI2 = .2*G2
      D = ATAN(VXT/VZT)
      IF (VXT) 904,905,905
  904 F2 = .3*COS(D)
      GO TO 906
  905 F2 = .3/COS(D)
  906 CONTINUE
      GO TO 8
    7 CONTINUE
      CALL CFCAL (ALFQ,BETQ,GETQ,PHI,PSI,CF3)
      V2X3 = SIN(PHI)*COS(PSI)
      V2Y3 = COS(PHI)
      V2Z3 = SIN(PHI)*SIN(PSI)
      CALL ROTATE (V2X3,V2Y3,V2Z3,CF3,VXT,VYT,VZT,0)
      UJ3(1) = 1.
      D3(1) = 1.
      X3(1) = 0.
      Z3(1) = 0.
      DXDZ3(1) = VXT/VZT
      XBAS3(1) = XJ3
      YBAS3(1) = YJ3
      ZBAS3(1) = ZJ3
      G3 = G*DJET1/DJET3
      STEPI3 = .2*G3
      D = ATAN(VXT/VZT)
      IF (VXT) 907,908,908
  907 F3 = .3*COS(D)
      GO TO 909
  908 F3 = .3/COS(D)
  909 CONTINUE
    8 CONTINUE
C
```

```
C     TEST INITIAL JET EXHAUST DIRECTION (MUST BE THE SAME FOR ALL JETS)
C
      IF (MULT-2) 11,10,9
    9 CALL XPROD (ALFQ,BETQ,GETQ,V2X3,V2Y3,V2Z3,XT3,YT3,ZT3)
   10 CALL XPROD (ALFQ,BETQ,GETQ,V2X2,V2Y2,V2Z2,XT2,YT2,ZT2)
      CALL XPROD (ALFQ,BETQ,GETQ,V2X1,V2Y1,V2Z1,XT1,YT1,ZT1)
      IF (ABS(XT1-XT2)-.0001) 700,700,799
  700 IF (ABS(YT1-YT2)-.0001) 701,701,799
  701 IF (ABS(ZT1-ZT2)-.0001) 702,702,799
  702 IF (MULT-2) 11,11,12
   12 IF (ABS(XT1-XT3)-.0001) 703,703,799
  703 IF (ABS(YT1-YT3)-.0001) 704,704,799
  704 IF (ABS(ZT1-ZT3)-.0001) 11,11,799
  799 WRITE (6,620)
  620 FORMAT (1H0,71HJETS DO NOT EXHAUST IN PARALLEL PLANES, CONFIGURATI
     1ON CANNOT BE TREATED)
      STOP
   11 CONTINUE
      CALL VEL1 (MULT,ALFA,VK1,VK2)
      PAR(1) = E1
      PAR(2) = E2
      PAR(3) = E3
      PAR(7) = PI
      PAR(8) = Q1
      PAR(9) = 1.
C
C     TESTS FOR BLOCKAGE AND INTERSECTION,PART OF INTEGRATION LOOP
C
      N2 = 0
      N3 = 0
      N4 = 0
      N5 = 0
      IHOLD1 = 0
      IHOLD2 = 0
      IHOLD3 = 0
      KOUNT1 = 0
      KOUNT2 = 0
      TNEG = BETQ*V2Y1
      DREF = DJET1
      DO 50 I=1,N
      IONE = I
      ITWO = I
      ITHR = I
      IFOUR = I
      IFIV = I
      VKCNST = VK1
      IF (MULT-2) 21,22,23
   22 IF (IHOLD1-1) 25,25,21
   23 IF (IHOLD3-1) 25,25,21
   25 CALL BITEST (I,TNEG,VK1,VK2)
   21 CONTINUE
C
C     INTEGRATION OF THE EQUATIONS OF MOTION FOR THE JET PATH
C
      CALL INTEG (I,TNEG)
   50 CONTINUE
C
C     READING IN CONTROL POINTS WHERE VELOCITIES WILL BE COMPUTED
C
```

```
      IF (IGEOM-2) 61,62,63
   61 READ (5,501) NTHT,NSMAX,NCOEF,IRECT
      CALL TRWING (NTHT,NSMAX,NCOEF,IRECT,XO,YO,ZO,NK)
      NSYM = 1
      GO TO 65
   62 READ (5,501) NTHT,NSMAX,NCOEF,NSYM
      CALL TRBODY (NTHT,NSMAX,NCOEF,NSYM,XO,YO,ZO,NK)
      GO TO 65
   63 READ (5,501) NSMAX,NC
      NK = NSMAX*NC
      READ (5,502) (XO(I),YO(I),ZO(I), I=1,NK)
   65 CONTINUE
      CALL TRANS2 (YO,ZO,NK)
C
C     EVALUATE INDUCED VELOCITIES AT EACH POINT
C
      DO 80 J=1,NK
      U(J) = 0.
      V(J) = 0.
      W(J) = 0.
      PAR(6) = VELJ1
      PAR(5) = F1
      PAR(9) = 1.
      CALL VELOC (1,N1,Z1,X1,DXDZ1,UJ1,D1,UUE1,XJ1,YJ1,ZJ1,DJET1,CF1,
     1 PAR,XO(J),YO(J),ZO(J),UIND,VIND,WIND,SDXDZ1)
      U(J) = U(J)+UIND
      V(J) = V(J)+VIND
      W(J) = W(J)+WIND
      IF (MULT-2) 80,51,51
   51 PAR(6) = VELJ2
      PAR(5) = F2
      PAR(9) = 1.
      CALL VELOC (1,N2,Z2,X2,DXDZ2,UJ2,D2,UUE2,XJ2,YJ2,ZJ2,DJET2,CF2,
     1 PAR,XO(J),YO(J),ZO(J),UIND,VIND,WIND,SDXDZ2)
      U(J) = U(J)+UIND
      V(J) = V(J)+VIND
      W(J) = W(J)+WIND
      IF (MULT-2) 80,52,53
   52 IF (IHOLD1-1) 80,80,54
   54 N3 = ITHR+1
      PAR(9) = DR3
      GO TO 55
   53 PAR(9) = 1.
   55 PAR(6) = VELJ3
      PAR(5) = F3
      CALL VELOC (1,N3,Z3,X3,DXDZ3,UJ3,D3,UUE3,XJ3,YJ3,ZJ3,DJET3,CF3,
     1 PAR,XO(J),YO(J),ZO(J),UIND,VIND,WIND,SDXDZ3)
      U(J) = U(J)+UIND
      V(J) = V(J)+VIND
      W(J) = W(J)+WIND
      IF (MULT-2)  80,80,56
   56 IF (IHOLD1-1) 57,57,58
   57 IF (IHOLD2-1) 80,80,58
   58 PAR(6) = VELJ4
      PAR(5) = F4
      PAR(9) = DR4
      CALL VELOC (1,N4,Z4,X4,DXDZ4,UJ4,D4,JUE4,XJ4,YJ4,ZJ4,DJET4,CF4,
     1 PAR,XO(J),YO(J),ZO(J),UIND,VIND,WIND,SDXDZ4)
      U(J) = U(J)+UIND
```

```
      V(J) = V(J)+VIND
      W(J) = W(J)+WIND
      IF (IHOLD3-1) 80,80,59
   59 N5 = IFIV+1
      PAR(6) = VELJ5
      PAR(5) = F5
      PAR(9) = DR5
      CALL VELOC (1,N5,Z5,X5,DXDZ5,UJ5,D5,UUE5,XJ5,YJ5,ZJ5,DJET5,CF5,
     1 PAR,XO(J),YO(J),ZO(J),UIND,VIND,WIND,SDXDZ5)
      U(J) = U(J)+UIND
      V(J) = V(J)+VIND
      W(J) = W(J)+WIND
   80 CONTINUE
C
C     COMPUTE FLAT PLATE PRESSURE COEFFICIENT
C
      IF (IGEOM-3) 90,90,81
   81 DO 85 J=1,NK
      CPT = 4.*(U(J)*(ALFQ+U(J))+W(J)*(GETQ+W(J)))
   85 CP(J) = 1.-(ALFQ*ALFQ +GETQ*GETQ +CPT)
   90 CONTINUE
      CALL TRANS3 (YO,ZO,V,W,NK)
C
C     PRINT OUT COMPUTED RESULTS
C
      WRITE (6,690)
  690 FORMAT (1H1)
      CALL PRTOUT (IGEOM,XO,YO,ZO,U,V,W,CP,NK,NTHT)
C
C     PUNCH OUT DATA FOR TRANSFORMATION METHOD OR LIFTING SURFACE PROG.
C
      IF (IGEOM-2) 96,96,97
   96 IF (IPUNCH)  95,99,95
   95 CALL ADAPT(U,V,W,NTHT,NSMAX,NCOEF,IGEOM)
      GO TO 99
   97 IF (IPUNCH)  98,99,98
   98 DO 101 I=1,NK
  101 W(I) = -W(I)
      J1 = 1
      DO 102 I=1,NSMAX
      J2 = J1+NC-1
      WRITE (7,710) (W(J), J=J1,J2)
  102 J1 = J2+1
  710 FORMAT (5E14.7)
   99 CONTINUE
      STOP
      END


      SUBROUTINE BITEST (I,TNEG,VK1,VK2)
C
C     TESTS FOR BLOCKAGE AND INTERSECTION,CALLED AS PART OF INTEGRATION
C     LOOP
C
      DIMENSION COEFR(15,25),COEFI(15,25)
      DIMENSION STATN(25),RADIUS(25),SLP3D(25)
      DIMENSION X1(100),Z1(100),UJ1(100),D1(100),DXDZ1(100)
      DIMENSION X2(100),Z2(100),UJ2(100),D2(100),DXDZ2(100)
      DIMENSION X3(100),Z3(100),UJ3(100),D3(100),DXDZ3(100)
```

```
      DIMENSION X4(100),Z4(100),UJ4(100),D4(100),DXDZ4(100)
      DIMENSION X5(100),Z5(100),UJ5(100),D5(100),DXDZ5(100)
      DIMENSION XBAS1(100),YBAS1(100),ZBAS1(100)
      DIMENSION XBAS2(100),YBAS2(100),ZBAS2(100)
      DIMENSION XBAS3(100),YBAS3(100),ZBAS3(100)
      DIMENSION XBAS4(100),YBAS4(100),ZBAS4(100)
      DIMENSION XBAS5(100),YBAS5(100),ZBAS5(100)
      DIMENSION CF1(3,3),CF2(3,3),CF3(3,3),CF4(3,3),CF5(3,3)
      DIMENSION UUE1(100),UUE2(100),UUE3(100),UUE4(100),UUE5(100)
      DIMENSION PAR(10)
C
      COMMON/BLK1/STATN,RADIUS,SLP3D,COEFR,COEFI
      COMMON/BLK2/CF1,CF2,CF3,CF4,CF5,UUE1,UUE2,UUE3,UUE4,UUE5,PAR
      COMMON/BLK3/X1,Z1,UJ1,D1,DXDZ1,X2,Z2,UJ2,D2,DXDZ2
      COMMON/BLK4/X3,Z3,UJ3,D3,DXDZ3,X4,Z4,UJ4,D4,DXDZ4
      COMMON/BLK5/X5,Z5,UJ5,D5,DXDZ5
      COMMON/BLK6/XBAS1,YBAS1,ZBAS1,XBAS2,YBAS2,ZBAS2,XBAS3,YBAS3,ZBAS3
      COMMON/BLK7/XBAS4,YBAS4,ZBAS4,XBAS5,YBAS5,ZBAS5
      COMMON/BLK8/ALFQ,BETQ,GETQ,F1,F2,F3,F4,F5,VKONST
      COMMON/BLK9/MULT,IHOLD1,IHOLD2,IHOLD3,KOUNT1,KOUNT2
      COMMON/BLK10/IONE,ITWO,ITHR,IFOUR,IFIV,N1,N2,N3,N4,N5
      COMMON/BLK11/IFIX1,IFIX2,IFIX3
      COMMON/BLK12/XJ1,YJ1,ZJ1,DJET1,VELJ1,XJ2,YJ2,ZJ2,DJET2,VELJ2
      COMMON/BLK13/XJ3,YJ3,ZJ3,DJET3,VELJ3,XJ4,YJ4,ZJ4,DJET4,VELJ4
      COMMON/BLK14/XJ5,YJ5,ZJ5,DJET5,VELJ5
      COMMON/BLK15/G,G2,G3,G4,G5,STEPI,STEPI2,STEPI3,STEPI4,STEPI5
      COMMON/BLK16/V2X1,V2Y1,V2Z1,V2X2,V2Y2,V2Z2,V2X3,V2Y3,V2Z3
      COMMON/BLK17/V2X4,V2Y4,V2Z4
      COMMON/BLK18/DR3,DR4,DR5
C
      DE = .0001*DJET1
      IF (MULT-2) 21,200,300
  200 IF (IHOLD1-1) 201,202,21
  201 IF (TNEG) 203,203,204
  203 CALL XPROD (V2X1,V2Y1,V2Z1,ALFQ,BETQ,GETQ,XT1,YT1,ZT1)
      CALL XPROD (XT1,YT1,ZT1,ALFQ,BETQ,GETQ,CFNX,CFNY,CFNZ)
      CALL PLANE (CFNX,CFNY,CFNZ,XBAS1(I),YBAS1(I),ZBAS1(I),V2X2,V2Y2,
     1 V2Z2,XJ2,YJ2,ZJ2,XINT,YINT,ZINT)
      IF (YINT-YJ2-DE) 205,205,22
  204 UUE2(I) = 1.
      CALL XPROD (V2X2,V2Y2,V2Z2,ALFQ,BETQ,GETQ,XT2,YT2,ZT2)
      CALL XPROD (XT2,YT2,ZT2,ALFQ,BETQ,GETQ,CFNX,CFNY,CFNZ)
      CALL PLANE (CFNX,CFNY,CFNZ,XBAS2(I),YBAS2(I),ZBAS2(I),V2X1,V2Y1,
     1 V2Z1,XJ1,YJ1,ZJ1,XINT,YINT,ZINT)
      IF (YINT-YJ1-DE) 205,205,22
  205 IHOLD1 = 1
  202 IF (TNEG) 206,206,207
  206 ITWO = I-KOUNT1
      GO TO 208
  207 IONE = I-KOUNT1
  208 IT1 = IONE
      IT2 = ITWO
      N1 = IT1+1
      N2 = IT2+1
      CALL COMP (V2X1,V2Y1,V2Z1,V2X2,V2Y2,V2Z2,XBAS1(IT1),YBAS1(IT1),
     1 ZBAS1(IT1),XBAS2(IT2),YBAS2(IT2),ZBAS2(IT2),Z1(IT1),Z2(IT2),
     2 D1(IT1),DJET1,D2(IT2),DJET2,VELJ1,VELJ2,DXDZ1(IT1),UUE2(IT2),
     3 A1,A2,DR3,F1,INT)
      IF (INT) 21,21,209
```

```
209  IHCLD1 = 2
     N1 = IT1
     N2 = IT2
     PAR(9) = DR3
     IFIX1 = I
     CALL BALANC (XBAS1(IT1),YBAS1(IT1),ZBAS1(IT1),XBAS2(IT2),
    1 YBAS2(IT2),ZBAS2(IT2),UJ1(IT1),UJ2(IT2),VELJ1,VELJ2,A1,A2,V2X1,
    2 V2Y1,V2Z1,V2X2,V2Y2,V2Z2,DR3,XJ3,YJ3,ZJ3,DJET3,V2X3,V2Y3,V2Z3,
    3 VELJ3)
     PHI = ACOS(V2Y3)
     PSI = ATAN(V2Z3/V2X3)
     CALL CFCAL (ALFQ,BETQ,GETQ,PHI,PSI,CF3)
     CALL ROTATE (V2X3,V2Y3,V2Z3,CF3,VXT,VYT,VZT,0)
     UJ3(1) = 1.
     D3(1) = 1.
     X3(1) = 0.
     Z3(1) = 0.
     DXDZ3(1) = VXT/VZT
     XBAS3(1) = XJ3
     YBAS3(1) = YJ3
     ZBAS3(1) = ZJ3
     PAR(6) = VELJ3
     D = ATAN(VXT/VZT)
     IF  (VXT) 901,902,902
901  F3 = .3*COS(D)
     GO TO 903
902  F3 = .3/COS(D)
903  PAR(5) = F3
     G3 = G*DJET1/DJET3
     STEPI3 = .2*G3
     GO TO 21
300  IF (IHOLD3-1) 301,301,21
301  IF (TNEG) 302,302,303
303  WRITE (6,680)
680  FORMAT (1HO,70HNEGATIVE ANGLE OF ATTACK FOR THREE-JET CONFIGURATIO
    1N CANNOT BE TREATED)
     STOP
302  IF (IHOLD1-1) 320,321,322
320  CALL XPROD (V2X1,V2Y1,V2Z1,ALFQ,BETQ,GETQ,XT1,YT1,ZT1)
     CALL XPROD (XT1,YT1,ZT1,ALFQ,BETQ,GETQ,CFNX,CFNY,CFNZ)
     CALL PLANE (CFNX,CFNY,CFNZ,XBAS1(I),YBAS1(I),ZBAS1(I),V2X2,V2Y2,
    1 V2Z2,XJ2,YJ2,ZJ2,XINT,YINT,ZINT)
     IF (YINT-YJ2-DE) 323,323,22
323  IHOLD1 = 1
321  IF (IHOLD2-1) 324,324,325
324  ITWC   I-KOUNT1
     IT1 = IONE
     IT2 = ITWC
     N1 = IT1+1
     N2 = IT2+1
     VKONST = VK1
     CALL CCFP (V2X1,V2Y1,V2Z1,V2X2,V2Y2,V2Z2,XBAS1(IT1),YBAS1(IT1),
    1 ZBAS1(IT1),XBAS2(IT2),YBAS2(IT2),ZBAS2(IT2),Z1(IT1),Z2(IT2),
    2 D1(IT1),DJET1,D2(IT2),DJET2-VELJ1,VELJ2,DXDZ1(IT1),UUE2(IT2).
    3 A1,A2,DR4 F1,INT)
     IF (INT) 330,330,331
331  IHCLD1 = 2
     N1 = IT1
     N2 = IT2
```

```
      IFIX1 = I
      VKCNST = VK2
      CALL BALANC (XBAS1(IT1),YBAS1(IT1),ZBAS1(IT1),XBAS2(IT2),
     1 YBAS2(IT2),ZBAS2(IT2),UJ1(IT1),UJ2(IT2),VELJ1,VELJ2,A1,A2,V2X1,
     2 V2Y1,V2Z1,V2X2,V2Y2,V2Z2,DR4,XJ4,YJ4,ZJ4,CJET4,V2X4,V2Y4,V2Z4,
     3 VELJ4)
340   PHI = ACOS(V2Y4)
      PSI = ATAN(V2Z4/V2X4)
      CALL CFCAL (ALFQ,BETQ,GETQ,PHI,PSI,CF4)
      CALL ROTATE (V2X4,V2Y4,V2Z4,CF4,VXT,VYT,VZT,0)
      UJ4(1) = 1.
      D4(1) = 1.
      X4(1) = 0.
      Z4(1) = 0.
      DXDZ4(1) = VXT/VZT
      XBAS4(1) = XJ4
      YBAS4(1) = YJ4
      ZBAS4(1) = ZJ4
      D = ATAN(VXT/VZT)
      IF (VXT) 904,905,905
904   F4 = .3*COS(D)
      GO TO 906
905   F4 = .3/COS(D)
906   CONTINUE
      G4 = G*DJET1/DJET4
      STEPI4 = .2*G4
      IF (IHOLD2-IHOLD1) 322,322,325
330   IF (IHOLD2-1) 332,333,325
332   CALL XPROD (V2X2,V2Y2,V2Z2,ALFQ,BETQ,GETQ,XT2,YT2,ZT2)
      CALL XPROD (XT2,YT2,ZT2,ALFQ,BETQ,GETQ,CFNX,CFNY,CFNZ)
      CALL PLANE (CFNX,CFNY,CFNZ,XBAS2(IT2),YBAS2(IT2),ZBAS2(IT2),V2X3,
     1 V2Y3,V2Z3,XJ3,YJ3,ZJ3,XINT,YINT,ZINT)
      IF (YINT-YJ3-DE) 334,334,23
334   IHOLD2 = 1
333   ITHR = I-KOUNT2
      IT3 = ITHR
      N3 = IT3+1
      VKONST = VK2
      CALL COMP (V2X2,V2Y2,V2Z2,V2X3,V2Y3,V2Z3,XBAS2(IT2),YBAS2(IT2),
     1 ZBAS2(IT2),XBAS3(IT3),YBAS3(IT3),ZBAS3(IT3),Z2(IT2),Z3(IT3),
     2 D2(IT2),DJET2,D3(IT3),DJET3,VELJ2,VELJ3,DXDZ2(IT2),UUE3(IT3),
     3 A2,A3,DR4,F2,INT)
      IF (INT) 21,21,335
335   IHOLD2 = 2
      N3 = IT3
      N2 = IT2
      IFIX2 = I
      VKCNST = VK1
      CALL BALANC (XBAS2(IT2),YBAS2(IT2),ZBAS2(IT2),XBAS3(IT3),
     1 YBAS3(IT3),ZBAS3(IT3),UJ2(IT2),UJ3(IT3),VELJ2,VELJ3,A2,A3,V2X2,
     2 V2Y2,V2Z2,V2X3,V2Y3,V2Z3,DR4,XJ4,YJ4,ZJ4,CJET4,V2X4,V2Y4,V2Z4,
     3 VELJ4)
      GO TO 340
322   IFCLR = I-IFIX1+1
      ITHR = I-KOUNT2
      IT4 = IFCLR
      IT3 = ITHR
      N4 = IT4+1
      N3 = IT3+1
```

```
      UUE4(IT4) = 1.
      CALL COMP (V2X4,V2Y4,V2Z4,V2X3,V2Y3,V2Z3,XBAS4(IT4),YBAS4(IT4),
     1 ZBAS4(IT4),XBAS3(IT3),YBAS3(IT3),ZBAS3(IT3),Z4(IT4),Z3(IT3),
     2 D4(IT4),DJET4,D3(IT3),DJET3,VELJ4,VELJ3,DXDZ4(IT4),UUE3(IT3),
     3 A4,A3,DR5,F4,INT)
      IF (INT) 21,21,341
341   IHCLD3 = 2
      N3 = IT3
      N4 = IT4
      IFIX3 = I
      CALL BALANC (XBAS4(IT4),YBAS4(IT4),ZBAS4(IT4),XBAS3(IT3),
     1 YBAS3(IT3),ZBAS3(IT3),UJ4(IT4),UJ3(IT3),VELJ4,VELJ3,A4,A3,V2X4,
     2 V2Y4,V2Z4,V2X3,V2Y3,V2Z3,DR5,XJ5,YJ5,ZJ5,DJET5,V2X5,V2Y5,V2Z5,
     3 VELJ5)
350   PHI = ACOS(V2Y5)
      PSI = ATAN(V2Z5/V2X5)
      CALL CFCAL (ALFQ,BETQ,GETQ,PHI,PSI,CF5)
      CALL ROTATE (V2X5,V2Y5,V2Z5,CF5,VXT,VYT,VZT,0)
      UJ5(1) = 1.
      D5(1) = 1.
      X5(1) = 0.
      Z5(1) = 0.
      DXDZ5(1) = VXT/VZT
      XBAS5(1) = XJ5
      YBAS5(1) = YJ5
      ZBAS5(1) = ZJ5
      D = ATAN(VXT/VZT)
      IF (VXT) 907,908,908
907   F5 = .3*COS(D)
      GO TO 909
908   F5 = .3/COS(D)
909   PAR(5) = F5
      G5 = G*DJET1/DJET5
      STEPI5 = .2*G5
      PAR(9) = DR5
      PAR(6) = VELJ5
      GO TO 21
325   IFCUR = I-IFIX2+1
      IT1 = IONE
      IT4 = IFOUR
      N1 = IT1+1
      N4 = IT4+1
      CALL COMP (V2X1,V2Y1,V2Z1,V2X4,V2Y4,V2Z4,XBAS1(IT1),YBAS1(IT1),
     1 ZBAS1(IT1),XBAS4(IT4),YBAS4(IT4),ZBAS4(IT4),Z1(IT1),Z4(IT4),
     2 D1(IT1),DJET1,D4(IT4),DJET4,VELJ1,VELJ4,DXDZ1(IT1),UUE4(IT4),
     3 A1,A4,DR5,F1,INT)
      IF (INT) 21,21,342
342   IHCLD3 = 2
      N1 = IT1
      N4 = IT4
      IFIX3 = I
      CALL BALANC (XBAS1(IT1),YBAS1(IT1),ZBAS1(IT1),XBAS4(IT4),
     1 YBAS4(IT4),ZBAS4(IT4),UJ1(IT1),UJ4(IT4),VELJ1,VELJ4,A1,A4,V2X1,
     2 V2Y1,V2Z1,V2X4,V2Y4,V2Z4,DR5,XJ5,YJ5,ZJ5,DJET5,V2X5,V2Y5,V2Z5,
     3 VELJ5)
      GO TO 350
 22   KOUNT1 = KOUNT1+1
 23   KOUNT2 = KOUNT2+1
 21   CONTINUE
```

```
      RETURN
      END

      SUBROUTINE INTEG (I,TNEG)
C
C     INTEGRATION OF THE EQUATIONS OF MOTION FOR THE JET PATH
C
      EXTERNAL DERIV
C
      DIMENSION COEFR(15,25),COEFI(15,25)
      DIMENSION STATN(25),RADIUS(25),SLP3D(25)
      DIMENSION X1(100),Z1(100),UJ1(100),D1(100),DXDZ1(100)
      DIMENSION X2(100),Z2(100),UJ2(100),D2(100),DXDZ2(100)
      DIMENSION X3(100),Z3(100),UJ3(100),D3(100),DXDZ3(100)
      DIMENSION X4(100),Z4(100),UJ4(100),D4(100),DXDZ4(100)
      DIMENSION X5(100),Z5(100),UJ5(100),D5(100),DXDZ5(100)
      DIMENSION XBAS1(100),YBAS1(100),ZBAS1(100)
      DIMENSION XBAS2(100),YBAS2(100),ZBAS2(100)
      DIMENSION XBAS3(100),YBAS3(100),ZBAS3(100)
      DIMENSION XBAS4(100),YBAS4(100),ZBAS4(100)
      DIMENSION XBAS5(100),YBAS5(100),ZBAS5(100)
      DIMENSION CF1(3,3),CF2(3,3),CF3(3,3),CF4(3,3),CF5(3,3)
      DIMENSION UUE1(100),UUE2(100),UUE3(100),UUE4(100),UUE5(100)
      DIMENSION PAR(10)
      DIMENSION SDXDZ1(100),SDXDZ2(100),SDXDZ3(100),SDXDZ4(100),
     1 SDXDZ5(100)
C
      COMMON/BLK1/STATN,RADIUS,SLP3D,COEFR,COEFI
      COMMON/BLK2/CF1,CF2,CF3,CF4,CF5,UUE1,UUE2,UUE3,UUE4,UUE5,PAR
      COMMON/BLK3/X1,Z1,UJ1,D1,DXDZ1,X2,Z2,UJ2,D2,DXDZ2
      COMMON/BLK4/X3,Z3,UJ3,D3,DXDZ3,X4,Z4,UJ4,D4,DXDZ4
      COMMON/BLK5/X5,Z5,UJ5,D5,DXDZ5
      COMMON/BLK6/XBAS1,YBAS1,ZBAS1,XBAS2,YBAS2,ZBAS2,XBAS3,YBAS3,ZBAS3
      COMMON/BLK7/XBAS4,YBAS4,ZBAS4,XBAS5,YBAS5,ZBAS5
      COMMON/BLK8/ALFQ,BETQ,GETQ,F1,F2,F3,F4,F5,VKONST
      COMMON/BLK9/MULT,IHOLD1,IHOLD2,IHOLD3,KOUNT1,KOUNT2
      COMMON/BLK10/IONE,ITWO,ITHR,IFOUR,IFIV,N1,N2,N3,N4,N5
      COMMON/BLK11/IFIX1,IFIX2,IFIX3
      COMMON/BLK12/XJ1,YJ1,ZJ1,DJET1,VELJ1,XJ2,YJ2,ZJ2,DJET2,VELJ2
      COMMON/BLK13/XJ3,YJ3,ZJ3,DJET3,VELJ3,XJ4,YJ4,ZJ4,DJET4,VELJ4
      COMMON/BLK14/XJ5,YJ5,ZJ5,DJET5,VELJ5
      COMMON/BLK15/G,G2,G3,G4,G5,STEPI,STEPI2,STEPI3,STEPI4,STEPI5
      COMMON/BLK16/V2X1,V2Y1,V2Z1,V2X2,V2Y2,V2Z2,V2X3,V2Y3,V2Z3
      COMMON/BLK17/V2X4,V2Y4,V2Z4
      COMMON/BLK18/DR3,DR4,DR5
      COMMON/BLK19/SDXDZ1,SDXDZ2,SDXDZ3,SDXDZ4,SDXDZ5
C
      DIMENSION FIN(4),FOUT(4)
C
      IF (MULT-2) 53,51,52
   51 IF (IHOLD1-2) 53,30,30
   52 IF (IHOLD3-2) 53,40,40
   53 IF (MULT-2) 24,25,26
   25 IF (TNEG) 24,24,27
   27 IF (IHOLD1) 28,28,24
   26 IF (IHOLD1-1) 24,24,31
   24 PAR(6) = VELJ1
      PAR(5) = F1
```

```
      PAR(9) = 1.
      UUE1(IONE) = 1.
      Z1(IONE+1) = Z1(IONE)+G
      FIN(1) = UJ1(IONE)
      FIN(2) = D1(IONE)
      FIN(3) = X1(IONE)
      FIN(4) = DXDZ1(IONE)
      CALL ADAMS(4,Z1(IONE),Z1(IONE+1),STEPI,G,999,1.0E-04,1.0E-05,
     1 0,FIN,FOUT,PAR,DERIV)
      UJ1(IONE+1) = FOUT(1)
      D1(IONE+1) = FOUT(2)
      X1(IONE+1) = FOUT(3)
      DXDZ1(IONE+1) = FOUT(4)
      SDXDZ1(IONE+1) = PAR(10)
      CALL OUTPT(X1(IONE+1),Z1(IONE+1),DXDZ1(IONE+1),CF1,DJET1,XJ1,YJ1,
     1 ZJ1,XBAS1(IONE+1),YBAS1(IONE+1),ZBAS1(IONE+1),V2X1,V2Y1,V2Z1)
      IF (MULT-2) 50,41,42
   41 IF (IHOLD1) 50,50,28
   42 IF (IHOLD2-1) 50,28,46
   28 PAR(6) = VELJ2*UUE2(ITWO)
      PAR(5) = F2
      PAR(9) = 1.
      Z2(ITWO+1) = Z2(ITWO)+G2
      FIN(1) = UJ2(ITWO)
      FIN(2) = D2(ITWO)
      FIN(3) = X2(ITWO)
      FIN(4) = DXDZ2(ITWO)
      CALL ADAMS(4,Z2(ITWO),Z2(ITWO+1),STEPI2,G2,999,1.0E-04,
     1 1.0E-05,0,FIN,FOUT,PAR,DERIV)
      UJ2(ITWO+1) = FOUT(1)
      D2(ITWO+1) = FOUT(2)
      X2(ITWO+1) = FOUT(3)
      DXDZ2(ITWO+1) = FOUT(4)
      SDXDZ2(ITWO+1) = PAR(10)
      CALL OUTPT (X2(ITWO+1),Z2(ITWO+1),DXDZ2(ITWO+1),CF2,DJET2,XJ2,YJ2,
     1 ZJ2,XBAS2(ITWO+1),YBAS2(ITWO+1),ZBAS2(ITWO+1),V2X2,V2Y2,V2Z2)
      IF (MULT-2) 50,50,31
   31 IF (IHOLD2-1) 50,32,46
   32 PAR(6) = VELJ3*UUE3(ITHR)
      PAR(5) = F3
      PAR(9) = 1.
      GO TO 35
   30 ITHR = I-IFIX1+1
      UUE3(ITHR) = 1.
   35 Z3(ITHR+1) = Z3(ITHR)+G3
      FIN(1) = UJ3(ITHR)
      FIN(2) = D3(ITHR)
      FIN(3) = X3(ITHR)
      FIN(4) = DXDZ3(ITHR)
      CALL ADAMS(4,Z3(ITHR),Z3(ITHR+1),STEPI3,G3,999,1.0E-04,
     1 1.0E-05,0,FIN,FOUT,PAR,DERIV)
      UJ3(ITHR+1) = FOUT(1)
      D3(ITHR+1) = FOUT(2)
      X3(ITHR+1) = FOUT(3)
      DXDZ3(ITHR+1) = FOUT(4)
      SDXDZ3(ITHR+1) = PAR(10)
      CALL OUTPT (X3(ITHR+1),Z3(ITHR+1),DXDZ3(ITHR+1),CF3,DJET3,XJ3,YJ3,
     1 ZJ3,XBAS3(ITHR+1),YBAS3(ITHR+1),ZBAS3(ITHR+1),V2X3,V2Y3,V2Z3)
      IF (MULT-2) 50,50,47
```

```
   47 IF (IHOLD1-1) 50,50,46
   46 PAR(6) = VELJ4*UUE4(IFOUR)
      PAR(5) = F4
      PAR(9) = DR4
      Z4(IFOUR+1) = Z4(IFOUR)+G4
      FIN(1) = UJ4(IFOUR)
      FIN(2) = D4(IFOUR)
      FIN(3) = X4(IFOUR)
      FIN(4) = DXDZ4(IFOUR)
      CALL ADAMS(4,Z4(IFOUR),Z4(IFOUR+1),STEPI4,G4,999,1.0E-04,
     1 1.0E-05,0,FIN,FOUT,PAR,DERIV)
      UJ4(IFOUR+1) = FOUT(1)
      D4(IFOUR+1) = FOUT(2)
      X4(IFOUR+1) = FOUT(3)
      DXDZ4(IFOUR+1) = FOUT(4)
      SDXDZ4(IFOUR+1) = PAR(10)
      CALL OUTPT (X4(IFOUR+1),Z4(IFOUR+1),DXDZ4(IFOUR+1),CF4,DJET4,XJ4,
     1 YJ4,ZJ4,XBAS4(IFOUR+1),YBAS4(IFOUR+1),ZBAS4(IFOUR+1),V2X4,V2Y4,
     2 V2Z4)
      GO TO 50
   40 IFIV = I-IFIX3+1
      UUE5(IFIV) = 1.
      Z5(IFIV+1) = Z5(IFIV)+G5
      FIN(1) = UJ5(IFIV)
      FIN(2) = D5(IFIV)
      FIN(3) = X5(IFIV)
      FIN(4) = DXDZ5(IFIV)
      CALL ADAMS(4,Z5(IFIV),Z5(IFIV+1),STEPI5,G5,999,1.0E-04,
     1 1.0E-05,0,FIN,FOUT,PAR,DERIV)
      UJ5(IFIV+1) = FOUT(1)
      D5(IFIV+1) =  FOUT(2)
      X5(IFIV+1) = FOUT(3)
      DXDZ5(IFIV+1) = FOUT(4)
      SDXDZ5(IFIV+1) = PAR(10)
      CALL OUTPT (X5(IFIV+1),Z5(IFIV+1),DXDZ5(IFIV+1),CF5,DJET5,XJ5,YJ5,
     1 ZJ5,XBAS5(IFIV+1),YBAS5(IFIV+1),ZBAS5(IFIV+1),DUMMY,DUMMY,DUMMY)
   50 CONTINUE
      RETURN
      END


      SUBROUTINE COMP(VX1,VY1,VZ1,VX2,VY2,VZ2,X1,Y1,Z1,X2,Y2,Z2,Z1L,Z2L,
     1                D1,DJ1,D2,DJ2,V1,V2,SL1,UUEFF,A1,A2,DRAT,F,IND)
C
C     COMPUTES U/UEFFECTIVE AND TESTS FOR INTERSECTION OF CENTERLINES
C
      COMMON/BLKC/ALFQ,BETQ,GETQ,F1,F2,F3,F4,F5,VKONST
      COMMON/BLK20/DIARAT,DREF
C
      IND = 0
      PI = 3.1416
      CALL XPROD (VX1,VY1,VZ1,ALFQ,BETQ,GETQ,CFNX,CFNY,CFNZ)
      CALL XPROD (VX2,VY2,VZ2,ALFQ,BETQ,GETQ,XT2,YT2,ZT2)
      CALL PLANE (CFNX,CFNY,CFNZ,X1,Y1,Z1,XT2,YT2,ZT2,X2,Y2,Z2,XI,YI,ZI)
      DIST = SQRT((XI-X2)**2+(YI-Y2)**2+(ZI-Z2)**2)
C
C     COMPUTE U/UEFFECTIVE
C
      R = D1*DJ1*.5-DIST
```

```
      FACT = (1.0+R/(D2*DJ2*.5))*.5
      IF (FACT-1.) 10,10,11
   11 UUEFF = VKONST
      GO TO 15
   10 IF (FACT) 13,13,12
   13 UUEFF = 1.
      GO TO 15
   12 UEFU = 1.+(1./VKONST-1.)*FACT
      UUEFF = 1./UEFU
   15 CONTINUE
C
C     TEST FOR INTERSECTION OF CENTERLINES
C
      COST = 1./SQRT(1.+SL1*SL1)
      SUMD = DJ1*D1*.5
      IF (DIST-SUMD) 22,99,99
   22 DISTN = SQRT((X1-XI)**2+(Y1-YI)**2+(Z1-ZI)**2)
      ZOVM = Z1L/V1
      IF (ZOVM-F) 24,24,25
   24 FACT1 = 1.-.75*ZOVM/F
      GO TO 26
   25 FACT1 = .25
   26 ZOVM = Z2L/(V2*UUEFF)
      IF (ZOVM-F) 27,27,28
   27 FACT2 = 1.-.75*ZOVM/F
      GO TO 29
   28 FACT2 = .25
   29 SUMD = DJ1*D1*FACT1*COST*.5
      IF (DISTN-SUMD) 30,30,40
   30 IND = 1
      GO TO 45
   40 IF (X2-X1) 30,30,99
   45 A1 =  PI*FACT1*D1*D1*DJ1*DJ1*.25
      A2 =  PI*FACT2*D2*D2*DJ2*DJ2*.25
      DRAT = DIARAT
   99 CONTINUE
      RETURN
      END


      SUBROUTINE BALANC (X1,Y1,Z1,X2,Y2,Z2,UJ1,UJ2,V1,V2,A1,A2,VX1,VY1,
     1                   VZ1,VX2,VY2,VZ2,FACT1,X3,Y3,Z3,DJ3,VX3,VY3,VZ3,
     2                   VELJ3)
C
C     ESTABLISHES INITIAL CONDITIONS FOR NEW JET FROM MOMENTUM BALANCE
C
      PI = 3.1416
      X3 = (X1+X2)*.5
      Y3 = (Y1+Y2)*.5
      Z3 = (Z1+Z2)*.5
      XM1 = UJ1*V1*A1
      XM2 = UJ2*V2*A2
      DEN = XM1+XM2
      UJX = (XM1*UJ1*V1*VX1+XM2*UJ2*V2*VX2)/DEN
      UJY = (XM1*UJ1*V1*VY1+XM2*UJ2*V2*VY2)/DEN
      UJZ = (XM1*UJ1*V1*VZ1+XM2*UJ2*V2*VZ2)/DEN
      VELJ3 = SQRT (UJX*UJX+UJY*UJY+UJZ*UJZ)
      VX3 =  UJX/VELJ3
      VY3 =  UJY/VELJ3
```

```
      VZ3 =  UJZ/VELJ3
      A3 = DEN/VELJ3
      DJ3 = SQRT (4.*A3/(PI*FACT1))
      RETURN
      END


      SUBROUTINE OUTPT (XL,ZL,DXDZ,CF,DJ,XJ,YJ,ZJ,XB,YB,ZB,VX,VY,VZ)
C
C     TRANSFORMS LOCAL COORDINATES TO PROGRAM COORDINATES (FIXED)
C
      DIMENSION CF(3,3)
C
      PHI = ATAN(DXDZ)
      VXT = SIN(PHI)
      VYT = 0.
      VZT = COS(PHI)
      CALL ROTATE (VX,VY,VZ,CF,VXT,VYT,VZT,1)
      CALL ROTATE (FX,FY,FZ,CF,XL,0.,ZL,1)
      XB = FX*DJ+XJ
      YB = FY*DJ+YJ
      ZB = FZ*DJ+ZJ
      RETURN
      END


      SUBROUTINE VELOC (N1,N2,Z,X,DXDZ,UJ,D,UUE,XJ,YJ,ZJ,DJET,CF,PAR,
     1 XO,YO,ZO,UIF,VIF,WIF,D2XDZ2)
C
C     EVALUATES INDUCED VELOCITIES AT ONE CONTROL POINT (XO,YO,ZO IN
C     FIXED COORDINATE SYSTEM) FOR A GIVEN JET
C
      COMMON/BLK20/DIARAT,DREF
C
      DIMENSION Z(1),X(1),DXDZ(1),UJ(1),D(1),UUE(1),PAR(1)
      DIMENSION CF(3,3)
      DIMENSION D2XDZ2(1)
C
      E2 = PAR(2)
      E3 = PAR(3)
      F  = PAR(5)
      VELJ=PAR(6)
      PI = PAR(7)
      C1 = PAR(8)
      DR = PAR(9)
      N = N2-N1+1
      IF (N/2-(N+1)/2) 1,2,2
    1 M = (N-1)/2
      GO TO 3
    2 M = (N-2)/2
    3 XPT = (XO-XJ)/DJET
      YPT = (YO-YJ)/DJET
      ZPT = (ZO-ZJ)/DJET
      CALL ROTATE (XPT,YPT,ZPT,CF,A,B,C,0)
      UI = 0.
      VI = 0.
      WI = 0.
      M1 = M+1
      DO 21 K=N1,M1
```

```
      E1 = PAR(1)
      IF (K-M) 11,11,10
10    IF (N/2-(N+1)/2) 22,12,12
12    I = 2*K-1
      ZINCR = Z(I+1)-Z(I)
      GO TO 14
11    I = 2*K
      ZINCR = Z(I+1)-Z(I-1)
14    COST = 1./SQRT(1.+DXDZ(I)*DXDZ(I))
      SINT = SIGN(1.,DXDZ(I))*SQRT(1.-COST*COST)
      SIE = -((Z(I)-C)*COST+(X(I)-A)*SINT)
      ETA =  B
      ZETA=  (Z(I)-C)*SINT-(X(I)-A)*COST
      D1 = .5*D(I)
      DOUB1 = SIE*SIE+ETA*ETA+ZETA*ZETA
      DOUB2 = SQRT(DOUB1)
      UBLOCK = .5*D1*D1*ZINCR*COST*(1.-3.*ZETA*ZETA/DOUB1)/(DOUB1*DOUB2)
     1           -SINT*1.5*SIE*ZETA*D1*D1*ZINCR/(DOUB1*DOUB1*DOUB2)
      VBLOCK = -1.5*ZETA*ETA*D1*D1*ZINCR/(DOUB1*DOUB1*DOUB2)
      WBLOCK = -.5*D1*D1*ZINCR*SINT*(1.-3.*ZETA*ZETA/DOUB1)/(DOUB1*
     1           DOUB2)-COST*1.5*SIE*ZETA*D1*D1*ZINCR/(DOUB1*DOUB1*DOUB2)
      VELJE  = VELJ*UUE(I)
      CURV = D2XDZ2(I)/((1.+DXDZ(I)*DXDZ(I))**1.5)
      CURV = 3.*CURV*DREF/DJET
      E1 = E1-CURV/COST
      E = E2/(1.+E3*COST/(VELJE*UJ(I)))
      IF (VELJE*UJ(I)-SINT) 51,52,52
51    E = 0.
52    ZSO = (1.-DR)*VELJE*F/.75
      ZP = Z(I)+ZSO
      IF (ZP-VELJE*F) 47,60,60
47    IF (ZP-10.) 40,60,60
40    IF (ZP-.6*VELJE*F) 42,43,43
42    E = E*.1/.32
      GO TO 60
43    IF (ZP-.8*VELJE*F) 44,45,45
44    E = E*.12/.32
      GO TO 60
45    E = E*.21/.32
60    ZOVM = ZP/VELJE
      IF (ZOVM-F) 31,32,32
31    VARB = (1.-.375*ZOVM/F)
      VAR  = SQRT((1.+(1.-.75*ZOVM/F)**2)/2.)
      HT3  = .25*ZINCR*(E1+E*PI*VAR*(VELJE*UJ(I)-SINT)/COST)
      GO TO 33
32    VARB = .625
      HT3 = .25*ZINCR*(E1+E*(VELJE*UJ(I)-SINT)*C1/COST)
33    UBLOCK = UBLOCK*VARB
      VBLOCK = VBLOCK*VARB
      WBLOCK = WBLOCK*VARB
      Z1 = (C-Z(I))*(C-Z(I))+(A-X(I))*(A-X(I))
      Z2 = SQRT((B-D1)*(B-D1)+Z1)
      Z3 = SQRT((B+D1)*(B+D1)+Z1)
      USINK = -HT3*(X(I)-A)*((B-D1)/(Z1*Z2)-(B+D1)/(Z1*Z3))/PI
      VSINK = -HT3*(1./Z2-1./Z3)/PI
      WSINK = -HT3*(Z(I)-C)*((B-D1)/(Z1*Z2)-(B+D1)/(Z1*Z3))/PI
      IF (UUE(I)-1.) 6,5,6
 6    FACT = 1./UUE(I)
      UBLOCK = UBLOCK*FACT
```

```
      VBLOCK = VBLOCK*FACT
      WBLOCK = WBLOCK*FACT
      USINK  = USINK*FACT
      VSINK  = VSINK*FACT
      WSINK  = WSINK*FACT
    5 UI = UI+USINK+UBLOCK
      VI = VI+VSINK+VBLOCK
   21 WI = WI+WSINK+WBLOCK
   22 CALL ROTATE (UIF,VIF,WIF,CF,UI,VI,WI,1)
  691 FORMAT (6F12.5)
      RETURN
      END

      SUBROUTINE  DERIV (Z,FN,FPR,PAR)
C
C     COMPUTES DERIVATIVES FOR ADAMS PREDICTOR/CORRECTOR METHOD
C
      DIMENSION FN(1),FPR(1),PAR(1)
C
      E1 = PAR(1)
      E2 = PAR(2)
      E3 = PAR(3)
      F  = PAR(5)
      VELJ=PAR(6)
      PI = PAR(7)
      C1 = PAR(8)
      DR = PAR(9)
      UJ = FN(1)
      D  = FN(2)
      DXDZ=FN(4)
      COST = 1./SQRT(1.+DXDZ*DXDZ)
      SINT = SIGN(1.,DXDZ)*SQRT(1.-COST*COST)
      E = E2/(1.+E3*COST/(VELJ*UJ))
      IF (VELJ*UJ-SINT) 11,12,12
   11 E = 0.
   12 ZSO = (1.-DR)*VELJ*F/.75
      ZP = Z+ZSO
      IF (ZP-VELJ*F)  47,60,60
   47 IF (ZP-10.) 40,60,60
   40 IF (ZP-.6*VELJ*F)  42,43,43
   42 E = E*.1/.32
      GO TO 60
   43 IF (ZP-.8*VELJ*F)  44,45,45
   44 E = E*.12/.32
      GO TO 60
   45 E = E*.21/.32
   60 ZOVM = ZP/VELJ
      IF (ZOVM-F) 22,23,23
   22 VAR = SQRT((1.+(1.-.75*ZOVM/F)**2)/2.)
      XT = 1.-.75*ZOVM/F
      XT = 1./XT
      CD = (-XT*XT+6.6*XT+.4)/6.
      VAR1 = E1*COST+E*(VELJ*UJ-SINT)*PI*VAR
      VAR2 = VELJ*VELJ*COST
      VAR3 = .25*PI*(1.-.75*ZOVM/F)*UJ*D
      DUJ  = (VAR1*SINT/VAR2-VAR1*UJ/(VELJ*COST))/VAR3
      DD   = (VAR1*D/(VELJ*COST)+3.*PI*D*D*UJ/(16.*F*VELJ)-VAR3*D*DUJ/
     1        UJ)/(2.*VAR3)
```

```
      VAR4 = (E1+.5*CD)*COST+E*(VELJ*UJ-SINT)*PI*VAR
      DDXDZ= VAR4/(VAR2*COST*VAR3*UJ)
      GO TO 15
   23 VAR1 = E1*COST+E*(VELJ*UJ-SINT)*C1
      CD = 1.8
      DUJ  = 16.*VAR1*(SINT/(VELJ*VELJ*COST)-UJ/(VELJ*COST))/(PI*D*UJ)
      DD   = 8.*(VAR1/(VELJ*COST)-PI*D*DUJ/16.)/(PI*UJ)
      VAR4 = (E1+.5*CD)*COST+E*(VELJ*UJ-SINT)*C1
      DDXDZ= 16.*VAR4/(PI*VELJ*VELJ*D*UJ*UJ*COST*COST)
   15 CONTINUE
      PAR(10) = DDXDZ
      FPR(1) = DUJ
      FPR(2) = DD
      FPR(3) = DXDZ
      FPR(4) = DDXDZ
      RETURN
      END


      SUBROUTINE TRWING (NTHT,NSMAX,NCOEF,IRECT,XO,YO,ZO,NK)
C
C     ESTABLISHES CONTROL POINTS IN THE BODY FIXED COORDINATES FOR WING
C     *A* IS THE REAL PART OF EACH COMPLEX COEFFICIENT
C     *B* IS THE IMAGINARY PART OF EACH COMPLEX COEFFICIENT
C     MAPPING AROUND 360DEG IS SPECIFIED
C     IRECT=0,RECTANGULAR WING, IRECT=1,NON-RECTANGULAR WING
C
      DIMENSION COEFR(15,25),COEFI(15,25)
      DIMENSION Y(25),RADIUS(25),DRDZ(25)
C
      COMMON/BLK1/Y,RADIUS,DRDZ,COEFR,COEFI
C
      DIMENSION XO(1),YO(1),ZO(1)
      DIMENSION A(15),B(15)
C
      XN = NTHT
      DTHT = 6.2832/XN
      DO 30 I=1,NSMAX
      READ (5,503) Y(I),RADIUS(I),DRDZ(I)
      IF (I-1) 2,2,3
    3 IF (IRECT) 4,4,2
    2 READ (5,502) (A(K),B(K),K=1,NCOEF)
      GO TO 10
    4 DO 8 J=1,NTHT
      JG = (I-1)*NTHT+J
      NS1 = JG-NTHT
      XO(JG) = XO(NS1)
      YO(JG) = Y(I)
    8 ZO(JG) = ZO(NS1)
      GO TO 25
   10 RW = RADIUS(I)
      DO 20  J=1,NTHT
      XJ1 = J-1
      THETA = XJ1*DTHT
      TERM1 = RW*COS(THETA)+A(2)
      TERM2 = RW*SIN(THETA)+B(2)
      RWJ = 1.
      DO 15 K=3,NCOEF
      XK = K-2
```

```
      COSTH = COS(XK*THETA)
      SINTH = SIN(XK*THETA)
      RWJ = RWJ/RW
      TERM1 = TERM1+(A(K)*COSTH+B(K)*SINTH)*RWJ
   15 TERM2 = TERM2+(-A(K)*SINTH+B(K)*COSTH)*RWJ
      JG = (I-1)*NTHT+J
      XO(JG) = TERM1
      YO(JG) = Y(I)
   20 ZO(JG) = TERM2
   25 DO 26 K=1,NCOEF
      COEFR(K,I) = A(K)
   26 COEFI(K,I) = B(K)
   30 CONTINUE
      NK = NTHT*NSMAX
      RETURN
  502 FORMAT (6E12.5)
  503 FORMAT(6F12.0)
      END


      SUBROUTINE TRBODY (NTHT,NSMAX,NCOEF,NSYM,XO,YO,ZO,NK)
C
C     ESTABLISHES CONTROL POINTS IN BODY-FIXED COORDINATES FOR BODY
C     *A* IS THE REAL PART OF EACH COMPLEX COEFFICIENT
C     BODY MUST BE SYMMETRIC
C     MAPPING DONE FOR 180DEG IF FLOW IS SYMMETRIC,FOR 360DEG IF FLOW
C     IS NOT SYMMETRIC
C
      DIMENSION COEFR(15,25),COEFI(15,25)
      DIMENSION X(25),RADIUS(25),DRDX(25)
C
      COMMON/BLK1/X,RADIUS,DRDX,COEFR,COEFI
C
      DIMENSION XO(1),YO(1),ZO(1)
      DIMENSION A(15)
C
      XN = NTHT
      XSYM = NSYM+1
      DTHT = XSYM*3.1416/XN
      IF (NSYM) 1,1,2
    1 NTHT = NTHT+1
    2 CONTINUE
      DO 30 I=1,NSMAX
      READ (5,503) X(I),RADIUS(I),DRDX(I)
      READ (5,502) (A(K),K=1,NCOEF)
      RB = RADIUS(I)
      DO 20 J=1,NTHT
      XJ1 = J-1
      THETA = XJ1*DTHT
      TERM1 = RB*SIN(THETA)
      TERM2 =-RB*COS(THETA)-A(2)
      RBJ = 1.
      DO 15 K=3,NCOEF
      XK = K-2
      COSTH = COS(XK*THETA)
      SINTH = SIN(XK*THETA)
      RBJ = RBJ/RB
      TERM1 = TERM1-A(K)*SINTH*RBJ
   15 TERM2 = TERM2-A(K)*COSTH*RBJ
```

```
      JG = (I-1)*NTHT+J
      XO(JG) = X(I)
      YO(JG) = TERM1
   20 ZO(JG) = TERM2
      DO 22 K=1,NCOEF
   22 COEFR(K,I) = A(K)
   30 CONTINUE
      NK = NTHT*NSMAX
      RETURN
  502 FORMAT (6E12.5)
  503 FORMAT(6F12.0)
      END

      SUBROUTINE ADAPT (U,V,W,NTHT,NSMAX,NCOEF,IGEOM)
C
C     PUNCHES OUT DATA TO SERVE AS INPUT TO THE TRANSFORMATION METHOD
C     DATA IN SETS BY X OR Y STATIONS. DATA CONSISTS OF  STATION,
C     RADIUS OF MAPPING CIRCLE, SLOPE, COEFFICIENTS AND VELOCITIES
C
      DIMENSION COEFR(15,25),COEFI(15,25)
      DIMENSION STATN(25),RADIUS(25),SLP3D(25)
C
      COMMON/BLK1/STATN,RADIUS,SLP3D,COEFR,COEFI
C
      DIMENSION U(1),V(1),W(1)
C
      DIMENSION WRTV(3)
C
      DATA WRTV/1HU,1HV,1HW/
C
      DO 50 I=1,NSMAX
      WRITE (7,701) STATN(I),RADIUS(I),SLP3D(I),I
      IF (IGEOM-1) 3,3,2
    2 NP = NCOEF/6
      IND = NP*6-NCOEF
      JPS = 1
      DO 4 J=1,NP
      JPF = JPS+5
      WRITE (7,702) (COEFR(K,I),K=JPS,JPF),I,J
    4 JPS = JPS+6
      IF (IND) 5,10,10
    5 NP1 = NP+1
      JPF = NCOEF
      NOP = JPF-JPS+1
      GO TO (61,62,63,64,65),NOP
   61 WRITE (7,711) (COEFR(K,I),K=JPS,JPF),I,NP1
      GO TO 70
   62 WRITE (7,712) (COEFR(K,I),K=JPS,JPF),I,NP1
      GO TO 70
   63 WRITE (7,713) (COEFR(K,I),K=JPS,JPF),I,NP1
      GO TO 70
   64 WRITE (7,714) (COEFR(K,I),K=JPS,JPF),I,NP1
      GO TO 70
   65 WRITE (7,715) (COEFR(K,I),K=JPS,JPF),I,NP1
   70 CONTINUE
      GO TO 10
    3 NP = NCOEF/3
      IND = NP*3-NCOEF
```

```
      JPS = 1
      DO 6 J=1,NP
      JPF = JPS+2
      WRITE (7,702) (COEFR(K,I),COEFI(K,I),K=JPS,JPF),I,J
    6 JPS = JPS+3
      IF (IND) 7,10,10
    7 NP1 = NP+1
      JPF = NCOEF
      NOP = JPF-JPS+1
      GO TO (71,72),NOP
   71 WRITE (7,712) (COEFR(K,I),COEFI(K,I),K=JPS,JPF),I,NP1
      GO TO 80
   72 WRITE (7,714) (COEFR(K,I),COEFI(K,I),K=JPS,JPF),I,NP1
   80 CONTINUE
   10 KOUNT = 1
      NP = NTHT/6
      IND = NP*6-NTHT
   11 JPS = (I-1)*NTHT+1
      DO 12 J=1,NP
      JPF = JPS+5
      WRITE (7,703) (U(L),L=JPS,JPF),WRTV(KOUNT),I,J
   12 JPS = JPS+6
      IF (IND) 14,15,15
   14 NP1 = NP+1
      JPF = I*NTHT
      NOP = JPF-JPS+1
      GO TO (81,82,83,84,85),NOP
   81 WRITE (7,721) (U(L),L=JPS,JPF),WRTV(KOUNT),I,NP1
      GO TO 90
   82 WRITE (7,722) (U(L),L=JPS,JPF),WRTV(KOUNT),I,NP1
      GO TO 90
   83 WRITE (7,723) (U(L),L=JPS,JPF),WRTV(KOUNT),I,NP1
      GO TO 90
   84 WRITE (7,724) (U(L),L=JPS,JPF),WRTV(KOUNT),I,NP1
      GO TO 90
   85 WRITE (7,725) (U(L),L=JPS,JPF),WRTV(KOUNT),I,NP1
   90 CONTINUE
   15 IF (KOUNT-2) 20,25,50
   20 NSTART = (I-1)*NTHT+1
      NFIN  = I*NTHT
      DO 21 ID=NSTART,NFIN
   21 U(ID) = V(ID)
      KOUNT = KOUNT+1
      GO TO 11
   25 DO 26 ID= NSTART,NFIN
   26 U(ID) = W(ID)
      KOUNT = KOUNT+1
      GO TO 11
   50 CONTINUE
      RETURN
  701 FORMAT (3F12.6,I41)
  702 FORMAT (6E12.5,I5,I3)
  711 FORMAT (1E12.5,I65,I3)
  712 FORMAT (2E12.5,I53,I3)
  713 FORMAT (3E12.5,I41,I3)
  714 FORMAT (4E12.5,I29,I3)
  715 FORMAT (5E12.5,I17,I3)
  703 FORMAT (6E12.5,1X,A1,2I3)
  721 FORMAT (1E12.5,61X,A1,2I3)
```

```
 722  FORMAT (2E12.5,49X,A1,2I3)
 723  FORMAT (3E12.5,37X,A1,2I3)
 724  FORMAT (4E12.5,25X,A1,2I3)
 725  FORMAT (5E12.5,13X,A1,2I3)
      END

      SUBROUTINE PRTOUT (IGEOM,X0,Y0,Z0,U,V,W,CP,NK,NTHT)
C
C     PRINTS OUT COMPUTED ANSWERS. INFORMATION INCLUDES JET CENTERLINE
C     DATA AND INDUCED VELOCITIES AT CONTROL POINTS
C
      DIMENSION X1(100),Z1(100),UJ1(100),D1(100),DXDZ1(100)
      DIMENSION X2(100),Z2(100),UJ2(100),D2(100),DXDZ2(100)
      DIMENSION X3(100),Z3(100),UJ3(100),D3(100),DXDZ3(100)
      DIMENSION X4(100),Z4(100),UJ4(100),D4(100),DXDZ4(100)
      DIMENSION X5(100),Z5(100),UJ5(100),D5(100),DXDZ5(100)
      DIMENSION XBAS1(100),YBAS1(100),ZBAS1(100)
      DIMENSION XBAS2(100),YBAS2(100),ZBAS2(100)
      DIMENSION XBAS3(100),YBAS3(100),ZBAS3(100)
      DIMENSION XBAS4(100),YBAS4(100),ZBAS4(100)
      DIMENSION XBAS5(100),YBAS5(100),ZBAS5(100)
C
      COMMON/BLK3/X1,Z1,UJ1,D1,DXDZ1,X2,Z2,UJ2,D2,DXDZ2
      COMMON/BLK4/X3,Z3,UJ3,D3,DXDZ3,X4,Z4,UJ4,D4,DXDZ4
      COMMON/BLK5/X5,Z5,UJ5,D5,DXDZ5
      COMMON/BLK6/XBAS1,YBAS1,ZBAS1,XBAS2,YBAS2,ZBAS2,XBAS3,YBAS3,ZBAS3
      COMMON/BLK7/XBAS4,YBAS4,ZBAS4,XBAS5,YBAS5,ZBAS5
      COMMON/BLK9/MULT,IHOLD1,IHOLD2,IHOLD3,KOUNT1,KOUNT2
      COMMON/BLK10/IONE,ITWO,ITHR,IFOUR,IFIV,N1,N2,N3,N4,N5
      COMMON/BLK13/XJ3,YJ3,ZJ3,DJET3,VELJ3,XJ4,YJ4,ZJ4,DJET4,VELJ4
      COMMON/BLK14/XJ5,YJ5,ZJ5,DJET5,VELJ5
C
      DIMENSION X0(1),Y0(1),Z0(1),U(1),V(1),W(1),CP(1)
C
      WRITE (6,601)
 601  FORMAT (1H0,///)
      IF (MULT-2) 1,2,3
   1  WRITE (6,602)
 602  FORMAT (1H0,46X,27H** SINGLE JET CENTERLINE **)
      GO TO 20
   2  WRITE (6,603)
 603  FORMAT (1H0,43X,33H** CENTERLINES OF JETS 1 AND 2 **)
      GO TO 4
   3  WRITE (6,604)
 604  FORMAT (1H0,42X,35H** CENTERLINES OF JETS 1,2 AND 3 **)
   4  IF (MULT-2) 5,5,6
   5  IF (IHOLD1-2) 20,7,7
   7  WRITE (6,605)
 605  FORMAT (1H ,51X,17HAND COALESCED JET)
      GO TO 20
   6  IF (IHOLD1-2) 10,8,8
   8  WRITE (6,606)
 606  FORMAT (1H ,37X,46HTHE JET RESULTING FROM COALESCENCE OF JETS 1,2)
      GO TO 16
  10  IF (IHOLD2-2) 15,9 9
   9  WRITE (6,607)
 607  FORMAT (1H ,37X,46HTHE JET RESULTING FROM COALESCENCE OF JETS 2,3)
  15  IF (IHOLD3-2) 20,11,11
```

```
 11  WRITE (6,608)
608  FORMAT (1H ,26X,70HTHE JET RESULTING FROM COALESCENCE OF JET 1 AND
    1THE JET DESCRIBED ABOVE)
     GO TO 20
 16  IF (IHOLD3-2) 20,12,12
 12  WRITE (6,609)
609  FORMAT (1H ,26X,70HTHE JET RESULTING FROM COALESCENCE OF THE ABOVE
    1DESCRIBED JET AND JET 3)
 20  CONTINUE
     WRITE (6,630)
630  FORMAT (1H0,45X,32H********************************//)
     IF (MULT.GE.1) WRITE (6,610)
     IF (MULT.GE.2) WRITE (6,611)
     IF (MULT.GE.3) WRITE (6,617)
610  FORMAT (1H0,3X,6HXCOORD,3X,6HYCOORD,3X,6HZCOORD,3X,2HUJ,4X,3HDIA)
611  FORMAT (1H+,42X,6HXCOORD,3X,6HYCOORD,3X,6HZCOORD,3X,2HUJ,4X,3HDIA)
617  FORMAT (1H+,81X,6HXCOORD,3X,6HYCOORD,3X,6HZCOORD,3X,2HUJ,4X,3HDIA)
     WRITE (6,612)
612  FORMAT (1H0)
     IF (MULT-2) 30,40,60
 30  CONTINUE
     WRITE (6,616) (XBAS1(I),YBAS1(I),ZBAS1(I),UJ1(I),D1(I), I=1,N1)
616  FORMAT (1H ,1X,F8.2,1X,F8.2,1X,F8.2,1X,F5.3,1X,F5.2)
     GO TO 90
 40  IF (N1-N2) 41,42,42
 41  IP1 = N1
     IP2 = N2
     GO TO 43
 42  IP1 = N2
     IP2 = N1
 43  CONTINUE
     DO 47 I=1,IP1
 47  WRITE (6,613) XBAS1(I),YBAS1(I),ZBAS1(I),UJ1(I),D1(I),XBAS2(I),
    1 YBAS2(I),ZBAS2(I),UJ2(I),D2(I)
613  FORMAT (1H ,1X,F8.2,1X,F8.2,1X,F8.2,1X,F5.3,1X,F5.2,1X,F8.2,1X,
    1 F8.2,1X,F8.2,1X,F5.3,1X,F5.2,1X,F8.2,1X,F8.2,1X,F8.2,1X,F5.3,1X,
    2 F5.2)
     IF (N1-N2) 48,50,44
 48  IPP = IP1+1
     DO 45 I=IPP,IP2
 45  WRITE (6,614) XBAS2(I),YBAS2(I),ZBAS2(I),UJ2(I),D2(I)
614  FORMAT (1H ,40X,F8.2,1X,F8.2,1X,F8.2,1X,F5.3,1X,F5.2,1X,F8.2,1X,
    1 F8.2,1X,F8.2,1X,F5.3,1X,F5.2)
     GO TO 50
 44  IPP = IP1+1
     DO 46 I=IPP,IP2
 46  WRITE (6,613) XBAS1(I),YBAS1(I),ZBAS1(I),UJ1(I),D1(I)
 50  CONTINUE
     IF (IHOLD1-2) 90,51,51
 51  CONTINUE
     V3 = 1./VELJ3
     ZP = YJ3
     YP = -ZJ3
     WRITE (6,615) XJ3,YP,ZP,V3,DJET3
615  FORMAT (1H0,3X,27HPROPERTIES OF COALESCED JET,3X,2HX=,F9.2,3X,2HY=
    1,F8.2,3X,2HZ=,F8.2,3X,6HU/UJO=,F5.2,3X,5HD/DO=,F5.2)
     WRITE (6,610)
     WRITE (6,616) (XBAS3(I),YBAS3(I),ZBAS3(I),UJ3(I),D3(I), I=1,N3)
     GO TO 90
```

```
 60  CONTINUE
     IF (N1-N2) 61,72,62
 61  IF (N1-N3) 63,80,64
 63  IP1 = N1
     IND1 = 1
     IF (N2-N3) 65,76,66
 65  IP2 = N2
     IP3 = N3
     IND2 = 2
     GO TO 70
 66  IP2 = N3
     IP3 = N2
     IND2 = 3
     GO TO 70
 64  IP1 = N3
     IP2 = N1
     IP3 = N2
     IND1 = 3
     IND2= 1
     GO TO 70
 62  IF (N2-N3) 67,76,68
 67  IP1 = N2
     IND1 = 2
     IF (N1-N3) 69,80,71
 69  IP2 = N1
     IP3 = N3
     IND2 = 1
     GO TO 70
 71  IP2 = N3
     IP3 = N2
     IND2 = 3
     GO TO 70
 68  IP1 = N3
     IP2 = N2
     IP3 = N1
     IND1= 3
     IND2= 2
     GO TO 70
 72  IND1 =-1
     IF (N1-N3) 73,74,75
 73  IP1= N1
     IP3= N3
     IND2 = 3
     GO TO 70
 74  IND1 = 0
     IP1 = N1
     GO TO 70
 75  IP1 = N3
     IP3 = N1
     IND2 = 1
     GO TO 70
 76  IND1 =-2
     IF (N1-N2) 77,74,78
 77  IP1 = N1
     IP3 = N3
     IND2 = 3
     GO TO 70
 78  IP1 = N2
     IP3 = N1
```

```
      IND2 = 1
      GO TO 70
 80   IND1 =-3
      IF (N1-N2) 81,74,82
 81   IP1 = N1
      IP3 = N2
      IND2 = 2
      GO TO 70
 82   IP1 = N2
      IP3 = N1
      IND2 = 1
 70   CONTINUE
      DO 85 I=1,IP1
 85   WRITE (6,613) XBAS1(I),YBAS1(I),ZBAS1(I),UJ1(I),D1(I),XBAS2(I),
     1 YBAS2(I),ZBAS2(I),UJ2(I),D2(I),XBAS3(I),YBAS3(I),ZBAS3(I),UJ3(I),
     2 D3(I)
      IF (IND1) 120,150,100
100   IF (IND1-2) 101,102,103
101   IPP = IP1+1
      DO 111 I=IPP,IP2
111   WRITE (6,614) XBAS2(I),YBAS2(I),ZBAS2(I),UJ2(I),D2(I),XBAS3(I),
     1 YBAS3(I),ZBAS3(I),UJ3(I),D3(I)
      IF (IND2-2) 104,104,105
104   IPP = IP2+1
      DO 106 I=IPP,IP3
106   WRITE  (6,618) XBAS3(I),YBAS3(I),ZBAS3(I),UJ3(I),D3(I)
618   FORMAT (1H ,79X,F8.2,1X,F8.2,1X,F8.2,1X,F5.3,1X,F5.2)
      GO TO 150
105   IPP = IP2+1
      DO 107 I=IPP,IP3
107   WRITE (6,614) XBAS2(I),YBAS2(I),ZBAS2(I),UJ2(I),D2(I)
      GO TO 150
102   CONTINUE
      IPP = IP1+1
      DO 110 I=IPP,IP2
110   WRITE (6,620) XBAS1(I),YBAS1(I),ZBAS1(I),UJ1(I),D1(I),XBAS3(I),
     1 YBAS3(I),ZBAS3(I),UJ3(I),D3(I)
620   FORMAT (1H ,1X,F8.2,1X,F8.2,1X,F8.2,1X,F5.3,1X,F5.2,40X,F8.2,1X,
     1 F8.2,1X,F8.2,1X,F5.3,1X,F5.2)
      IF (IND2-2) 104,104,108
108   IPP = IP2+1
      DO 112 I=IPP,IP3
112   WRITE (6,613) XBAS1(I),YBAS1(I),ZBAS1(I),UJ1(I),D1(I)
      GO TO 150
103   CONTINUE
      IPP = IP1+1
      DO 109 I=IPP,IP2
109   WRITE (6,613) XBAS1(I),YBAS1(I),ZBAS1(I),UJ1(I),D1(I),XBAS2(I),
     1 YBAS2(I),ZBAS2(I),UJ2(I),D2(I)
      IF (IND2-2) 105,108,108
150   CONTINUE
      IF (IHOLD1-2) 151,152,152
151   IF (IHOLD2-2)  90,153,153
152   IF (N4) 170,170,154
154   V4 = 1./VELJ4
      ZP = YJ4
      YP = -ZJ4
      WRITE (6,621) XJ4,YP,ZP,V4,DJET4
621   FORMAT (1H0,3X,41HJET FORMED BY COALESCENCE OF JETS 1 AND 2,3X,
```

```
     1 2HX=,F9.2,3X,2HY=,F8.2,3X,2HZ=,F8.2,3X,6HU/UJO=,F5.2,3X,5HD/DO=,
     2 F5.2)
      GO TO 158
153   IF (N4) 170,170,155
155   V4 = 1./VELJ4
      ZP = YJ4
      YP = -ZJ4
      WRITE (6,622) XJ4,YP,ZP,V4,DJET4
622   FORMAT (1H0,3X,41HJET FORMED BY COALESCENCE OF JETS 2 AND 3,3X,
     1 2HX=,F9.2,3X,2HY=,F8.2,3X,2HZ=,F8.2,3X,6HU/UJO=,F5.2,3X,5HD/DO=,
     2 F5.2)
158   WRITE (6,610)
      WRITE (6,616) (XBAS4(I),YBAS4(I),ZBAS4(I),UJ4(I),D4(I), I=1,N4)
170   CONTINUE
      IF (IHOLD3-2) 90,171,171
171   V5 = 1./VELJ5
      ZP = YJ5
      YP = -ZJ5
      WRITE (6,615) XJ5,YP,ZP,V5,DJET5
      WRITE (6,610)
      WRITE (6,616) (XBAS5(I),YBAS5(I),ZBAS5(I),UJ5(I),D5(I), I=1,N5)
      GO TO 90
120   CONTINUE
      IF (IABS(IND1)-2) 130,135,140
130   IF (IND2-2) 121,121,123
121   IPP = IP1+1
      DO 122 I=IPP,IP3
122   WRITE (6,613) XBAS1(I),YBAS1(I),ZBAS1(I),UJ1(I),D1(I),XBAS2(I),
     1 YBAS2(I),ZBAS2(I),UJ2(I),D2(I)
      GO TO 150
123   IP2 = IP1
      GO TO 104
135   IF (IND2-2) 124,126,126
124   IP2 = IP1
      GO TO 108
126   IPP = IP1+1
      DO 127 I=IPP,IP3
127   WRITE (6,614) XBAS2(I),YBAS2(I),ZBAS2(I),UJ2(I),D2(I),XBAS3(I),
     1 YBAS3(I),ZBAS3(I),UJ3(I),D3(I)
      GO TO 150
140   IF (IND2-2) 142,141,142
141   IP2 = IP1
      GO TO 105
142   IPP = IP1+1
      DO 143 I=IPP,IP3
143   WRITE (6,620) XBAS1(I),YBAS1(I),ZBAS1(I),UJ1(I),D1(I),XBAS3(I),
     1 YBAS3(I),ZBAS3(I),UJ3(I),D3(I)
      GO TO 150
 90   CONTINUE
      IF (IGEOM) 200,99,200
200   WRITE (6,640)
640   FORMAT (1H1)
      IF (IGEOM-2) 201,202,203
201   CONTINUE
      WRITE (6,631)
631   FORMAT (1H0,44X,34H*** INDUCED VELOCITIES ON WING ***)
632   FORMAT (1H0,27X,1HX,8X,1HY,8X,1HZ,12X,1HU,14X,1HV,14X,1HW/)
      GO TO 205
202   CONTINUE
```

```
      WRITE (6,633)
  633 FORMAT (1H0,44X,34H*** INDUCED VELOCITIES ON BODY ***)
  205 CONTINUE
      WRITE (6,630)
      WRITE (6,632)
      KOUNT = 1
      DO 210 I=1,NK
      WRITE (6,634) XO(I),YO(I),ZO(I),U(I),V(I),W(I)
  634 FORMAT (1H ,21X,F9.3,1X,F9.3,1X,F9.3,3E15.5)
      IF (I-KOUNT*NTHT) 210,206,210
  206 KOUNT = KOUNT+1
      WRITE  (6,630)
      WRITE (6,640)
      IF (I-NK) 214,210,210
  214 CONTINUE
      IF (IGEOM-2) 211,212,212
  211 WRITE (6,631)
      GO TO 213
  212 WRITE (6,633)
  213 WRITE (6,630)
      WRITE (6,632)
  210 CONTINUE
      GO TO 99
  203 CONTINUE
      WRITE (6,635)
  635 FORMAT (1H0,38X,44H*** INDUCED VELOCITIES AT CONTROL POINTS ***)
      IF (IGEOM-3) 221,221,222
  221 WRITE (6,632)
      WRITE (6,634) (XO(I),YO(I),ZO(I),U(I),V(I),W(I), I=1,NK)
      GO TO 99
  222 WRITE (6,636)
  636 FORMAT (1H ,40X,39HPRESSURE COEFFICIENTS AT CONTROL POINTS)
      WRITE (6,637)
  637 FORMAT (1H0,20X,1HX,8X,1HY,8X,1HZ,12X,2HCP,14X,1HU,14X,1HV,14X,
     1 1HW/)
      WRITE (6,638) (XO(I),YO(I),ZO(I),CP(I),U(I),V(I),W(I), I=1,NK)
  638 FORMAT (1H ,14X,F9.3,1X,F9.3,1X,F9.3,4E15.5)
   99 CONTINUE
      RETURN
      END

      SUBROUTINE TRANS1 (MULT,ALFA,BETA,PSID)
C
C     TRANSFORMS INPUT COORDINATES TO PROGRAM COORDINATES (FIXED)
C     CONVERTS ANGLE OF ATTACK AND SIDESLIP TO FRSTRM DIRECTION COS.
C
      COMMON/BLK8/ALFQ,BETQ,GETQ,F1,F2,F3,F4,F5,VKONST
      COMMON/BLK12/XJ1,YJ1,ZJ1,DJET1,VELJ1,XJ2,YJ2,ZJ2,DJET2,VELJ2
      COMMON/BLK13/XJ3,YJ3,ZJ3,DJET3,VELJ3,XJ4,YJ4,ZJ4,DJET4,VELJ4
C
      DIMENSION PSID(1)
C
      A = ALFA*.0174533
      B = BETA*.0174533
      ALFQ = COS(A)*COS(B)
      BETQ = SIN(A)*COS(B)
      GETQ = SIN(B)
      YS = YJ1
```

```
      YJ1 = ZJ1
      ZJ1=-YS
      PSID(1) = -PSID(1)
      IF (MULT-2) 5,4,3
    3 YS = YJ3
      YJ3 = ZJ3
      ZJ3 =-YS
      PSID(3) = -PSID(3)
    4 YS = YJ2
      YJ2 = ZJ2
      ZJ2 =-YS
      PSID(2) = -PSID(2)
    5 CONTINUE
      RETURN
      END


      SUBROUTINE VEL1 (MULT,ALFA,VK1,VK2)
C
C     COMPUTES EFFECTIVE VELOCITY RATIO FOR DOWNSTREAM JET AT EXIT
C
      COMMON/BLK8/ALFQ,BETQ,GETQ,F1,F2,F3,F4,F5,VKONST
      COMMON/BLK12/XJ1,YJ1,ZJ1,DJET1,VELJ1,XJ2,YJ2,ZJ2,DJET2,VELJ2
      COMMON/BLK13/XJ3,YJ3,ZJ3,DJET3,VELJ3,XJ4,YJ4,ZJ4,DJET4,VELJ4
      COMMON/BLK16/V2X1,V2Y1,V2Z1,V2X2,V2Y2,V2Z2,V2X3,V2Y3,V2Z3
C
      VELJ1 = 1./VELJ1
      IF (MULT-2) 5,1,1
    1 VELJ2 = 1./VELJ2
      DOTP = (XJ2-XJ1)*ALFQ+(YJ2-YJ1)*BETQ+(ZJ2-ZJ1)*GETQ
      DEN = SQRT((XJ2-XJ1)**2+(YJ2-YJ1)**2+(ZJ2-ZJ1)**2)
      DOTP = DOTP/DEN
      IF (ABS(DOTP)-.02) 10,10,11
   10 VK1 = 1.
      GO TO 15
   11 CONTINUE
      A = ALFA*.0174533
      ALF = COS(A)
      BET = SIN(A)
      GET = 0.
      CALL XPROD (V2X1,V2Y1,V2Z1,ALF,BET,GET,XT1,YT1,ZT1)
      CALL XPROD (XT1,YT1,ZT1,ALF,BET,GET,CFNX,CFNY,CFNZ)
      CALL PLANE (CFNX,CFNY,CFNZ,XJ1,YJ1,ZJ1,V2X2,V2Y2,V2Z2,XJ2,YJ2,ZJ2,
     1 XI,YI,ZI)
      S = SQRT ((XJ1-XI)**2 +(YJ1-YI)**2 +(ZJ1-ZI)**2)/DJET1
      VK1 = (S+.75)/(S-1.)
   15 CONTINUE
      IF (MULT-2) 5,5,2
    2 VELJ3 = 1./VELJ3
      IF (ABS(DOTP)-.02) 12,12,14
   12 VK2 = 1.
      GO TO 5
   14 CONTINUE
      CALL PLANE (CFNX,CFNY,CFNZ,XJ1,YJ1,ZJ1,V2X3,V2Y3,V2Z3,XJ3,YJ3,ZJ3,
     1 XI,YI,ZI)
      S = SQRT ((XJ1-XI)**2 +(YJ1-YI)**2 +(ZJ1-ZI)**2)/DJET1
      VK2 = (S+.75)/(S-1.)
      CALL XPROD (V2X2,V2Y2,V2Z2,ALF,BET,GET,XT1,YT1,ZT1)
      CALL XPROD (XT1,YT1,ZT1,ALF,BET,GET,CFNX,CFNY,CFNZ)
```

```
      CALL PLANE (CFNX,CFNY,CFNZ,XJ2,YJ2,ZJ2,V2X3,V2Y3,V2Z3,XJ3,YJ3,ZJ3,
     1 XI,YI,ZI)
      S = SQRT ((XJ2-XI)**2 +(YJ2-YI)**2 +(ZJ2-ZI)**2)/DJET1
      VK2 = (S+.75)/(S-1.)*VK2
    5 CONTINUE
      RETURN
      END


      SUBROUTINE TRANS2 (Y,Z,NO)
C
C     TRANSFORMS INPUT COORDINATES TO PROGRAM COORDINATES (FIXED)
C
      DIMENSION Y(1),Z(1)
C
      DO 1 I=1,NO
      YS = Y(I)
      Y(I) = Z(I)
    1 Z(I) = -YS
      RETURN
      END


      SUBROUTINE TRANS3 (Y,Z,V,W,NO)
C
C     TRANSFORMS PROGRAM COORDINATES (FIXED) TO OUTPUT COORDINATES.
C     JET CENTERLINE AND CONTROL POINT COORDINATES ARE AFFECTED
C
      DIMENSION XBAS1(100),YBAS1(100),ZBAS1(100)
      DIMENSION XBAS2(100),YBAS2(100),ZBAS2(100)
      DIMENSION XBAS3(100),YBAS3(100),ZBAS3(100)
      DIMENSION XBAS4(100),YBAS4(100),ZBAS4(100)
      DIMENSION XBAS5(100),YBAS5(100),ZBAS5(100)
C
      COMMON/BLK6/XBAS1,YBAS1,ZBAS1,XBAS2,YBAS2,ZBAS2,XBAS3,YBAS3,ZBAS3
      COMMON/BLK7/XBAS4,YBAS4,ZBAS4,XBAS5,YBAS5,ZBAS5
      COMMON/BLK10/IONE,ITWO,ITHR,IFOUR,IFIV,N1,N2,N3,N4,N5
C
      DIMENSION Y(1),Z(1),V(1),W(1)
C
      DO 1 I=1,NO
      YS = Y(I)
      Y(I) = -Z(I)
      Z(I) = YS
      VS = V(I)
      V(I) = -W(I)
    1 W(I) = VS
      DO  2 I=1,N1
      YS = YBAS1(I)
      YBAS1(I) = -ZBAS1(I)
    2 ZBAS1(I) = YS
      IF (N2) 3,10,3
    3 DO 4 I=1,N2
      YS = YBAS2(I)
      YBAS2(I) = -ZBAS2(I)
    4 ZBAS2(I) = YS
   10 IF (N3) 5,20,5
    5 DO 6 I=1,N3
      YS = YBAS3(I)
```

```
      YBAS3(I) = -ZBAS3(I)
    6 ZBAS3(I) = YS
   20 IF (N4) 7,30,7
    7 DO 8 I=1,N4
      YS = YBAS4(I)
      YBAS4(I) = -ZBAS4(I)
    8 ZBAS4(I) = YS
   30 IF (N5) 9,40,9
    9 DO 11 I=1,N5
      YS = YBAS5(I)
      YBAS5(I) = -ZBAS5(I)
   11 ZBAS5(I) = YS
   40 CONTINUE
      RETURN
      END


      SUBROUTINE PLANE (CFN1,CFN2,CFN3,X1,Y1,Z1,CSN1,CSN2,CSN3,XL1,XL2,
     1                  XL3,COOR1,COOR2,COOR3)
C
C     COMPUTES INTERSECTION OF A GIVEN PLANE WITH A LINE
C
      DIMENSION CFN(3),CSN(3),XL(3),COOR(3)
C
      CFN(1) = CFN1
      CFN(2) = CFN2
      CFN(3) = CFN3
      CSN(1) = CSN1
      CSN(2) = CSN2
      CSN(3) = CSN3
      XL(1)  = XL1
      XL(2)  = XL2
      XL(3)  = XL3
      IL = 1
      IM = 1
      IN = 1
      SUB1 = 0.
      IF (ABS(CSN(1))-1.0E-04) 1,1,2
    1 IL = 0
      SUB1 = CFN(1)*XL(1)
      COOR(1) = XL(1)
    2 IF (ABS(CSN(2))-1.0E-04) 3,3,4
    3 IM = 0
      SUB1 = SUB1+CFN(2)*XL(2)
      COOR(2) = XL(2)
    4 IF (ABS(CSN(3))-1.0E-04) 5,5,6
    5 IN = 0
      SUB1 = SUB1+CFN(3)*XL(3)
      COOR(3) = XL(3)
    6 D = CFN(1)*X1+CFN(2)*Y1+CFN(3)*Z1
      IF (IL+IM+IN-2) 10,30,50
   10 IF (IL) 12,11,12
   11 IF (IM) 14,13,14
   12 IP = 1
      GO TO 15
   14 IP = 2
      GO TO 15
   13 IP = 3
   15 COOR(IP) = (D-SUB1)/CFN(IP)
```

```
      GO TO 90
   30 IF (IL) 32,31,32
   31 IP1 = 2
      IP2 = 3
      GO TO 35
   32 IF (IM) 34,33,34
   33 IP1 = 1
      IP2 = 3
      GO TO 35
   34 IP1 = 1
      IP2 = 2
   35 SLOPE = CSN(IP1)/CSN(IP2)
      COOR(IP2) = (D-SUB1+CFN(IP1)*SLOPE*XL(IP2)-CFN(IP1)*XL(IP1))/
     1            (CFN(IP1)*SLOPE+CFN(IP2))
      COOR(IP1) = SLOPE*(COOR(IP2)-XL(IP2))+XL(IP1)
      GO TO 90
   50 COEFX1 = 1./CSN(1)
      COEFY1 = -1./CSN(2)
      D1 = XL(1)/CSN(1)-XL(2)/CSN(2)
      COEFX2 = 1./CSN(1)
      COEFZ2 = -1./CSN(3)
      D2 = XL(1)/CSN(1)-XL(3)/CSN(3)
      CALL SOL (CFN(1),CFN(2),CFN(3),D,COEFX1,COEFY1,0.,D1,COEFX2,0.,
     1 COEFZ2,D2,COOR(1),COOR(2),COOR(3))
   90 COOR1 = COOR(1)
      COOR2 = COOR(2)
      COOR3 = COOR(3)
      RETURN
      END


      SUBROUTINE ADAMS(N,START,FINAL,H,PRINT,ICOUNT,RELB,ABSB,ISKIP,
     1                 XO,XP,PAR,DDERIV)
C
C     SUBROUTINE ADAMS SOLVES A SYSTEM OF *N* FIRST ORDER DIFFERENTIAL
C     EQUATIONS BY MEANS OF A FOURTH ORDER ADAMS PREDICTOR/CORRECTOR
C     METHOD. THE STARTING SOLUTION IS BY RUNGE-KUTTA METHOD.
C     AUTOMATIC ERROR CONTROL IS OPTIONAL.
C
      DIMENSION X(50,5),VK(50,4),F(50,5),E(50)
      DIMENSION XP(1),XO(1),PAR(1)
C
      IBOOL = 0
      IF (PRINT) 20,10,20
 10   IF (ICOUNT) 20,31,20
C
 20   CONTINUE
C20   WRITE (6,400) ID,N
      IBOOL = 1
C400  FORMAT (17HOPROBLEM NUMBER  I10,5X12HSOLUTION OF
C    1  I3,5X35HFIRST ORDER DIFFERENTIAL EQUATIONS.)
C
C     SETUP INITIAL VALUES
C
      DO 30 I=1,N
      X(I,1) = XO(I)
 30   CONTINUE
 31   CONTINUE
      IF (ICOUNT) 40,35,40
```

```
 35    ICOUNT = 9999
 40    ITEMP = 0
       BOUND = START+PRINT
       T = START
       IF (ISKIP) 45,50,45
 45    IA = 2
       IB = 4
       GO TO 2222
 50    RLTEST = 14.2*RELB
       ABTEST = 14.2*ABSB
       FACTOR = RELB/ABSB
       BLB = RLTEST/200.0
       H = 2.0*H
C
C      RUNGE-KUTTA STARTING METHOD
C
 1111 IA = 2
       IB = 2
C
 2222 DO 90 J=IA,IB
       CALL DDERIV (T,X(1,J-1),F(1,J-1),PAR)
       DO 60 I=1,N
       VK(I,1) = H*F(I,J-1)
       X(I,J) = X(I,J-1)+.5*VK(I,1)
 60    CONTINUE
       TTEMP = T+.5*H
C
       CALL DDERIV (TTEMP,X(1,J),F(1,J),PAR)
       DO 70 I=1,N
       VK(I,2) = H*F(I,J)
       X(I,J) = X(I,J-1)+.5*VK(I,2)
 70    CONTINUE
C
       CALL DDERIV (TTEMP,X(1,J),F(1,J),PAR)
       DO 80 I=1,N
       VK(I,3) = H*F(I,J)
       X(I,J) = X(I,J-1)+VK(I,3)
 80    CONTINUE
       T = T+H
C
       CALL DDERIV (T,X(1,J),F(1,J),PAR)
       DO 85 I=1,N
       VK(I,4) = H*F(I,J)
       X(I,J) = X(I,J-1)+.16666667*(VK(I,1)+2.0*(VK(I,2)+
      1  VK(I,3))+VK(I,4))
 85    CONTINUE
 90    CONTINUE
C
       IF (IB-2) 150,3333,150
 3333 DO 100 I=1,N
       XP(I) = X(I,2)
 100   CONTINUE
C
C      XP(I)=DOUBLE INTERVAL RESULT TO BE USED IN ERROR
C      ANALYSIS
C
       T = T-H
       H = .5*H
C
```

```
      IF (IBOOL) 120,125,120
  120 CONTINUE
 C120 WRITE (6,410) N
 C410 FORMAT (34HOIN THE FOLLOWING CALCULATIONS N =E14.8)
  125 IF (H-.0000001) 130,130,140
  130 WRITE (6,420)
  420 FORMAT (1H0,10(1H*),////
     1  49HOEQUATIONS CAN NOT BE SOLVED FURTHER WITHIN GIVEN
     2  14H ERRCR BOUNDS.)
      RETURN
C
  140 IB = 3
      GO TO 2222
C
  150 IF (IB-3) 200,160,200
C
C     IS ACCURACY CRITERION MET
C
  160 J = 3
 4444 DO 190 I=1,N
      E(I)=ABS(XP(I)-X(I,J))
      IF(E(I)-ABS(X(I,J)*RLTEST))170,175,175
  170 E(I)=E(I)/ABS(X(I,J))
      GO TO 190
  175 IF (E(I)-ABTEST) 180,185,185
  180 E(I) = E(I)*FACTOR
      GO TO 190
C
  185 T =T-H
      IF (J-5) 3333,187,3333
  187 DO 188 K=1,N
  188 X(K,1) = X(K,4)
      GO TO 1111
  190 CONTINUE
C
      IF (J-5)195,6666,195
  195 IA = 4
      IB = 4
      GO TO 2222
C
C     SHOULD ANY OF THE STARTING VALUES BE PRINTED OUT
C
  200 T = T-3.0*H
      DO 250 J=2,4
      T = T+H
      ITEMP = ITEMP+1
      IF (PRINT) 210,230,210
  210 IF (T-BOUND) 230,220,220
  220 BOUND = BOUND+PRINT
 9999 CONTINUE
C9999 WRITE (6,430) T,(I,X(I,J),I=1,N)
C430  FORMAT (4HOT =E14.8/ 5( 2H X,I2,1H=1PE12.5))
      ITEMP = 0
C
  230 IF (ITEMP-ICOUNT) 240,9999,240
  240 IF (T-(FINAL-H/10.0)) 250,999,999
  250 CONTINUE
C
C     BEGIN ADAMS METHOD
```

```
C
 5555 CALL DDERIV (T,X(1,4),F(1,4),PAR)
      DO 260 I=1,N
      XP(I) = X(I,4)+.041666667*H*(55.0*F(I,4)-59.0*F(I,3)
     1  +37.0*F(I,2)-9.0*F(I,1))
  260 CONTINUE
C
      T = T+H
      CALL DDERIV (T,XP(1),F(1,5),PAR)
      DO 270 I=1,N
      X(I,5) = X(I,4)+.041666667*H*(9.0*F(I,5)+19.0*F(I,4)-
     1  5.0*F(I,3)+F(I,2))
  270 CONTINUE
C
      IF (ISKIP) 6666,280,6666
  280 J = 5
      GO TO 4444
C
 6666 IF (T-(FINAL-H/10.0)) 295,290,290
  290 J = 5
      GO TO 999
C
  295 DO 300 I=1,N
      X(I,4) = X(I,5)
      DO 300 J=2,5
      F(I,J-1) = F(I,J)
  300 CONTINUE
C
      ITEMP = ITEMP+1
C
C     TEST WHETHER COMPUTED VALUES SHOULD BE PPINTED
C
      IF (PRINT) 310,330,310
  310 IF (T-(BOUND-H/10.0))330,320,320
  320 BOUND = BOUND+PRINT
 7777 J = 4
C     WRITE (6,430) T,(I,X(I,J),I=1,N)
      ITEMP = 0
C
  330 IF (ITEMP-ICOUNT) 340,7777,340
  340 IF (ISKIP) 5555,350,5555
C
C     TEST WHETHER INTERVAL CAN BE DOUBLED
C
  350 DO 355 I=1,N
      IF (E(I)-BLB) 355,355,5555
  355 CONTINUE
C
      IF (PRINT) 358,380,358
  358 D1 = PRINT/(2.0*H)
      D1I=ABS(FLOAT(IFIX(D1))-D1)
      IF (D1I-.1) 362,362,360
  360 IF (D1I-.9) 5555,362,362
  362 D2 = (BOUND-T)/(2.0*H)
      D2I=ABS(FLOAT(IFIX(D2))-D2)
      IF (D2I-.1) 380,380,365
  365 IF (D2I-.9) 5555,380,380
  380 DO 382 I=1,N
      X(I,1) = X(I,4)
```

```
 382  CONTINUE
      H = 4.0*H
      GO TO 1111
C
 999  CONTINUE
C999  WRITE (6,440)
C440  FORMAT (20HOFINAL T AND XP(I)...)
      DO 385 I=1,N
      XP(I) = X(I,J)
 385  CONTINUE
      FINAL = T
C     WRITE (6,430) T,(I,X(I,J),I=1,N)
      RETURN
      END


      SUBROUTINE CFCAL(ALFQ,BETQ,GETQ,PHI,PSI,CF)
C
C     COMPUTES DIRECTION COSINES FOR THE LOCAL COORDINATE SYSTEM, X IN
C     DIRECTION OF FREESTREAM,Y NORMAL TO FREESTREAM AND INITIAL JET
C     DIRECTION, Z IS XCROSSY
C
      DIMENSION CF(3,3)
C
      CXJ = SIN(PHI)*COS(PSI)
      CYJ = COS(PHI)
      CZJ = SIN(PHI)*SIN(PSI)
      CF(1,1) = ALFQ
      CF(1,2) = BETQ
      CF(1,3) = GETQ
      CALL XPROD (CXJ,CYJ,CZJ,CF(1,1),CF(1,2),CF(1,3),CF(2,1),CF(2,2),
     1 CF(2,3))
      CALL XPROD (CF(1,1),CF(1,2),CF(1,3),CF(2,1),CF(2,2),CF(2,3),
     1 CF(3,1),CF(3,2),CF(3,3))
      RETURN
      END


      SUBROUTINE ROTATE (A,B,C,CF,S,T,U,L)
C
C     L=0 ROTATES A,B,C INTO S,T,U,(FIXED COORDINATES TO ROTATED)
C     L=1 ROTATES S,T,U INTO A,B,C,(ROTATED COORDINATES TO FIXED)
C
      DIMENSION CF(3,3),D(3),V(3)
C
      IF (L) 1,1,2
    1 D(1) = A
      D(2) = B
      D(3) = C
      GO TO 3
    2 D(1) = S
      D(2) = T
      D(3) = U
    3 CONTINUE
      DO 4 I=1,3
    4 V(I) = 0.
      DO 5 I=1,3
      DO 5 J=1,3
      IF (L) 9,9,10
```

```
    9 M = I
      N = J
      GO TO 5
   10 M = J
      N = I
    5 V(I) = V(I)+D(J)*CF(M,N)
      IF (L) 6,6,7
    6 S = V(1)
      T = V(2)
      U = V(3)
      GO TO 8
    7 A = V(1)
      B = V(2)
      C = V(3)
    8 CONTINUE
      RETURN
      END

      SUBROUTINE XPROD (ALF1,BET1,GET1,ALF2,BET2,GET2,ALF3,BET3,GET3)
C
C     COMPUTES CROSS PRODUCT OF TWO VECTORS,RETURNS A UNIT VECTOR
C
      ALF3 = BET1*GET2-BET2*GET1
      BET3 = ALF2*GET1-ALF1*GET2
      GET3 = ALF1*BET2-ALF2*BET1
      DENOM = SQRT(ALF3*ALF3+BET3*BET3+GET3*GET3)
      ALF3 = ALF3/DENOM
      BET3 = BET3/DENOM
      GET3 = GET3/DENOM
      RETURN
      END

      SUBROUTINE SOL (A11,A12,A13,AK1,A21,A22,A23,AK2,A31,A32,A33,AK3,
     1 X1,X2,X3)
C
C     SOLVES A SET OF THREE EQUATIONS BY METHOD OF DETERMINANTS
C
      DELTA = A11*(A22*A33-A23*A32)+A21*(A32*A13-A12*A33)
     1        +A31*(A12*A23-A13*A22)
      X1 = (AK1*(A22*A33-A23*A32)+AK2*(A32*A13-A12*A33)
     1      +AK3*(A12*A23-A13*A22))/DELTA
      X2 = (A11*(AK2*A33-A23*AK3)+A21*(AK3*A13-AK1*A33)
     1      +A31*(AK1*A23-A13*AK2))/DELTA
      X3 = (A11*(A22*AK3-AK2*A32)+A21*(A32*AK1-A12*AK3)
     1      +A31*(A12*AK2-AK1*A22))/DELTA
      RETURN
      END
```

```
      PROGRAM MAPFN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
      DIMENSION NCOR(20),X(100),Y(100),XCOR(20),YCOR(20),DALPHA(20),
     1B(50),C(50,50),ALPHA(100),S(100),BETA(20),EXPON(20),OMEGA(100),
     2R(100),OMEGAA(11),SA(11),EPS1(11),RA(11),A(50,50),D(50),VEL(100),
     3PHI(100),DUMMY(20,2)
C
      COMMON NPT,NSYM,NTERM,KORN,NCOR,RC,DALPHA,PHI,DUMMY,ALPHA,S
C
 1    READ (5,5) NPT,KORN,NTERM,NSYM
 5    FORMAT(20I3)
      IF (EOF(5)) 500,6
    6 READ (5,10) (X(I),I=1,NPT)
      READ (5,10) (Y(I),I=1,NPT)
 10   FORMAT(8F9.5)
      READ (5,10) DX
      DO 12 I=1,NPT
 12   X(I)=X(I)+DX
      IF (NSYM)500,15,20
 15   X(NPT+1)=X(NPT-1)
      Y(NPT+1)=-Y(NPT-1)
      GO TO 25
 20   X(NPT+1)=X(2)
      Y(NPT+1)=Y(2)
 25   IF (KORN) 500,55,30
 30   READ (5,5) (NCOR(I),I=1,KORN)
      DO 35 I=1,KORN
 35   READ (5,10) XCOR(I),YCOR(I),DALPHA(I)
      DO 36 I=1,KORN
 36   XCOR(I)=XCOR(I)+DX
      KOR1=KORN
      DO 50 I=1,KOR1
      EXPON(I)=-DALPHA(I)/(3.141593+DALPHA(I))
      IF (NSYM) 500,40,50
 40   IF (YCOR(I)) 45,50,45
 45   KORN=KORN+1
      NCOR(KORN)=0
      YCOR(KORN)=-YCOR(I)
      XCOR(KORN)=XCOR(I)
      EXPON(KORN)=EXPON(I)
 50   CONTINUE
 55   ALPHA(1)=1.570796
      NC=1
      KB=0
      IF (NSYM) 500,65,60
 60   READ (5,10) ALPHA(1)
 65   IF (KORN) 500,90,70
 70   IF (NCOR(1)-1) 80,75,80
 75   ALPHA(1)=ALPHA(1)+DALPHA(1)/2.
      BETA(1)=ALPHA(1)
      NC=2
      KB=1
      IF (NC-KORN) 80,80,90
 80   DO 85 I=NC,KORN
 85   BETA(I)=ATAN2((YCOR(I)-Y(1)),(XCOR(I)-X(1)))-3.141593
 90   S(1)=0.
      I=1
      OMEGA(1)=ATAN2(Y(1),X(1))
      R(1)=SQRT(Y(1)**2+X(1)**2)
```

```
      NCOL=NTERM*(NSYM+1)
      DO 95 I=1,NCOL
      B(I)=0.
      DO 95 J=1,NCOL
95    C(I,J)=0.
      EPS1(11)=ALPHA(1)-OMEGA(1)-1.570796
      IF (KCRN) 500,110,100
100   DO 105 I=1,KORN
105   EPS1(11)=EPS1(11)+EXPON(I)*(BETA(I)-OMEGA(1))
110   DO 230 I=2,NPT
      I1=I-1
      KA=KB
      KB=C
      EPS1(1)=EPS1(11)
      OMEGAA(1)=OMEGA(I1)
      RA(1)=R(I1)
      SA(1)=0.
      IJ=I-12
      SN=SIN(ALPHA(I1))
      CS=COS(ALPHA(I1))
      U1=(X(I)-X(I1))*CS+(Y(I)-Y(I1))*SN
      C12=U1**2
      C11=C12*U1
      V1=(Y(I)-Y(I1))*CS-(X(I)-X(I1))*SN
      IF (IJ-1) 500,115,120
115   U2=(X(I+1)-X(I1))*CS+(Y(I+1)-Y(I1))*SN
      V2=(Y(I+1)-Y(I1))*CS-(X(I+1)-X(I1))*SN
      GO TO 125
120   U2=(X(I1-1)-X(I1))*CS+(Y(I1-1)-Y(I1))*SN
      V2=(Y(I1-1)-Y(I1))*CS-(X(I1-1)-X(I1))*SN
125   C22=U2**2
      C21=C22*U2
      DEN=C11*C22-C12*C21
      AA=(V1*C22-V2*C12)/DEN
      BB=(V2*C11-V1*C21)/DEN
      U=C.
      DU=U1/10.
      C3=C.
      XB=X(I1)
      YB=Y(I1)
      DO 175 J=2,11
      C2=C3
      U=U+DU
      XA=XB
      YA=YB
      V=(AA*U+BB)*U**2
      XB=X(I1)+U*CS-V*SN
      YB=Y(I1)+U*SN+V*CS
      RA(J)=SQRT(XB**2+YB**2)
      TN=(YB*XA-XB*YA)/(XA*XB+YA*YB)
      OMEGAA(J)=OMEGAA(J-1)+ATAN(TN)
      C3=(3.*AA*U+2.*BB)*U
      DALP=ATAN(C3)
      EPS1(J)=ALPHA(I1)+DALP-OMEGAA(J)-1.570796
      SA(J)=SA(J-1)+DU*SQRT(1.+.25*(C2+C3)**2)
      IF (KCRN) 500,175,130
130   IF (J-11) 155,135,500
135   IF (IJ-1) 500,155,140
140   DO 150 K=1,KCRN
```

```
      IF (I-NCOR(K)) 150,145,150
145   KB=K
      GO TO 155
150   CONTINUE
155   DO 170 K=1,KORN
      IF (K-KA) 160,157,160
157   BETA(K)=ALPHA(I1)+ATAN(V/U)
      GO TO 170
160   IF (K-KB) 165,162,165
162   BETA(K)=ALPHA(I1)+DALP-3.141593
      GO TO 170
165   ANUM=(YB-YA)*(XA-XCOR(K))-(XB-XA)*(YA-YCOR(K))
      DEN=(XB-XCOR(K))*(XA-XCOR(K))+(YB-YCOR(K))*(YA-YCOR(K))
      BETA(K)=BETA(K)+ATAN(ANUM/DEN)
170   EPS1(J)=EPS1(J)+EXPON(K)*(BETA(K)-OMEGAA(J))
175   CONTINUE
      R(I)=RA(11)
      OMEGA(I)=OMEGAA(11)
      S(I)=S(I1)+SA(11)
      ALPHA(I)=ALPHA(I1)+DALP
      IF (IJ-1) 500,185,180
180   IF (NSYM) 182,182,181
181   IF (I-NPT) 182,185,500
182   BETA(KB)=ALPHA(I)+DALPHA(KB)
      ALPHA(I)=BETA(KB)
185   I2=I
      IF (KORN) 500,205,190
190   DO 200 K=1,KORN
      IF (I+1-NCOR(K)) 200,195,200
195   I2=I-1
      GO TO 205
200   CONTINUE
      IF (NSYM) 205,205,201
201   IF (I+1-NPT) 205,202,205
202   IF (NCOR(1)-1) 205,203,205
203   I2=I-1
205   CONTINUE
      DO 230 J=2,11
      DS=SA(J)-SA(J-1)
      RK1=1.
      RK2=1.
      DO 230 K=1,NTERM
      AK=K
      OMK1=AK*OMEGAA(J-1)
      OMK2=AK*OMEGAA(J)
      RK1=RK1*RA(J-1)
      RK2=RK2*RA(J)
      SKR1=SIN(OMK1)/RK1
      SKR2=SIN(OMK2)/RK2
      B(K)=B(K)+.5*(EPS1(J)*SKR2+EPS1(J-1)*SKR1)*DS
      RL1=RK1
      RL2=RK2
      DO 210 L=K,NTERM
      AL=L
      SLR1=SIN(AL*OMEGAA(J-1))/RL1
      SLR2=SIN(AL*OMEGAA(J))/RL2
      RL1=RL1*RA(J-1)
      RL2=RL2*RA(J)
210   C(K,L)=C(K,L)+.5*(SKR2*SLR2+SKR1*SLR1)*DS
```

```
      IF (NSYM) 500,230,215
215   K1=NTERM+K
      CKR1=COS(CMK1)/RK1
      CKR2=COS(CMK2)/RK2
      B(K1)=B(K1)-.5*(EPS1(J)*CKR2+EPS1(J-1)*CKR1)*DS
      RL1=1.
      RL2=1.
      DO 225 L=1,NTERM
      AL=L
      L1=NTERM+L
      RL1=RL1*RA(J-1)
      RL2=RL2*RA(J)
      CLR1=COS(AL*OMEGAA(J-1))/RL1
      CLR2=COS(AL*OMEGAA(J))/RL2
      C(K,L1)=C(K,L1)-.5*(SKR2*CLR2+SKR1*CLR1)*DS
      IF (L-K) 225,220,220
220   C(K1,L1)=C(K1,L1)+.5*(CKR2*CLR2+CKR1*CLR1)*DS
225   CONTINUE
230   CONTINUE
      DO 235 I=2,NCOL
      I1=I-1
      DO 235 J=1,I1
235   C(I,J)=C(J,I)
      CALL MATINV(C,NCOL,A)
      DO 240 I=1,NCOL
      D(I)=0.
      DO 240 J=1,NCOL
240   D(I)=D(I)+A(I,J)*B(J)
      KA=0
      PHI(1)=0.
      PHIA=0.
      IF (KCRN) 500,255,245
245   IF (NCOR(1)-1) 255,250,255
250   VEL(1)=0.
      VEL2=0.
      KA=1
      KB=1
      GO TO 282
255   VEL2=1./R(1)
      IF (KORN) 500,270,260
260   DEN=X(1)**2+Y(1)**2
      DO 265 I=1,KORN
      AMP=((1.-(XCOR(I)*X(1)+YCOR(I)*Y(1))/DEN)**2+
     1     ((XCOR(I)*Y(1)-YCOR(I)*X(1))/DEN)**2)**(EXPON(I)/2.)
265   VEL2=VEL2*AMP
270   EXPN=0.
      RJ=1.
      DO 280 J=1,NTERM
      AJ=J
      RJ=RJ*R(1)
      EXPN=EXPN+D(J)*COS(AJ*OMEGA(1))/RJ
      IF (NSYM) 500,280,275
275   J1=NTERM+J
      EXPN=EXPN+D(J1)*SIN(AJ*OMEGA(1))/RJ
280   CONTINUE
      VEL2=VEL2*[illegible]XPN)
      VEL(1)=V[illegible]
282   I2=1
      DO 4[illegible] I=[illegible]
```

```
      I1=I-1
      IJ=I-I2
      SN=SIN(ALPHA(I1))
      CS=COS(ALPHA(I1))
      U1=(X(I)-X(I1))*CS+(Y(I)-Y(I1))*SN
      C12=U1**2
      C11=C12*U/
      V1=(Y(I)-Y(I1))*CS-(X(I)-X(I1))*SN
      IF (IJ-1) 500,285,290
  285 U2=(X(I+1)-X(I1))*CS+(Y(I+1)-Y(I1))*SN
      V2=(Y(I+1)-Y(I1))*CS-(X(I+1)-X(I1))*SN
      GO TO 295
  290 U2=(X(I1-1)-X(I1))*CS+(Y(I1-1)-Y(I1))*SN
      V2=(Y(I1-1)-Y(I1))*CS-(X(I1-1)-X(I1))*SN
  295 C22=U2**2
      C21=C22*U2
      DEN=C11*C22-C12*C21
      AA=(V1*C22-V2*C12)/DEN
      BB=(V2*C11-V1*C21)/DEN
      U=0.
      C3=0.
      DU=U1/10.
      DO 367 J=2,11
      C2=C3
      U=U+DU
      C3=(3.*AA*U+2.*BB)*U
      V=(AA*U+BB)*U**2
      DS=DU*SQRT(1.+.25*(C2+C3)**2)
      XB=X(I1)+U*CS-V*SN
      YB=Y(I1)+U*SN+V*CS
      VEL1=VEL2
      VEL2=1./SQRT(XB**2+YB**2)
      IF (KORN) 500,335,300
  300 IF (J-11) 325,305,500
  305 IF (IJ-1) 500,325,310
  310 DO 320 K=1,KORN
      IF (I-NCOR(K)) 320,315,320
  315 KA=-1
      KB=K
      GO TO 350
  320 CONTINUE
      IF (NSYM) 325,325,321
  321 IF (I-NPT) 325,322,325
  322 IF (NCOR(1)-1) 325,323,325
  323 KA=-1
      KB=1
      GO TO 350
  325 DEN=XB**2+YB**2
      DO 330 K=1,KORN
      AMP=((1.-(XCOR(K)*XB+YCOR(K)*YB)/DEN)**2+
     1     ((XCOR(K)*YB-YCOR(K)*XB)/DEN)**2)**(EXPON(K)/2.)
  330 VEL2=VEL2*AMP
  335 EXPN=0.
      RK=1.
      RU=SQRT(XB**2+YB**2)
      OMEG=QATAN(YB,XB)
      DO 345 K=1,NTERM
      AK=K
      RK=RK*RU
```

```
      EXPN=EXPN+D(K)*COS(AK*OMEG)/RK
      IF (NSYM) 500,345,340
  340 K1=NTERM+K
      EXPN=EXPN+D(K1)*SIN(AK*OMEG)/RK
  345 CONTINUE
      VEL2=VEL2*EXP(EXPN)
  350 IF (KA) 355,365,360
  355 PHIA=PHIA+VEL1*DS/(1.+EXPON(KB))
      KA=1
      GO TO 367
  360 PHIA=PHIA+VEL2*DS/(1.+EXPON(KB))
      KA=0
      GO TO 367
  365 PHIA=PHIA+.5*(VEL2+VEL1)*DS
  367 CONTINUE
      PHI(I)=PHIA
      VEL(I)=VEL2
      I2=I
      IF (KORN) 500,400,370
  370 DO 380 K=1,KORN
      IF (I+1-NCOR(K)) 380,375,380
  375 I2=I-1
      GO TO 400
  380 CONTINUE
      IF (NSYM) 400,400,381
  381 IF (I+1-NPT) 400,382,400
  382 IF (NCOR(1)-1) 400,383,400
  383 I2=I-1
  400 CONTINUE
      AF=NSYM+1
      PHIF=PHI(NPT)/(180.*AF)
      WRITE (6,402)
  402 FORMAT(43H1COMPUTATIONS FOR S AND ALPHA VERSUS THETA.)
      WRITE (6,405)
  405 FORMAT(6H0     X,12X1HY,12X1HR,12X1HS,12X1HV,10X5HALPHA,8X5HOMEGA,
     1       8X5HTHETA/1H )
      DO 410 I=1,NPT
      PHI(I)=PHI(I)/PHIF
      ALPHA(I)=57.29578*ALPHA(I)
      OMEGA(I)=57.29578*OMEGA(I)
  410 WRITE (6,415) X(I),Y(I),R(I),S(I),VEL(I),ALPHA(I),OMEGA(I),PHI(I)
  415 FORMAT(1H ,9E13.5)
      CALL MAPP1
      CALL MAPP5
      GO TO 1
  500 STOP
      END


      SUBROUTINE MAPP1
C
      DIMENSION ALPHA(100),THETA(100),S(100),NCOR(20),A(20,2),C(21,2),
     1DALPHA(20),SNN1(19),SNN2(19),CSN1(19),CSN2(19),TH(22),D(20,2)
C
      COMMON NPT,NSYM,NTERM,KORN,NCOR,RC,DALPHA,THETA,A,ALPHA,S
C
      DO 15 I=1,NPT
      THETA(I)=.01745329*THETA(I)
   15 ALPHA(I)=.01745329*ALPHA(I)
```

```
      IF (NSYM) 500,20,25
20    THETA(NPT+1)=6.283185-THETA(NPT-1)
      ALPHA(NPT+1)=9.424778-ALPHA(NPT-1)
      S(NPT+1)=2.*S(NPT)-S(NPT-1)
      GO TO 40
25    THETA(NPT+1)=6.283185+THETA(2)
      IF (NCOR(1)-1) 30,35,30
30    ALPHA(NPT+1)=6.283185+ALPHA(2)
      GO TO 38
35    ALPHA(NPT+1)=6.283185+ALPHA(2)-DALPHA(1)
38    S(NPT+1)=S(NPT)+S(2)
40    NTERM1=NTERM-1
      CS2=COS(ALPHA(1)-THETA(1))
      SN2=SIN(ALPHA(1)-THETA(1))
      DO 45 I=1,NTERM1
      AI=I
      ANG=ALPHA(1)+AI*THETA(1)
      CSN2(I)=COS(ANG)
45    SNN2(I)=SIN(ANG)
      DO 50 I=1,NTERM
      DO 50 J=1,2
50    A(I,J)=0.
      IT=0
      IA=0
      IF (KORN) 500,80,55
55    IF (NCOR(1)-1) 80,60,80
60    IT=1
      EXP1=3.141593/(3.141593+DALPHA(1))
      S0=S(1)
      TH0=THETA(1)
      A11=(S(2)-S(1))**EXP1
      A12=(S(2)-S(1))**2
      B1=THETA(2)-THETA(1)
      IF (NSYM) 500,65,70
65    A21=-(S(1)+S(2))**EXP1
      A22=(S(1)+S(2))**2
      B2=-THETA(2)-THETA(1)
      GO TO 75
70    A21=-(S(1)+S(NPT)-S(NPT-1))**EXP1
      A22=(S(1)+S(NPT)-S(NPT-1))**2
      B2=-THETA(1)-THETA(NPT)+THETA(NPT-1)
75    DEN=A11*A22-A12*A21
      C1=(A22*B1-A12*B2)/DEN
      C2=(A11*B2-A21*B1)/DEN
80    DO 200 I=2,NPT
      IF (IT) 500,90,85
85    IT=0
      GO TO 120
90    IF (KORN) 500,110,95
95    DO 105 J=1,KORN
      IF (NCOR(J)-I) 105,100,105
100   IT=1
      EXP1=3.141593/(3.141593+DALPHA(J))
      GO TO 115
105   CONTINUE
110   EXP1=1.
115   A11=(S(I+1)-S(I))**EXP1
      A12=(S(I+1)-S(I))**2
      B1=THETA(I+1)-THETA(I)
```

```
      A21=-(S(I)-S(I-1))**EXP1
      A22=(S(I)-S(I-1))**2
      B2=THETA(I-1)-THETA(I)
      S0=S(I)
      THO=THETA(I)
      DEN=A11*A22-A12*A21
      C1=(A22*B1-A12*B2)/DEN
      C2=(A11*B2-A21*B1)/DEN
120   IAB=0
      IF (IA) 500,130,125
125   IA=0
      IAB=1
      GO TO 160
130   IF (KORN) 500,150,135
135   DO 145 J=1,KORN
      IF (NCOR(J)-I-1) 145,140,145
140   IA=1
      AL2=ALPHA(I+1)-DALPHA(J)
      GO TO 155
145   CONTINUE
150   AL2=ALPHA(I+1)
155   S1=S(I)
      ALO=ALPHA(I)
      A11=S(I+1)-S(I)
      A12=A11**2
      B1=AL2-ALPHA(I)
      A21=S(I-1)-S(I)
      A22=A21**2
      B2=ALPHA(I-1)-ALPHA(I)
      DEN=A11*A22-A12*A21
      C3=(A22*B1-A12*B2)/DEN
      C4=(A11*B2-A21*B1)/DEN
160   AL2=ALPHA(I-1)
      TH2=THETA(I-1)
      SA=S(I-1)
      DS=(S(I)-S(I-1))/10.
      DO 165 J=2,11
      TH1=TH2
      SA=SA+DS
      TH2=THO+SIGN(C1,SA-S )*ABS(SA-SO)**EXP1+C2*(SA-SO)**2
      AL2=ALO+C3*(SA-S1)+C4*(SA-S1)**2
      SN1=SN2
      CS1=CS2
      ANG=AL2-TH2
      SN2=SIN(ANG)
      CS2=COS(ANG)
      A(1,1)=A(1,1)+(SN2+SN1)*DS/2.
      A(1,2)=A(1,2)+(CS2+CS1)*DS/2.
      K1=1
      DO 165 K=1,NTERM1
      K1=K1+1
      AK=K
      ANG=AL2+AK*TH2
      SNN1(K)=SNN2(K)
      CSN1(K)=CSN2(K)
      SNN2(K)=SIN(ANG)
      CSN2(K)=COS(ANG)
      A(K1,1)=A(K1,1)+(SNN2(K)+SNN1(K))*DS/2.
165   A(K1,2)=A(K1,2)-(CSN2(K)+CSN1(K))*DS/2.
```

```
      IF (IAB) 500,180,170
170   ANG=ALPHA(I+1)-THETA(I+1)
      CS2=COS(ANG)
      SN2=SIN(ANG)
      DO 175 K=1,NTERM1
      AK=K
      ANG=ALPHA(I+1)+AK*THETA(I+1)
      CSN2(K)=COS(ANG)
175   SNN2(K)=SIN(ANG)
180   CONTINUE
200   CONTINUE
      IF (NSYM) 500,215,225
215   RC=A(1,1)/3.141593
      A(1,1)=0.
      A(1,2)=0.
      PIRC=3.141593*RC
      DO 220 I=2,NTERM
      A(I,1)=A(I,1)/PIRC
220   A(I,2)=0.
      GO TO 235
225   RC=A(1,1)/6.283185
      A(1,1)=0.
      A(1,2)=0.
      PIRC=6.283185*RC
      DO 230 I=2,NTERM
      DO 230 J=1,2
230   A(I,J)=A(I,J)/PIRC
235   DO 240 I=1,NTERM
      DO 240 J=1,2
      D(I,J)=0.
240   C(I+1,J)=0.
      C(1,1)=1.
      C(1,2)=0.
      IF (KORN) 500,285,245
245   DO 280 I=1,KORN
      IF (NCOR(I)) 500,280,250
250   NSYM1=1
      IF (NSYM) 500,255,270
255   IF (NCOR(I)-1) 500,270,260
260   IF (NCOR(I)-NPT) 265,270,500
265   NSYM1=2
270   IA=NCOR(I)
      ANG=THETA(IA)
      SN=-SIN(ANG)
      CS=COS(ANG)
      DO 275 J=1,NSYM1
      SN=-SN
      EXPI=DALPHA(I)/3.141593
      COEFR=1.
      COEFI=0.
      DO 172 K=1,NTERM
      DO 172 L=1,2
172   C(K+1,L)=D(K,L)
      DO 275 K=1,NTERM
      AK=K
      COEF1=COEFR
      COEFR=-EXPI*(COEF1*CS-COEFI*SN)/AK
      COEFI=-EXPI*(COEFI*CS+COEF1*SN)/AK
      EXPI=EXPI-1.
```

```
      N1=NTERM+1
      NA=N1-K
      DO 275 N=K,NTERM
      N1=N1-1
      D(N1,1)=D(N1,1)+C(NA,1)*COEFR-C(NA,2)*COEFI
      D(N1,2)=D(N1,2)+C(NA,1)*COEFI+C(NA,2)*COEFR
  275 NA=NA-1
  280 CONTINUE
  285 A(1,1)=-D(1,1)
      A(1,2)=-D(1,2)
      DO 290 I=2,NTERM
      A(I,1)=A(I,1)-D(I,1)
      A(I,2)=A(I,2)-D(I,2)
      DO 290 J=2,I
      J1=I-J+1
      A(I,1)=A(I,1)-D(J-1,1)*A(J1,1)+D(J-1,2)*A(J1,2)
  290 A(I,2)=A(I,2)-D(J-1,1)*A(J1,2)-D(J-1,2)*A(J1,1)
      WRITE (6,295)
  295 FORMAT(42H1SECTION MAPPING BY NUMERICAL INTEGRATION./49H0
     1  X                    Y                    THETA)
      READ (5,305) X,Y,THO,THF,DTH
  305 FORMAT(5F6.2)
      DTH=.01745329*DTH
      THO=.01745329*THO
      THF=.01745329*THF
      NSEG=1
      TH(NSEG)=THO
      IF (KORN) 500,335,310
  310 DO 330 I=1,KORN
      IF (NCOR(I)) 500,330,315
  315 IA=NCOR(I)
      IF (THETA(IA)-THO) 330,500,320
  320 IF (THF-THETA(IA)) 335,500,325
  325 NSEG=NSEG+1
      TH(NSEG)=THETA(IA)
  330 CONTINUE
      IF (NSYM) 500,331,335
  331 DO 337 I=1,KORN
      IF (NCOR(I)-1) 337,337,332
  332 IF (NCOR(I)-NPT) 333,337,500
  333 IA=NCOR(I)
      THT=6.283185-THETA(IA)
      IF (THT-THO) 337,500,334
  334 IF (THF-THT) 335,500,336
  336 NSEG=NSEG+1
      TH(NSEG)=THT
  337 CONTINUE
  335 TH(NSEG+1)=THF
      TH2=THO
      DEL = 10.
      IF (NSEG-1) 500,350,340
  340 DO 345 I=1,NSEG
      DEL1=(TH(I+1)-TH(I))/3.
  345 DEL=AMIN1(DEL,DEL1)
      DEL=AMIN1(DEL,.0349066)
  350 DO 385 I=1,NSEG
      NPSEG=(TH(I+1)-TH(I))/DTH
      NPSEG=NPSEG+1
      PSEG=NPSEG
```

```
      AI=C.
      IF (I-1) 500,360,355
355   AI=AI+1.
360   IF (I-NSEG) 365,370,500
365   AI=AI+1.
370   DT=(TH(I+1)-TH(I)-AI*DEL)/PSEG
      DO 385 J=1,NPSEG
      TH1=TH2
      TH2=TH1+DT
      CALL MAPP(TH1,TH2,1.,1.,X,Y,1)
      WRITE (6,390) X,Y,TH2
      IF (J-NPSEG) 385,375,500
375   IF (I-NSEG) 380,385,500
380   TH1=TH2
      TH2=TH1+2.*DEL
      CALL MAPP(TH1,TH2,1.,1.,X,Y,3)
      WRITE (6,390) X,Y,TH2
385   CONTINUE
390   FORMAT(1H ,3E17.5)
500   RETURN
      END


      SUBROUTINE MAPP(TH1,TH2,R1,R2,X,Y,KODE)
C
      DIMENSION NCOR(20),DALPHA(20),THETA(100),A(20,2),RA(11),THA(11),
     1AML(11),ANU(11)
C
      COMMON NPT,NSYM,NTERM,KORN,NCOR,RC,DALPHA,THETA,A
C
      IF (KODE-2) 5,20,35
5     DO 10 I=1,11
10    RA(I)=R1
      DTH=(TH2-TH1)/10.
      THA(1)=TH1
      DO 15 I=1,10
15    THA(I+1)=THA(I)+DTH
      GO TO 45
20    DO 25 I=1,11
25    THA(I)=TH1
      RA(1)=R1
      DR=(R2-R1)/10.
      DO 30 I=1,10
30    RA(I+1)=RA(I)+DR
      GO TO 45
35    C=2.*SIN((TH2-TH1)/4.)
      DEL=(TH1-TH2-6.283185)/4.
      DDEL=-DEL/5.
      THC=(TH1+TH2)/2.
      RA(1)=1.
      RA(11)=1.
      THA(1)=TH1
      THA(11)=TH2
      DO 40 I=2,10
      DEL=DEL+DDEL
      CD=COS(DEL)
      SD=SIN(DEL)
      RA(I)=SQRT(1.+C*(C+2.*CD))
      ANG=C*SD/(1.+C*CD)
```

```
40    THA(I)=THO+ATAN(ANG)
45    DO 100 K=1,11
      AMU(K)=RC
      ANU(K)=0.
      IF (KCRN) 500,90,50
50    DO 85 I=1,KCRN
      IF (NCOR(I)) 500,85,55
55    NSYM1=1
      IF (NSYM) 500,60,75
60    IF (NCOR(I)-1) 500,75,65
65    IF (NCOR(I)-NPT) 70,75,500
70    NSYM1=2
75    IA=NCOR(I)
      AI=-1.
      EXPN=DALPHA(I)/6.283185
      DO 80 J=1,NSYM1
      AI=-AI
      DANG=AI*THETA(IA)-THA(K)
      SN=SIN(DANG)
      CS=COS(DANG)
      SN=-SN/RA(K)
      CS=1.-CS/RA(K)
      R=(CS**2+SN**2)**EXPN
      ANG=2.*EXPN*ATAN(SN/CS)
      SN=R*SIN(ANG)
      CS=R*COS(ANG)
      AM1=AMU(K)
      AMU(K)=AM1*CS-ANU(K)*SN
80    ANU(K)=AM1*SN+ANU(K)*CS
85    CONTINUE
90    RE=RA(K)*COS(THA(K))
      AIM=RA(K)*SIN(THA(K))
      RN=1./RA(K)
      AN=-1.
      DO 95 I=1,NTERM
      RN=RN*RA(K)
      AN=AN+1.
      ANGN=AN*THA(K)
      CS=COS(ANGN)/RN
      SN=SIN(ANGN)/RN
      RE=RE+A(I,1)*CS+A(I,2)*SN
95    AIM=AIM+A(I,2)*CS-A(I,1)*SN
      AM1=AMU(K)
      AMU(K)=AM1*RE-ANU(K)*AIM
100   ANU(K)=AM1*AIM+ANU(K)*RE
      IF (KODE-2) 105,115,105
105   DO 110 I=1,10
      DTH=(THA(I+1)-THA(I))/2.
      X=X-(ANU(I+1)+ANU(I))*DTH
110   Y=Y+(AMU(I+1)+AMU(I))*DTH
115   IF (KODE-1) 500,500,120
120   DO 125 I=1,10
      DR=(RA(I+1)-RA(I))/2.
      X=X+(AMU(I+1)/RA(I+1)+AMU(I)/RA(I))*DR
125   Y=Y+(ANU(I+1)/RA(I+1)+ANU(I)/RA(I))*DR
500   RETURN
      END
```

```
      SUBROUTINE MAPPS
C
      DIMENSION NCOR(20),DALPHA(20),THETA(100),A(20,2),ALPHA(100),
     1S(100),B(21,2)
C
      COMMON NPT,NSYM,NTERM,KCRN,NCOR,RC,DALPHA,THETA,A,ALPHA,S
C
      IF (NSYM) 500,5,12
 5    DO 10 I=1,NTERM
 10   A(I,2)=0.
 12   IF (KCRN) 500,60,15
 15   DO 55 I=1,KCRN
      IF (NCOR(I)) 500,55,20
 20   J1=1
      IF (NSYM) 500,25,40
 25   IF (NCOR(I)-1) 30,40,30
 30   IF (NCOR(I)-NPT) 35,40,35
 35   J1=2
 40   THET=THETA(NCOR(I))
      CS=COS(THET)
      SN=-SIN(THET)
      DO 50 J=1,J1
      SN=-SN
      B(1,1)=1.
      B(1,2)=0.
      DO 45 K=1,NTERM
      DO 45 L=1,2
 45   B(K+1,L)=A(K,L)
      RE=1.
      AM=0.
      COEF=1.
      DO 50 K=1,NTERM
      AK=K
      COEF=-COEF*(DALPHA(I)/3.141593-AK+1.)/AK
      RE1=RE
      RE=RE1*CS-AM*SN
      AM=RE1*SN+AM*CS
      DO 50 L=K,NTERM
      LK=L-K+1
      A(L,1)=A(L,1)+COEF*(RE*B(LK,1)-AM*B(LK,2))
 50   A(L,2)=A(L,2)+COEF*(RE*B(LK,2)+AM*B(LK,1))
 55   CONTINUE
 60   WRITE (6,65) RC
 65   FORMAT(27H1RADIUS OF MAPPING CIRCLE =,E13.5)
      NTERM1=NTERM-1
      RN=RC
      DO 70 I=1,NTERM1
      I1=I+1
      RN=RN*RC
      AI=I
      A(I,1)=-A(I1,1)*RN/AI
 70   A(I,2)=-A(I1,2)*RN/AI
      WRITE (6,71)
 71   FORMAT(28H0REAL PARTS OF COEFFICIENTS.)
      WRITE (6,75) (A(I,1),I=1,NTERM1)
      IF (NSYM) 500,76,74
 76   DO 73 I=1,NTERM1
 73   A(I,2)=0.
 74   WRITE (6,72)
```

```
 72   FORMAT(33H0IMAGINARY PARTS OF COEFFICIENTS.)
      WRITE (6,75) (A(I,2),I=1,NTERM1)
 75   FORMAT(1H0,7E13.5)
      READ (5,95) N,DTH,THO
 95   FORMAT(I3,2F6.2)
      DTH=.01745329*DTH
      THO=.01745329*THO
      TH=THO-DTH
      WRITE (6,96)
 96   FORMAT(41H1MAPPING OF SECTION WITH CORNERS REMOVED.)
      WRITE (6,100)
 100  FORMAT(20H        X            Y)
      DO 110 I=1,N
      TH=TH+DTH
      CS=COS(TH)
      SN=SIN(TH)
      X=RC*CS
      Y=RC*SN
      RN=1.
      DO 105 J=1,NTERM1
      AJ=J
      THN=AJ*TH
      CS=COS(THN)
      SN=SIN(THN)
      RN=RN*RC
      X=X+(A(J,1)*CS+A(J,2)*SN)/RN
 105  Y=Y+(A(J,2)*CS-A(J,1)*SN)/RN
 110  WRITE (6,115) X,Y
 115  FORMAT(1H ,2F12.5)
 500  RETURN
      END


      SUBROUTINE MATINV(A,N,B)
C
      DIMENSION A(50,50),B(50,50),C(50,50)
C
      DO 1 I=1,N
      DO 1 J=1,N
 1    B(I,J)=0.0
      DO    I=1,N
      B(I,I)=1.0
      DO 2 J=1,N
 2    C(J,I)=A(J,I)
      DO 6 I=1,N
      IF(C(I,I))24,50,24
   50 DO 21 IZ=I,N
      IF(C(IZ,I))22,21,22
   21 CONTINUE
      WRITE(6,100)
  100 FORMAT(19H0MATRIX IS SINGULAR)
      GO TO 7
   22 DO 23 M=1,N
      C(I,M)=C(I,M)+C(IZ,M)
   23 B(I,M)=B(I,M)+B(IZ,M)
   24 TC=C(I,I)
      DO 3 J=1,N
      C(I,J)=C(I,J)/TC
 3    B(I,J)=B(I,J)/TC
```

```
      DC 6 K=1,N
      IF(K-I)4,6,4
 4    T=C(K,I)
      DC 5 L=1,N
      C(K,L)=C(K,L)-T*C(I,L)
 5    B(K,L)=B(K,L)-T*B(I,L)
 6    CCNTINUE
      RETURN
    7 STCP
      END


      FUNCTICN CATAN(SN,CS)
C
      IF (SN) 45,20,5
 5    IF (CS) 10,15,60
 10   QATAN=3.141593+ATAN(SN/CS)
      GC TC 100
 15   QATAN=1.570796
      GC TC 100
 20   IF (CS) 25,30,40
 25   QATAN=3.141593
      GC TC 100
 30   WRITE (6,35)
 35   FCRMAT(30H OANGLE UNDEFINED. SET TO ZERO.)
 40   QATAN=0.
      GC TC 100
 45   IF (CS) 10,50,55
 50   QATAN=4.712389
      GC TC 100
 55   QATAN=6.283185+ATAN(SN/CS)
      GC TC 100
 60   CATAN=ATAN(SN/CS)
 100  RETURN
      END
```

```
      PROGRAM TRANS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE2)
C
C     *** MAIN PROGRAM FOR COMBINED STRIP METHOD AND 3DMODIFICATION ***
C     IGEOM = 1 FOR WING, IGEOM = 2 FOR BODY
C     MODIN = 0 SKIP 3D MODIFICATION, MODIN = 1 PERFORM 3D MODIFICATION
C     JSTOP=NUMBER OF ITERATIONS,  IDIS=NUMBER OF LAYERS IN 3D MODIFICAT
C     JPOWER=0,  POWER EFFECT;    JPOWER=1,  POWER 3%.
C     IRECT=0,  RECTANGULAR WING;     IRECT=1,  NONRECTANGULAR WING OR BO
C     IFORCE=0,NO FORCE/MOMENT COMPUTED  IFORCE=1,FORCE/MOMENT COMPUTED
C
      DIMENSION UJHK(16,40),VJHK(16,40),WJHK(16,40),APART(20),RBHK(7,16)
     1          ,AHK(12,16),VXX(1,16,40),VYY(1.16,40),DW(30)
      DIMENSION BHK(12,16)
      DIMENSION CP(16,40),ORDX(16)
      DIMENSION X(4,18,40),Y(4,18,40),Z(4,18,40),SI(40,20),CS(40,20)
      DIMENSION DX(16,40),DY(16,40),DZ(16,40),AC(150)
C
      COMMON/BLKHK1/NSTA,N,NFOUR,NSYM,ITAPE
      COMMON/BLKHK2/UJHK,VJHK,WJHK,APART,RBHK,Z
      COMMON/BLKHK3/SI,CS
      COMMON/BLKHK4/ORDX
      COMMON/BLKHK5/UJ,ALPHA,BETA
      COMMON/BLKHK6/CP
      COMMON/BLKHK7/X
      COMMON/BLKHK8/Y
      COMMON/BLKH9/DZ
      COMMON/BLKH10/DX
      COMMON/BLKH11/DY
      COMMON/BLKH13 /VXX
      COMMON/BLKH14 /VYY
      COMMON/BLKH15 /NDOWN,IREPET
      COMMON/BLKH16 /DW
C
      ITAPE = 2
  101 CONTINUE
      READ (5,501) IGEOM,MODIN,JSTOP,IDIS,JPOWER,IRECT,IFORCE
      IF (EOF(5)) 999,102
  102 CONTINUE
      DO 1113 K=1,16
      DO 1113 J=1,40
      VXX(1,K,J)=0.0
 1113 VYY(1,K,J)=0.0
      DO 1114 K=1,30
 1114 DW(K)=0.0
      NDOWN=0
      IREPET=1
      IF (IGEOM-2) 1,2,201
    1 WRITE (6,601)
  601 FORMAT (1H1,52X,16HWING COMPUTATION/51X,20H********************)
      WRITE (6,610)
      IF (MODIN) 60,60,61
   60 WRITE (6,611)
  611 FORMAT (1H0,15X,22H1. SEGMENT METHOD ONLY)
      GO TO 3
    2 WRITE (6,602)
  602 FORMAT (1H1,52X,16HBODY COMPUTATION/51X,20H********************)
      WRITE (6,610)
```

```
61C  FCRPAT (1HO,///10X,34HOPTIONS SPECIFIED FOR THIS RUN ARE/)
     IF (PCDIN) 60,60,61
  61 WRITE (6,612) JSTOP
612  FCRPAT (1HO,15X,36H1. THREE DIMENSIONAL MODIFICATION CF,I3,3X,
    19HITERATION)
  3  RFAC (5,502) NSTA,N,NFOUR,NSYM,MTHET,UJ,ALPHA,BETA
     IF (JPCWER)  62,64,65
 62  WRITE (6,613)
613  FCRPAT (1HO,15X,26H2. POWER OFF CCNFIGURATION)
     GC TC 70
 64  WRITE (6,614)
614  FORMAT (1HO,15X,20H2. POWER EFFECT ONLY)
     GC TC 7C
 65  WRITE (6,615)
615  FCRPAT (1HO,15X,25H2. POWER ON CCNFIGURATION)
 70  WRITE (6,616)
616  FORMAT (1HO,///53X,14H**INPUT DATA**)
     WRITE (6,617)NSTA,N,NFOUR,NSYM,MTHET,IRECT,IFORCE,UJ,ALPHA,BETA
617  FCRPAT (1HO,5X,5HNSTA=,I3,3X,2HN=,I3,3X,6HNFOUR=,I3,3X,5HNSYM=,I2,
    1          3X,6HMTHET=,I3,3X,6HIRECT=,I3,3X,7HIFORCE=,I3,/6X,3HUJ=,F7
    2          .3,3X,6HALPHA=,F8.3,3X,5HBETA=,F8.3)
     DO 20 I=1,NSTA
     RFAD (5,503) APART(I),RBHK(1,I),DRDX(I)
     WRITE (6,628) APART(I),RBHK(1,I),DRDX(I)
628  FCRPAT (1HO,2X,8HSTATION=,F12.6,3X,7HRADIUS=,F12.6,3X,6HDERIV=,
    1          F12.6)
     BEAB= ABS(BETA)
     IF (NSYM) 202,5,6
   5 IF (BEAB-0.001) 1131,1131,1132
1131 NTHET= MTHET+1
     GC TC 1133
1132 NTHET= MTHET
1133 RFAD (5,505) (AHK(J,I),J=1,N)
     WRITE (6,618) I,(AHK(J,I), J=1,N)
618  FCRPAT(1HO,2X,36HGEOMETRY COEFFICIENT *A* FOR STATION,I3/(6E15.6))
     GC TC 8
  6  NTHET = MTHET
     READ  (5,505) (AHK(J,I),BHK(J,I), J=1,N)
     WRITE (6,619) I,(AHK(J,I),BHK(J,I),J=1,N)
619  FCRPAT (1HO,2X,41HGECMETRY COEFFICIENTS *A*,*B* FCR STATION,I3/
    1          (6E15.6))
  8  IF (JPOWER) 12,11,11
 11  READ (5,505) (UJHK(I,J),J=1,NTHET)
     RFAD (5,505) (VJHK(I,J),J=1,NTHET)
     READ (5,505) (WJHK(I,J),J=1,NTHET)
     WRITE (6,620) I,(UJHK(I,J),J=1,NTHET)
     WRITE (6,621) I,(VJHK(I,J),J=1,NTHET)
     WRITE (6,622) I,(WJHK(I,J),J=1,NTHET)
620  FORMAT (1HO,2X,33HVELOCITY COMPONENT *U* AT STATION,I3/(6E15.5))
621  FORMAT (1HO,2X,33HVELOCITY COMPONENT *V* AT STATION,I3/(6E15.5))
622  FCRPAT (1HO,2X,33HVELOCITY COMPONENT *W* AT STATION,I3/(6E15.5))
     GC TC 20
 12  DC 15 J=1,NTHET
     UJHK(I,J) = 0.
     VJHK(I,J) = 0.
 15  WJHK(I,J) = 0.
 20  CCNTINUE
     DC 900 K=1,NSTA
     RBHK(2,K)= 1.5*RBHK(1,K)
```

```
      DO 905 I=3,IDIS
      AI=I-2
      AI=AI*RBHK(1,K)
  905 RBHK(I,K)=RBHK(2,K)*AI
  900 CONTINUE
      IF (NFOUR-N) 800,805,805
  800 NFCL= N
      GO TO 801
  805 NFCL= NFOUR
  801 IF (NSYM) 202,841,842
  841 IF (BEAB-0.001) 837,837,842
  837 MT=2*MTHET
      GO TO 843
  842 MT=MTHET
  843 AN=6.283185/FLOAT(MT)
      DO 835 I=1,MT
      AI=I-1
      AC(I)=AI*AN
      ANG=AN*AI
      SI(I,1)=SIN(ANG)
      CS(I,1)=COS(ANG)
      SI(I,2)=2.0*SI(I,1)*CS(I,1)
  835 CS(I,2)=1.0-2.0*SI(I,1)**2
      NTEST1=NFCU/2
      NTEST2=(NFOU+1)/2
      IF (NTEST1-NTEST2) 1220,1221,1220
 1220 NCOF1= NFCU-1
      NCOF2= NFCU
      GO TO 1222
 1221 NCOF1= NFCU
      NCOF2= NFCU-1
 1222 DO 840 J=4,NCOF1,2
      DO 840 I=1,MT
      SI(I,J)=SI(I,2)*CS(I,J-2)+CS(I,2)*SI(I,J-2)
  840 CS(I,J)=CS(I,2)*CS(I,J-2)-SI(I,2)*SI(I,J-2)
      DO 845 J=3,NCOF2,2
      DO 845 I=1,MT
      SI(I,J)=SI(I,1)*CS(I,J-1)+CS(I,1)*SI(I,J-1)
  845 CS(I,J)=CS(I,1)*CS(I,J-1)-SI(I,1)*SI(I,J-1)
      IF (IGEOM-2) 810,815,201
  810 IF (IRECT) 201,846,847
  846 NNN=1
      GO TO 848
  847 NNN=NSTA
  848 DO 850 K=1,NNN
      DO 850 I=1,IDIS
      DO 850 J=1,MTHET
      AA=RBHK(I,K)*(AHK(1,K)*CS(J,1) +BHK(1,K)*SI(J,1)) +AHK(2,K)
      BB=RBHK(I,K)*(AHK(1,K)*SI(J,1) +BHK(1,K)*CS(J,1)) +BHK(2,K)
      REV=1.0
      DO 855 NS=3,N
      LL=NS-2
      REV=REV/RBHK(I,K)
      AA=AA +REV*(AHK(NS,K)*CS(J,LL) +BHK(NS,K)*SI(J,LL))
  855 BB=BB +REV*(-AHK(NS,K)*SI(J,LL) +BHK(NS,K)*CS(J,LL))
      X(I,K,J)=AA
  850 Z(I,K,J)=BB
      DO 860 K=1,NNN
      DO 860 J=1,MTHET
```

```
      AD=RBHK(1,K)*(-AHK(1,K)*SI(J,1) +BHK(1,K)*CS(J,1))
      BD=RBHK(1,K)*(AHK(1,K)*CS(J,1) -BHK(1,K)*SI(J,1))
      REV=1.0
      DO 865 ND=3,N
      CD=ND-2
      REV=REV/RBHK(1,K)
      AD=AD +REV*(-AHK(ND,K)*SI(J,ND-2) +BHK(ND,K)*CS(J,ND-2))*CD
  865 BD=BD -REV*( AHK(ND,K)*CS(J,ND-2) +BHK(ND,K)*SI(J,ND-2))*CD
      DX(K,J)=AD
  860 DZ(K,J)=BD
      IF (NNN.NE.1) GO TO 856
      DO 857 K=2,NSTA
      DO 857 I=1,IDIS
      DO 857 J=1,MTHET
      X(I,K,J)=X(I,1,J)
  857 Z(I,K,J)=Z(I,1,J)
      DO 858 K=2,NSTA
      DO 858 J=1,MTHET
      DX(K,J)=DX(1,J)
  858 DZ(K,J)=DZ(1,J)
  856 NSTA2=0
  710 NSTA1=NSTA2+1
      NSTA2=MINO(NSTA,NSTA2+4)
      WRITE (6,702)
      WRITE (6,703) (APART(I),I=NSTA1,NSTA2)
      WRITE (6,704)
      ATHET= 360.0/FLOAT(MTHET)
      DO 715 J=1,MTHET
      TEJ=J-1
      THEE=TEJ*ATHET
  715 WRITE (6,705) THEE,(X(1,I,J),Z(1,I,J),I=NSTA1,NSTA2)
      IF (NSTA-NSTA2)1041,1041,710
  815 IF (BEAB-0.001) 920,920,925
  920 ITH= MTHET+1
      GO TO 930
  925 ITH= 1+MTHET/2
  930 DO 935 K=1,NSTA
      DO 935 I=1,IDIS
      DO 935 J=1,ITH
      AA=-RBHK(I,K)*CS(J,1) -AHK(2,K)
      BB= RBHK(I,K)*SI(J,1)
      REV=1.0
      DO 940 NS=3,N
      LL=NS-2
      REV=REV/RBHK(I,K)
      AA=AA -REV*AHK(NS,K)*CS(J,LL)
  940 BB=BB -REV*AHK(NS,K)*SI(J,LL)
      Y(I,K,J)=BB
  935 Z(I,K,J)=AA
      DO 945 K=1,NSTA
      DO 945 J=1,ITH
      AD= RBHK(1,K)*SI(J,1)
      BD= RBHK(1,K)*CS(J,1)
      REV=1.0
      DO 950 ND=3,N
      CD=ND-2
      LL=ND-2
      REV=REV/RBHK(1,K)
      AD=AD +REV*AHK(ND,K)*SI(J,LL)*CD
```

```
 950 BD=BD -REV*AHK(ND,K)*CS(J,LL)*CD
     DY(K,J)=BD
 945 DZ(K,J)=AD
     ITHM=ITH-1
     DO 955 K=1,NSTA
     DO 955 I=1,IDIS
     DO 955 J=2,ITHM
     LL=2*ITHM+2-J
     Y(I,K,LL)=-Y(I,K,J)
 955 Z(I,K,LL)= Z(I,K,J)
     DO 956 K=1,NSTA
     DO 956 J=2,ITHM
     LL=2*ITHM+2-J
     DY(  K,LL)= DY(  K,J)
 956 DZ(  K,LL)=-DZ(  K,J)
     NSTA2=0
 720 NSTA1=NSTA2+1
     NSTA2=MINO(NSTA,NSTA2+4)
     WRITE (6,706)
     WRITE (6,707) (APART(I),I=NSTA1,NSTA2)
     WRITE (6,708)
     MTHET2=2*(ITH-1)
     ATHET= 360.0/FLOAT(MTHET2)
     DO 725 J=1,MTHET2
     TEJ=J-1
     THEE= TEJ*ATHET
 725 WRITE (6,705) THEE,(Y(1,I,J),Z(1,I,J),I=NSTA1,NSTA2)
     IF (NSTA-NSTA2) 1041,1041,720
1041 KOUNT=0
     IF (NSYM) 202,1115,1120
1115 IF (BEAB-0.001) 1125,1125,1120
1125 NTH= 2*MTHET
     GO TO 50
1120 NTH= MTHET
 50  CALL STRIP (IGEOM,KOUNT,MTHET,JPOWER,AC)
     IF (MODIN) 90,90,22
 22  IF (IGEOM-2) 23,24,201
 23  IF (KOUNT-1) 30,40,90
 30  KOUNT = KOUNT+1
     NTH = MTHET
     READ (5,501) NBOOL,MEXIT
     GO TO 1015
1001 KOUNT=1
     IREPET=IREPET+1
1015 CALL WMOD3 (NTH,IDIS,NBOOL,MEXIT)
     GO TO 50
 40  KOUNT = KOUNT+1
     IF (IREPET -1) 1020,1020,1025
1020 READ (5,501) MOD
1025 CALL DNWASH (NTH,MOD)
     GO TO 50
  24 IF (IREPET-1) 1024,1024,1030
1030 IF (IREPET-JSTOP) 1035,1035,1002
1024 IF (KOUNT) 38,38,90
 38  KOUNT = KOUNT+1
     READ (5,501) NJET
     READ (5,504)        APART(NSTA+1)
1035 CALL BMOD3 (NTH,IDIS,NJET)
     IREPET=IREPET+1
```

```
      GO TO 50
   90 IF (IREPET-JSTOP) 1001,1002,1003
 1002 IF (IGEOM-2) 1305,1310,201
 1305 WRITE (6,731) IDIS,NBOOL,MEXIT,MOD
      GO TO 1003
 1310 WRITE (6,732) IDIS,NJET,APART(NSTA+1)
 1003 IF (IFORCE.EQ.0) GO TO 101
      IF (IGEOM-2) 91,92,201
   91 READ (5,506) NDJ,DIAM,XCG,ZCG,CHORD
      WRITE (6,734) NDJ,DIAM,XCG,ZCG,CHORD
      CALL FMWING (NTH,IRECT,NDJ,DIAM,XCG,ZCG,CHORD)
      WRITE  (6,660)
 660  FORMAT (1H0,/45X,29H***END OF WING COMPUTATION***)
      GO TO 101
   92 READ (5,506) NDJ,DIAM,XCG,CHORD
      READ (5,504) YTIP,ZTIP,APART(NSTA+1),YTAIL,ZTAIL
      ZERO = 0.
      WRITE  (6,733) NDJ,DIAM,XCG,CHORD,ZERO,YTIP,ZTIP,APART(NSTA+1),
     1 YTAIL,ZTAIL
      CALL FMBODY (NTH,YTIP,ZTIP,YTAIL,ZTAIL,NDJ,DIAM,XCG,CHORD)
      WRITE (6,661)
 661  FORMAT (1H0,/45X,29H***END OF BODY COMPUTATION***)
      GO TO 101
 201  WRITE (6,603)
 603  FORMAT (1H0,31H**ERROR IN GEOMETRY INDICATOR**)
      STOP
 202  WRITE (6,604)
 604  FORMAT (1H0,31H**ERROR IN SYMMETRY INDICATOR**)
 999  STOP
 501  FORMAT (12I6)
 502  FORMAT (5I3,4F7.3)
 503  FORMAT (3F12.6)
 504  FORMAT (6F12.6)
 505  FORMAT (6E12.5)
  506 FORMAT (I3,4F12.6)
  702 FORMAT (1H1,42X23HTABLE FOR WING GEOMETRY)
  703 FORMAT (1H0,6X,4(10X,2HY=,F6.2,10X))
  704 FORMAT (1H ,6H THETA,4(5X4HX(I)10X4HZ(I)5X))
  705 FORMAT (1H ,F6.2,8E14.5)
  706 FORMAT (1H1,38X27HTABLE FOR FUSELAGE GEOMETRY)
  707 FORMAT (1H0,6X,4(10X,2HX=,F6.2,10X))
  708 FORMAT (1H ,6H THETA,4(5X4HY(I)10X4HZ(I)5X))
  731 FORMAT (1H1,54HPARAMETERS USED IN 3D MODIFICATION OF WING COMPUTAT
     1ION,3X5HIDIS=,I3,1X6HNBOOL=,I3,1X6HMEXIT=,I3,1X4HMOD=,I3)
  732 FORMAT (1H1,58HPARAMETERS USED IN 3D MODIFICATION OF FUSELAGE COMP
     1UTATION,3X5HIDIS=,I3,1X5HNJET=,I3,1X19HLENGTH OF FUSELAGE=,F8.3)
  733 FORMAT (1H0,47HPARAMETERS USED IN FORCE AND MOMENT COMPUTATION,
     1I3,16HJET OF DIAMETER=,F8.3,6H  XCG=,F8.3,19H  REFERENCE LENGTH=,
     2F8.3,/ 5X23HCOORDINATES OF NOSE  X=,F8.3,4H  Y=,F8.3,4H  Z=,F8.3,
     325H  COORDINATES OF TAIL  X=,F8.3,4H  Y=,F8.3,4H  Z=,F8.3)
  734 FORMAT (1H0,38HPARAMETERS IN FORCE/MOMENT COMPUTATION,I3,16HJET OF
     1 DIAMETER  F8.3,6H  XCG=,F8.3,6H  ZCG=,F8.3,19H  REFERENCE LENGTH=
     2,F8.3)
      END

      SUBROUTINE THEC(NM,MA,NU,AC,PT,A,B)

      DIMENSION NU(1),AC(1),PT(1),A(1),B(1)
```

```
      DIMENSION CZ(37),SZ(37),CA(7),SA(7),VAR(10),ARG(10),CON(10)
C
      MZ=MA+1
      MAE=MA+4
      DO 59 M=MZ,MAE
      IF(AC(M)-AC(M-1)) 58,59,59
   58 AC(M)=AC(M)+6.283184
   59 CONTINUE
      DO 110 N=1,NM
      FN=FLOAT(N)
      DEL=0.17453288/FN
      ANGO=AC(1)-DEL
      DO 20 I=1,18
      ANGO=ANGO+DEL
      CZ(I)=COS(FN*ANGO)
      SZ(I)=SIN(FN*ANGO)
      CZ(I+18)=-CZ(I)
   20 SZ(I+18)=-SZ(I)
      CZ(37)=CZ(1)
      SZ(37)=SZ(1)
      A(N)=0.0
      B(N)=0.0
      MC=-3
      ARG(4)=AC(1)
      CA(7)=CZ(1)*PT(1)
      SA(7)=SZ(1)*PT(1)
      ANG=AC(1)
      DO 100 J=1,N
      DO 90 K=1,6
      CA(1)=CA(7)
      SA(1)=SA(7)
      LC=(K-1)*6
      DO 80 L=2,7
      LV=LC+L
      ANG=ANG+DEL
      IF(ARG(4)-ANG)50,70,70
   50 MC=MC+3
      IF(AC(MC+4)-ANG) 50,55,55
   55 DO 60 M=1,4
      MV=MC+M
      ARG(M)=AC(MV)
      VAR(M)=PT(MV)
   60 CONTINUE
      CALL SVCC(VAR,ARG,CON,4)
   70 ZA=SVIN(ANG,ARG,CON,4)
      CA(L)=ZA*CZ(LV)
      SA(L)=ZA*SZ(LV)
   80 CONTINUE
      B(N)=B(N)+SA(1)+SA(3)+SA(5)+SA(7)+5.0*(SA(2)+SA(6))+6.0*SA(4)
      A(N)=A(N)+CA(1)+CA(3)+CA(5)+CA(7)+5.0*(CA(2)+CA(6))+6.0*CA(4)
   90 CONTINUE
  100 CONTINUE
      HDE=DEL*0.0954930
      A(N)=A(N)*HDE
      B(N)=B(N)*HDE
  110 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SVCO(VAR,ARG,CON,NUM)
C
      DIMENSION ARG(1),VAR(1),CON(1)
C
      DEM=ARG(NUM)-ARG(1)
      DO 15 J=1,NUM
      DEN=1.
      DO 10 I=1,NUM
      DEL=(ARG(J)-ARG(I))/DEM
      IF (ABS(DEL)-0.000001) 5,5,10
 5    DEL=1.
 10   DEN=DEN*DEL
 15   CON(J)=VAR(J)/DEN
      RETURN
      END

      FUNCTION SVIN(ARK,ARG,CON,NUM)
C
      DIMENSION ARG(1),CON(1)
      DIMENSION DEL(10)
C
      DEM=ARG(NUM)-ARG(1)
      SUMC=0.
      PROA=1.
      JP=1
      DO 20 J=1,NUM
      DEL(J)=(ARK-ARG(J))/DEM
 5    IF (ABS(DEL(J))-.000001) 10,10,20
 10   SUMC=CON(J)
      JP=2
      DEL(J)=1.
 20   CONTINUE
      DO 30 J=1,NUM
      GO TO (25,30),JP
 25   SUMC=SUMC+CON(J)/DEL(J)
 30   PROA= PROA*DEL(J)
      SVIN=PROA*SUMC
      RETURN
      END

      SUBROUTINE STRIP (IGEOM,IPRINT,MTHET,JPOWER,AC)
C
      DIMENSION UJHK(16,40),VJHK(16,40),WJHK(16,40),X(20),RBHK(7,16),
     1   Z(4,18,40),VXX(1,16,40),VYY(1,16,40),DW(30)
      DIMENSION CP(16,40),ORDX(16)
      DIMENSION AC(1)
      DIMENSION VX(40),VY(40),VZ(40)
C
      COMMON/BLKHK1/NSTA,N,NFOUR,NSYM,ITAPE
      COMMON/BLKHK2/UJHK,VJHK,WJHK,X  ,RBHK,Z
      COMMON/BLKHK4/ORDX
      COMMON/BLKHK5/UJ,ALPHA,BETA
      COMMON/BLKHK6/CP
      COMMON/BLKH13 /VXX
      COMMON/BLKH14 /VYY
      COMMON/BLKH15 /NDOWN,IREPET
```

```
      COMMON/BLKH16 /DW
C
      BEAB=ABS(BETA)
      MTT=MTHET+1
      ALPC= 0.0174533*ALPHA
      BETR= 0.0174533*BETA
      COAF= COS(ALPC)
      SIAF= SIN(ALPC)
      COBE= COS(BETR)
      SIBE= SIN(BETR)
      Q= COAF*COBE
      R= SIBE
      S= SIAF*COBE
      U0= 1.0
      IF (JPOWER) 4,2,4
    2 U0=0.0
    4 DVX= U0*Q
      DVY= U0*R
      DVZ= U0*S
      REWIND ITAPE
      DO 920 I=1,NSTA
      IF (NSYM) 200,25,35
   25 IF (BEAB-0.001) 26,26,35
   26 NTHET= MTHET+1
      GO TO 40
   35 NTHET = MTHET
   40 DO 41 J=1,NTHET
      VX(J) = UJHK(I,J)
      VY(J) = VJHK(I,J)
   41 VZ(J) = WJHK(I,J)
C  THE SIGN CONVENTION FOR Z-VELOCITY COMPTS THROUGHOUT HERE IS POSITIVE
C          IN POSITIVE Z-DIR POINTED UPWARD
      DO 50 J=1,NTHET
      VX(J)=VX(J)+DVX
      VY(J)=VY(J)+DVY
   50 VZ(J)=VZ(J)+DVZ
      IF (NSYM) 200,55,65
   55 IF (BEAB-0.001) 303,303,65
  303 I1= 2*NTHET-1
      DO 60 J=2,NTHET
      I1=I1-1
      VX(I1)=VX(J)
      VY(I1)=-VY(J)
   60 VZ(I1)=VZ(J)
      NTHET=2*MTHET
   65 IF (IGEOM-2) 66,67,67
   66 CONTINUE
      CALL VLWING (NTHET,I,VX,VY,VZ,AC)
      GO TO 68
   67 CALL VLBODY (NTHET,I,VX,VY,VZ,AC)
   68 CONTINUE
      IF (JPOWER) 901,900,901
  900 DO 905 J=1,NTHET
  905 CP(I,J)=-2.0*(VX(J)*Q+VY(J)*R+VZ(J)*S)-VX(J)**2-VY(J)**2-VZ(J)**2
      GO TO 921
  901 DO 70 J=1,NTHET
   70 CP(I,J)=-2.0*U0*(VX(J)-U0)-(VX(J)-U0)**2-VY(J)**2-VZ(J)**2
  921 IF (IGEOM-2) 906,907,907
  906 IF (NDOWN-0) 300,920,300
```

```
 300 DC 908 J=1,NTHET
     VJHK(I,J)= VJHK(I,J)-VYY(1,I,J)
     WJHK(I,J)= WJHK(I,J)-DW(I)/3.0
 908 VYY(1,I,J)=0.0
     DW(I)=0.0
     GC TC 920
 907 DC 909 J=1,MTT
     UJHK(I,J)= UJHK(I,J)-VXX(1,I,J)
 909 VXX(1,I,J)=0.0
 920 CCNTINUE
     IPRIN=IPRINT+1
 203 FORMAT (47H1PRESSURE COEFFICIENTS AT WING, SEGMENT METHOD.)
 204 FCRMAT (71H1PRESSURE CCEFFICIENTS AT WING AFTER RESIDUAL SOURCE/SI
    1NK MODIFICATICN.)
 205 FCRMAT (72H1PRESSURE COEFFICIENTS AT WING, END OF THREE DIMENSIONA
    1L MCDIFICATION OF,I3,3X,10HITERATION.)
 206 FCRMAT (51H1PRESSURE CCEFFICIENTS AT FUSELAGE, SEGMENT METHOD.)
 207 FCRMAT (67H1PRESSURE COEFFICIENTS AT FUSELAGE, THREE DIMENSIONAL M
    1CDIFICATICN OF,I3,3X,10HITERATION.)
     NSTA2=0
80   NSTA1=NSTA2+1
     NSTA2=MINO(NSTA,NSTA2+7)
     IF (IGECM-2) 85,95,200
  85 GC TC (210,215,220),IPRIN
 210 WRITE (6,203)
     GC TC 225
 215 WRITE (6,204)
     GC TC 225
 220 WRITE (6,205)IREPET
 225 WRITE (6,110) (X(I),I=NSTA1,NSTA2)
110  FORMAT (1H0,6X,7(4X2HY=,F6.2,4X))
     WRITE (6,115) (RBHK(1,I),I=NSTA1,NSTA2)
     WRITE (6,121) (DRDX(I),I=NSTA1,NSTA2)
121  FCRMAT (1H ,6H THETA,7(1X,5HDRDY=,F6.2,4X))
     GO TC 105
  95 GC TC (230,235),IPRIN
 230 WRITE (6,206)
     GC TC 240
 235 IRETT= IREPET-1
     WRITE (6,207)IRETT
 240 WRITE (6,111) (X(I),I=NSTA1,NSTA2)
111  FCRMAT (1H0,6X,7(4X2HX=,F6.2,4X))
     WRITE (6,115) (RBHK(1,I),I=NSTA1,NSTA2)
     WRITE (6,120) (DRDX(I),I=NSTA1,NSTA2)
115  FCRMAT (1H ,6X,7(3X3HRB=,F6.2,4X))
120  FORMAT (1H ,6H THETA,7(1X5HDRDX=,F6.2,4X))
105  CCNTINUE
     WRITE (6,125)
125  FCRMAT (1H )
     ATHET=360./FLCAT(NTHET)
     DC 130 J=1,NTHET
     AJ=J-1
     THET=AJ*ATHET
130  WRITE (6,135) THET,(CP(I,J),I=NSTA1,NSTA2)
135  FCRMAT (1H ,F6.2,7E16.5)
     IF (NSTA-NSTA2) 201,201,80
200  STCP
201  RETURN
     END
```

```
      SUBROUTINE VLBODY (MTHET,K,VX,VY,VZ,AC)
C
      DIMENSION DRDX(16)
      DIMENSION UJHK(16,40),VJHK(16,40),WJHK(16,40),XL(20),RB(7,16),
     1    Z(4,18,40),Y(4,18,40),DY(16,40),DZ(16,40),DPSI(40)
      DIMENSION SI(40,20),CS(40,20)
      DIMENSION VX(1),VY(1),VZ(1),AC(1)
      DIMENSION AF(30),BF(30)
      DIMENSION NU(150),PT(150)
C
      COMMON/BLKHK1/NSTA,N,NFOUR,NSYM,ITAPE
      COMMON/BLKHK2/UJHK,VJHK,WJHK,XL,RB,Z
      COMMON/BLKHK3/SI,CS
      COMMON/BLKHK4/DRDX
      COMMON/BLKHK8/Y
      COMMON/BLKH9/DZ
      COMMON/BLKH11/DY
C
      DO 50 I=1,MTHET
      DS2=SQRT(DY(K,I)**2+DZ(K,I)**2)
      DVY=VX(I)*DRDX(K)*DZ(K,I)/DS2
      DVZ=-VX(I)*DRDX(K)*DY(K,I)/DS2
   50 DPSI(I)=(VY(I)-DVY)*DZ(K,I) -(VZ(I)-DVZ)*DY(K,I)
      PT(1)=0.0
      J=MTHET+1
      AC(J)=6.2831853
      DPSI(J)=DPSI(1)
      CALL INTEG (4,J,DPSI,AC,PT)
      B0=0.1591549*PT(J)
      AJ=0.0
      CORR=PT(J)/FLOAT(MTHET)
      DO 65 I=2,J
      AJ=AJ+1.0
 65   PT(I)=PT(I)-AJ*CORR
      DO 70 I=2,4
      J=J+1
      AC(J)=AC(I)
 70   PT(J)=PT(I)
      DO 75 I=1,150
 75   NU(I)=I
      CALL THEO (NFOUR,MTHET,NU,AC,PT,AF,BF)
      WRITE (ITAPE) B0,(AF(I),BF(I),I=1,NFOUR)
      IF (K-NSTA) 77,76,76
  76  END FILE ITAPE
  77  DO 110 I=1,MTHET
      YCOMP=B0*CS(I,1)
      ZCOMP=-B0*SI(I,1)
      DO 105 J=1,NFOUR
      NANG=(I-1)*(J+1)+1
 80   IF (NANG) 85,85,90
 85   NANG=NANG+MTHET
      GO TO 80
 90   IF (NANG-MTHET) 100,100,95
 95   NANG=NANG-MTHET
      GO TO 90
 100  AJ=J
      YCOMP=YCOMP+AJ*(BF(J)*CS(NANG,1)-AF(J)*SI(NANG,1))
```

```
105 ZCOMP=ZCOMP-AJ*(AF(J)*CS(NANG,1)+BF(J)*SI(NANG,1))
    DRE=DZ(K,I)*CS(I,1)-DY(K,I)*SI(I,1)
    DIM=-DY(K,I)*CS(I,1)-DZ(K,I)*SI(I,1)
    DEN2=DRE**2+DIM**2
    V1=-(YCOMP*DRE+ZCOMP*DIM)/DEN2
    V2=(ZCOMP*DRE-YCOMP*DIM)/DEN2
    VY(I)=VY(I)+V1
110 VZ(I)=VZ(I)+V2
200 RETURN
    END


    SUBROUTINE VLWING (MTHET,K,VX,VY,VZ,AC)
C
    DIMENSION UJHK(16,40),VJHK(16,40),WJHK(16,40),XL(20),RB(7,16),
   1    DRDX(16)
    DIMENSION X(4,18,40),Z(4,18,40),DX(16,40),DZ(16,40),SI(40,20),
   1    CS(40,20),DPSI(40)
    DIMENSION VX(1),VY(1),VZ(1),AC(1)
    DIMENSION AF(30),BF(30)
    DIMENSION NU(150),PT(150)
C
    COMMON/BLKHK1/NSTA,N,NFOUR,NSYM,ITAPE
    COMMON/BLKHK2/UJHK,VJHK,WJHK,XL,RB,Z
    COMMON/BLKHK3/SI,CS
    COMMON/BLKHK4/DRDX
    COMMON/BLKHK7/X
    COMMON/BLKH9/DZ
    COMMON/BLKH10/DX
C
    IF (ABS(DRDX(K)).GT.0.01) GO TO 40
    DO 35 I=1,MTHET
 35 DPSI(I)=VX(I)*DZ(K,I)-VZ(I)*DX(K,I)
    GO TO 50
 40 DO 45 I=1,MTHET
    DS2=SQRT(DX(K,I)**2+DZ(K,I)**2)
    DVX=VY(I)*DRDX(K)*DZ(K,I)/DS2
    DVZ=-VY(I)*DRDX(K)*DX(K,I)/DS2
 45 DPSI(I)=(VX(I)-DVX)*DZ(K,I)-(VZ(I)-DVZ)*DX(K,I)
 50 PT(1)=0.0
    J=MTHET+1
    AC(J)=6.2831853
    DPSI(J)=DPSI(1)
    CALL INTEG (4,J,DPSI,AC,PT)
    B0=0.1591549*PT(J)
    AJ=0.0
    CORR=PT(J)/FLOAT(MTHET)
    DO 65 I=2,J
    AJ=AJ+1.0
 65 PT(I)=PT(I)-AJ*CORR
    DO 70 I=2,4
    J=J+1
    AC(J)=AC(I)
 70 PT(J)=PT(I)
    DO 75 I=1,150
 75 NU(I)=I
    CALL THEO (NFOUR,MTHET,NU,AC,PT,AF,BF)
    AO=0.0
    DO 76 I=1,NFOUR
```

```
      AI=I
 76   AC=AO+AI*AF(I)
      WRITE (ITAPE) AO,BO,(AF(I),BF(I),I=1,NFOUR)
      DO 110 I=1,NTHET
      XCOMP=BO*CS(I,1)+AO*SI(I,1)
      ZCOMP=-BO*SI(I,1)+AO*CS(I,1)
      DO 105 J=1,NFOUR
      NANG=(I-1)*(J+1)+1
 80   IF (NANG) 85,85,90
 85   NANG=NANG+NTHET
      GO TO 80
 90   IF (NANG-NTHET) 100,100,95
 95   NANG=NANG-NTHET
      GO TO 90
 100  AJ=J
      XCOMP=XCOMP +AJ*(BF(J)*CS(NANG,1)-AF(J)*SI(NANG,1))
  105 ZCOMP=ZCOMP -AJ*(AF(J)*CS(NANG,1)+BF(J)*SI(NANG,1))
      DRE=DZ(K,I)*CS(I,1)-DX(K,I)*SI(I,1)
      DIM=-DX(K,I)*CS(I,1)-DZ(K,I)*SI(I,1)
      DEN2=DRE**2+DIM**2
      V1=-(XCOMP*DRE+ZCOMP*DIM)/DEN2
      V2=(ZCOMP*DRE-XCOMP*DIM)/DEN2
      VX(I)=VX(I)+V1
 110  VZ(I)=VZ(I)+V2
 200  RETURN
      END


      SUBROUTINE INTEG(N,NX,FPR,X,FCN)
C
      DIMENSION CON(10),FPR(1),X(1),FCN(1)
C
      NI=10
      XNI=NI
      NIM2=NI-2
      DO 75 I=2,NX
      J=I-1
      IF (J-1) 1,1,5
 1    J0=1
      GO TO 20
 5    IF (NX-J-N+2) 70,10,15
 10   J0=NX-N+1
      GO TO 20
 15   IF (NX-I) 70,70,16
 16   IF (J-J0-N+2) 70,18,18
 18   J0=J-1
   20 CALL SVCO(FPR(J0),X(J0),CON,4)
   70 SUM=0.0
      DELX=(X(I)-X(J))/XNI
      DO 80 K=2,NIM2,2
      DX=K-1
      DX=DX/XNI
      XX=(1.0-DX)*X(J)+DX*X(I)
      YY=SVIN(XX,X(J0),CON,4)
      XX=XX+DELX
      YY2=SVIN(XX,X(J0),CON,4)
  80  SUM=SUM+4.0*YY+2.0*YY2
      XX=XX+DELX
      SUM=SUM+SVIN(X(J),X(J0),CON,4)+SVIN(X(I),X(J0),CON,4)
```

```
     1 +4.0*SVIA(XX,X(JO),CON,4)
       SUM=SUM*DELX/3.0
       FCN(I)=FCN(J)+SUM
    75 CONTINUE
       RETURN
       END


       SUBROUTINE DNWASH (NTHET,MOD)
C
       DIMENSION UJHK(16,40),VJHK(16,40),WJHK(16,40),Y(20),RBHK(7,16),
     1  Z(4,18,40)
       DIMENSION AOHK(16)
       DIMENSION SI(40,20),CS(40,20),NU(150),EC(150),A(50),B(50),GAMA(40)
     1  ,CX(40),CY(40),FA(40),W(30)
C
       COMMON/BLKHK1/NSTI,NDUM,NFOUR,NSYM,ITAPE
       COMMON/BLKHK2/UJHK,VJHK,WJHK,Y,RBHK,Z
       COMMON/BLKHK5/UJ,ALPHA,BETF
       COMMON/BLKH15 /NDOWN,IREPET
       COMMON/BLKH16 /W
C
       REWIND ITAPE
       DO 10 I=1,NSTI
   10  READ (ITAPE) AOHK(I)
       NDOWN=1
       BETA= ABS(BETF)
       IF (BETA-0.001) 400,400,405
   405 MC=NSTI
       ISP= (NSTI+1)/2
       NSTA= ISP-1
       DO 150 I=1,ISP
       CT= Y(I)/Y(1)
  150  CX(I) = ACOS(CT)
       DO 155 I=1,ISP
   155 CX(NSTI+1-I)= 3.14159-CX(I)
       DO 160 I=1,NSTI
   160 CY(NSTI+1-I)= AOHK(I)
       CY(1)=0.0
       CY(NSTI)=0.0
       GO TO 420
   400 NSTA=NSTI-1
       MC = 2*NSTA
       DO 2 I=2,NSTI
       INV = NSTI-I+1
       CX(INV) = Y(I)
     2 CY(INV) = AOHK(I)
       CY(1) = 0.
       DO 255 I=2,NSTA
       CT=CX(I)/CX(1)
  255  CX(I) = ACOS(CT)
       CX(1)=0.0
   415 DO 262 I=1,NSTA
       J=MC+2-I
       CX(J)= 3.14159-CX(I)
   262 CY(J)= CY(I)
       CX(NSTI)= 1.5708
       CY(NSTI)= AOHK(1)
       MC=MC+1
```

```
 420 IF (BETA-0.001) 421,421,422
 422 DO 271 J=2,18
     AI=J-1
     DUM=0.174533*AI
     DO 272 I=2,NSTI
     IF (CX(I)-DUM) 272,1120,1121
1120 GAMA(J)=CY(I)
     GO TO 271
1121 GAMA(J)=CY(I-1) +(CY(I)-CY(I-1))*(DUM-CX(I-1))/(CX(I)-CX(I-1))
     GO TO 271
 272 CONTINUE
 271 CONTINUE
     GAMA(1)=0.
     GAMA(19)=0.
     GO TO 423
 421 DO 265 J=2,9
     AI=J-1
     DUM=0.174533*AI
     DO 266 I=2,NSTI
     IF (CX(I)-DUM) 266,1180,1181
1180 GAMA(J)=CY(I)
     GO TO 265
1181 GAMA(J)=CY(I-1) +(CY(I)-CY(I-1))*(DUM-CX(I-1))/(CX(I)-CX(I-1))
     GO TO 265
 266 CONTINUE
 265 CONTINUE
     GAMA(1)=0.
     GAMA(10)=CY(NSTI)
     DO 275 I=1,9
     J=20-I
 275 GAMA(J)= GAMA(I)
 423 DO 355 I=2,18
     J=38-I
 355 GAMA(J)=-GAMA(I)
     MA=36
     DO 350 I=1,150
 350 NU(I)=I
     DO 360 I=1,36
     AI=I-1
 360 EC(I)=0.174533*AI
     EC(37)=6.283185
     GAMA(37)= GAMA(1)
     DO 361 I=2,4
     J= 36+I
     EC(J)=EC(I)
 361 GAMA(J)=GAMA(I)
     CALL THEC (NFCUR,MA,NU,EC,GAMA,A,B)
     DO 365 I=1,NFCUR
 365 FA(I)=B(I)
     MTHET=MOD
     NTEST1=NFCUR/2
     NTEST2=(NFCUR+1)/2
     IF(NTEST1-NTEST2) 1160,1161,1160
1160 NCCF1=NFCUR-1
     NCCF2=NFCUR
     GO TO 1162
1161 NCCF1=NFCUR
     NCCF2=NFCUR-1
1162 DO 55 I=1,MTHET
```

```
      IF (I-1) 110,105,110
  105 ANG=3.14159/2.0
      GOTO 115
  110 N=NSTA+2-I
      ANG=CX(N)
  115 SI(I,1)=SIN(ANG)
      CS(I,1)=COS(ANG)
      SI(I,2)=2.0*SI(I,1)*CS(I,1)
   55 CS(I,2)=1.0-2.0*SI(I,1)**2
      DO 60 J=4,NCOF1,2
      DO 60 I=1,MTHET
      SI(I,J)=SI(I,2)*CS(I,J-2)+CS(I,2)*SI(I,J-2)
   60 CS(I,J)=CS(I,2)*CS(I,J-2)-SI(I,2)*SI(I,J-2)
      DO 65 J=3,NCOF2,2
      DO 65 I=1,MTHET
      SI(I,J)=SI(I,1)*CS(I,J-1)+CS(I,1)*SI(I,J-1)
   65 CS(I,J)=CS(I,1)*CS(I,J-1)-SI(I,1)*SI(I,J-1)
      FACT = 2.*Y(NSTI)
      DO 300 K=1,MOD
      S=0.0
      DO 301 I=1,NFOUR
      AI=I
  301 S=S+FA(I)*SI(K,I)*AI
  300 W(K) = 3.1416*S/(FACT*SI(K,1))
      IF (BETA-0.001) 425,425,430
  430 DO 165 K=1,MOD
  165 W(ISP-1+K)= W(K)
      MR=ISP+1-MOD
      DO 166 K=1,MR
  166 W(K)=0.0
      DO 170 J=1,NCOF2,2
      DO 170 I=2,MOD
      MM=ISP+1-I
  170 SI(MM,J)= SI(I,J)
      DO 175 J=2,NCOF1,2
      DO 175 I=2,MOD
      MM=ISP+1-I
  175 SI(MM,J)= -SI(I,J)
      MS=ISP-1
      MT=ISP-1+MOD
      DO 180 K=MR,MS
      S=0.0
      DO 185 I=1,NFOUR
      AI=I
  185 S=S+FA(I)*SI(K,I)*AI
  180 W(K)= 3.1416*S/(FACT*SI(K,1))
      DO 190 K=MR,MT
      DO 190 J=1,NTHET
  190 WJHK(K,J)= WJHK(K,J)+W(K)/3.0
      GO TO 435
  425 DO 3 I=1,MOD
      DO 3 J=1,NTHET
    3 WJHK(I,J)= WJHK(I,J)+W(I)/3.0
  435 RETURN
      END

      SUBROUTINE FMWING (MTHET,INDEX,NDJ,DIAM,XCG,ZCG,CHORD)
C
```

```
      DIMENSICN UJHK(16,40),VJHK(16,40),WJHK(16,40),Y(20),RF(7,16),
     1          Z(4,18,40),X(4,18,40)
      DIMENSION CP(16,40)
      DIMENSICN AREX(20,40),AREY(20,40),AREZ(20,40),FX(20,40),FY(20,40)
     1  ,FZ(20,40),FXTCT(20),FYTOT(20),FZTOT(20)
C
      COMMCN/BLKHK1/LS,MB,NFOUR,NSYM,ITAPE
      COMMCN/BLKHK2/UJHK,VJHK,WJHK,Y,RF,Z
      COMMCN/BLKHK5/UJ,ALPHA,BETF
      COMMCN/BLKHK6/CP
      COMMCN/BLKHK7/X
C
    5 FORMAT (1H0,/////45X,22H**FORCES AND MOMENTS**)
 6    FORMAT (1H )
 9    FORMAT (32H0X-FCRCE     Y-FORCE     Z-FORCE)
 10   FORMAT (3E11.3)
   12 FORMAT (47H0PITCHING MOMENT COMPUTED ABOUT AXIS THRU C.G.=,1E11.3)
   13 FORMAT (45H0YAWING MOMENT COMPUTED ABOUT AXIS THRU C.G.=,1E11.3)
   14 FORMAT (46H0RCLLING MCMENT COMPUTED ABOUT AXIS THRU C.G.=,1E11.3)
C INDEX=0 RECTANGULAR WING*   OTHERWISE, INDEX=1*
      BETA= ABS(BETF)
      LS1=LS-1
      NTHE= MTHET+1
      DO 20 K=1,LS
      X(1,K,NTHE)= X(1,K,1)
   20 Z(1,K,NTHE)= Z(1,K,1)
      IF (INDEX) 1125,1125,1130
 1125 DO 25 K=1,LS1
      DELY= Y(2)-Y(1)
      IF (K.NE.1)DELY=0.5*(Y(K+1)-Y(K-1))
      DO 25 J=2,MTHET
      AREZ(K,J)= 0.5*(X(1,K,J+1)-X(1,K,J-1))*DELY
      AREY(K,J)=0.0
   25 AREX(K,J)= 0.5*(Z(1,K,J+1)-Z(1,K,J-1))*DELY
      GO TO 1135
 1130 DO 30 K=2,LS1
      DELY= 0.5*(Y(K+1)-Y(K-1))
      DO 30 J=2,MTHET
      DX1= 0.5*(X(1,K-1,J+1)-X(1,K-1,J-1))
      DX2= 0.5*(X(1,K,J+1)-X(1,K,J-1))
      DX3= 0.5*(X(1,K+1,J+1)-X(1,K+1,J-1))
      AREZ(K,J)= 0.25*(DX3+2.0*DX2+DX1)*DELY
      AREY(K,J)= 0.25*(X(1,K,J+1)-X(1,K,J-1))*(Z(1,K+1,J)-Z(1,K-1,J))
      DZ1= 0.5*(Z(1,K-1,J+1)-Z(1,K-1,J-1))
      DZ2= 0.5*(Z(1,K,J+1)-Z(1,K,J-1))
      DZ3= 0.5*(Z(1,K+1,J+1)-Z(1,K+1,J-1))
   30 AREX(K,J)= 0.25*(DZ3+2.0*DZ2+DZ1)*DELY
      DELY= Y(2)-Y(1)
      DO 35 J=2,MTHET
      DX2= 0.5*(X(1,1,J+1)-X(1,1,J-1))
      DX3= 0.5*(X(1,2,J+1)-X(1,2,J-1))
      AREZ(1,J)= (DX2+0.5*(DX2+DX3))*DELY
      AREY(1,J)= 0.5*(X(1,1,J+1)-X(1,1,J-1))*(Z(1,2,J)-Z(1,1,J))
      DZ2= 0.5*(Z(1,1,J+1)-Z(1,1,J-1))
      DZ3= 0.5*(Z(1,2,J+1)-Z(1,2,J-1))
   35 AREX(1,J)= (DZ2+0.5*(DZ2+DZ3))*DELY
 1135 DELY= 0.5*(Y(LS)-Y(LS1))
      DO 40 J=2,MTHET
      DX2= 0.5*(X(1,LS1,J+1)-X(1,LS1,J-1))
```

```
      DX3= 0.5*(X(1,LS,J+1)-X(1,LS,J-1))
      AREZ(LS,J)= 0.5*(DX3+0.5*(DX2+DX3))*DELY
      AREY(LS,J)= 0.25*(X(1,LS,J+1)-X(1,LS,J-1))*(Z(1,LS,J) -Z(1,LS1,J))
      DZ2= 0.5*(Z(1,LS1,J+1)-Z(1,LS1,J-1))
      DZ3= 0.5*(Z(1,LS,J+1)-Z(1,LS,J-1))
   40 AREX(LS,J)= 0.5*(DZ3+0.5*(DZ2+DZ3))*DELY
      IF (BETA-0.001) 1136,1136,1137
 1137 DO 45 J=2,MTHET
      AREZ(1,J)=0.5*AREZ(1,J)
      AREY(1,J)=0.5*AREY(1,J)
   45 AREX(1,J)=0.5*AREX(1,J)
      DO 50 J=2,MTHET
      CPBAR= CP(2,J)-(CP(2,J)-CP(1,J))*0.75
      FX(1,J)=-AREX(1,J)*CPBAR
      FY(1,J)= AREY(1,J)*CPBAR
   50 FZ(1,J)= AREZ(1,J)*CPBAR
      GO TO 1138
 1136 DO 55 J=2,MTHET
      FX(1,J)=-AREX(1,J)*CP(1,J)
      FY(1,J)= 0.
   55 FZ(1,J)= AREZ(1,J)*CP(1,J)
 1138 DO 60 K=2,LS1
      DO 60 J=2,MTHET
      CPBAR= CP(K,J)+ (CP(K+1,J)-CP(K,J))*(0.5*(  Y(K-1)+Y(K)) +0.25*
     1 (Y(K+1)-Y(K-1))-Y(K))/(Y(K+1)-Y(K))
      FX(K,J)=-AREX(K,J)*CPBAR
      FY(K,J)= AREY(K,J)*CPBAR
   60 FZ(K,J)= AREZ(K,J)*CPBAR
      DO 65 J=2,MTHET
      CPBAR= CP(LS1,J)+(CP(LS,J)-CP(LS1,J))*0.75
      FX(LS,J)=-AREX(LS,J)*CPBAR
      FY(LS,J)= AREY(LS,J)*CPBAR
   65 FZ(LS,J)= AREZ(LS,J)*CPBAR
      DO 145 K=1,LS
      FXTOT(K)=0.0
      FYTOT(K)=0.0
      FZTOT(K)=0.0
      DO 145 J=2,MTHET
      FXTOT(K)= FXTOT(K)+FX(K,J)
      FYTOT(K)= FYTOT(K)+FY(K,J)
  145 FZTOT(K)= FZTOT(K)+FZ(K,J)
      XFORCE=0.0
      YFORCE=0.0
      ZFORCE=0.0
      TRUST= 3.14159*FLOAT(NOJ)*(DIAM/UJ)**2/2.0
      DO 155 K=2,LS
      XFORCE=XFORCE +FXTOT(K)
      YFORCE=YFORCE +FYTOT(K)
  155 ZFORCE=ZFORCE +FZTOT(K)
      IF (BETA-0.001) 1160,1160,1165
 1165 XFORCE= FXTOT(1)+XFORCE
      YFORCE= FYTOT(1)+YFORCE
      ZFORCE= FZTOT(1)+ZFORCE
      YFORCE= YFORCE/TRUST
      XFORCE= XFORCE/TRUST
      ZFORCE= ZFORCE/TRUST
      GO TO 1170
 1160 XFORCE= FXTOT(1)+2.0*XFORCE
      YFORCE= 0.0
```

```
      ZFORCE= FZTOT(1)+2.0*ZFORCE
      XFORCE= XFORCE/TRUST
      YFORCE= YFORCE/TRUST
      ZFORCE= ZFORCE/TRUST
 1170 WRITE (6,5)
      WRITE (6,6)
      WRITE (6,9)
      WRITE (6,10) XFORCE,YFORCE,ZFORCE
      YAW=0.0
      PITCH=0.0
      ROLL=0.0
      IF (BETA-0.001) 1175,1175,1180
 1180 DO 161 K=1,LS
      DO 161 J=2,MTHET
  161 PITCH= PITCH +FX(K,J)*(Z(1,K,J)-ZCG) +FZ(K,J)*(XCG-X(1,K,J))
      DO 162 K=2,LS1
  162 YAW= YAW+FXTOT(K)*Y(K)
      YAW= YAW+FXTOT(1)*(Y(2)+0.25*(Y(1)-Y(2)))+FXTOT(LS)*(Y(LS1)
     1  +0.25*(Y(LS)-Y(LS1)))
      DO 163 K=1,LS
      DO 163 J=2,MTHET
  163 YAW= YAW+FY(K,J)*(XCG-X(1,K,J))
      DO 164 K=2,LS1
  164 ROLL= ROLL-FZTOT(K)*Y(K)
      ROLL= ROLL-FZTOT(1)*(Y(2)+0.25*(Y(1)-Y(2)))-FZTOT(LS)*(Y(LS)
     1  +0.25*(Y(LS)-Y(LS1)))
      DO 166 K=1,LS
      DO 166 J=2,MTHET
  166 ROLL= ROLL +FY(K,J)*(Z(1,K,J)-ZCG)
      PITCH= PITCH/(TRUST*CHORD)
      YAW= YAW/(TRUST*CHORD)
      ROLL= ROLL/(TRUST*CHORD)
      GO TO 1185
 1175 DO 160 K=2,LS
      DO 160 J=2,MTHET
  160 PITCH= PITCH +FX(K,J)*(Z(1,K,J)-ZCG) +FZ(K,J)*(XCG-X(1,K,J))
      PITCH= 2.0*PITCH
      DO 165 J=2,MTHET
  165 PITCH= PITCH +FX(1,J)*(Z(1,1,J)-ZCG) +FZ(1,J)*(XCG-X(1,1,J))
      PITCH= PITCH/(TRUST*CHORD)
 1185 WRITE (6,6)
      WRITE (6,12) PITCH
      WRITE (6,13) YAW
      WRITE (6,14) ROLL
      RETURN
      END


      SUBROUTINE FMBODY (MTHET,YT,ZT,YTAIL,ZTAIL,NDJ,DIAM,XCG,CHORD)
C
      DIMENSION UJHK(16,40),VJHK(16,40),WJHK(16,40),X(20),RF(7,16),
     1  Z(4,18,40),Y(4,18,40)
      DIMENSION CP(16,40)
      DIMENSION AREX(20,40),AREY(20,40),AREZ(20,40),FX(20,40),FY(20,40)
     1   ,FZ(20,40),FXTOT(20),FYTOT(20),FZTOT(20)
C
      COMMON/BLKHK1/LS,MB,NFOUR,NSYM,ITAPE
      COMMON/BLKHK2/UJHK,VJHK,WJHK,X,RF,Z
      COMMON/BLKHK5/UJ,ALPHA,BETA
```

```
      COMMON/BLKHK6/CP
      COMMON/BLKHK8/Y
C
    5 FORMAT (1H0,/////45X,22H**FORCES AND MOMENTS**)
    6 FORMAT (1H )
    9 FORMAT (32H0X-FORCE      Y-FORCE      Z-FORCE)
   10 FORMAT(3E11.3)
   12 FORMAT (47H0PITCHING MOMENT COMPUTED ABOUT AXIS THRU C.G.=,1E11.3)
   13 FORMAT (45H0YAWING MOMENT COMPUTED ABOUT AXIS THRU C.G.=,1E11.3)
      NTHE=MTHET/2 +1
      LST=LS+1
      LS1=LS-1
      NTH=NTHE+1
      DO 20 K=1,LS
      Y(1,K,NTH)= -Y(1,K,NTHE-1)
   20 Z(1,K,NTH)= Z(1,K,NTHE-1)
      DO 25 J=1,NTH
      Y(1,LST,J)= YTAIL
   25 Z(1,LST,J)= ZTAIL
      DO 30 K=2,LS
      DELX= 0.5*(X(K+1)-X(K-1))
      AREX(K,1)= 0.5*(Z(1,K+1,1)-Z(1,K-1,1))*Y(1,K,2)
      AREY(K,1)= 0.0
      AREZ(K,1)= 0.25*(Y(1,K+1,2)+2.0*Y(1,K,2)+Y(1,K-1,2))*DELX
      DO 30 J=2,NTHE
      DY1= 0.5*(Y(1,K-1,J+1)-Y(1,K-1,J-1))
      DY2= 0.5*(Y(1,K,J+1)-Y(1,K,J-1))
      DY3= 0.5*(Y(1,K+1,J+1)-Y(1,K+1,J-1))
      AREZ(K,J)= 0.25*(DY3+2.0*DY2+DY1)*DELX
      DZ1= 0.5*(Z(1,K-1,J+1)-Z(1,K-1,J-1))
      DZ2= 0.5*(Z(1,K,J+1)-Z(1,K,J-1))
      DZ3= 0.5*(Z(1,K+1,J+1)-Z(1,K+1,J-1))
      AREY(K,J)= 0.25*(DZ3+2.0*DZ2+DZ1)*DELX
   30 AREX(K,J)= 0.25*(Z(1,K+1,J)-Z(1,K-1,J))*(Y(1,K,J+1)-Y(1,K,J-1))
      DELX=0.5*X(2)
      AREX(1,1)= 0.5*(Z(1,2,1)-ZT)*Y(1,1,2)
      AREY(1,1)=0.0
      AREZ(1,1)= 0.25*(Y(1,2,2)+2.0*Y(1,1,2)+YT)*DELX
      DO 35 J=2,NTHE
      DY2= 0.5*(Y(1,1,J+1)-Y(1,1,J-1))
      DY3= 0.5*(Y(1,2,J+1)-Y(1,2,J-1))
      AREZ(1,J)= 0.25*(DY3+2.0*DY2)*DELX
      DZ2= 0.5*(Z(1,1,J+1)-Z(1,1,J-1))
      DZ3= 0.5*(Z(1,2,J+1)-Z(1,2,J-1))
      AREY(1,J)= 0.25*(DZ3+2.0*DZ2)*DELX
   35 AREX(1,J)= 0.25*(Z(1,2,J)-ZT)*(Y(1,1,J+1)-Y(1,1,J-1))
      DO 40 K=1,LS
      DO 40 J=NTH,MTHET
      NON= NTH -(J-MTHET/2)
      AREZ(K,J)= AREZ(K,NON)
      AREY(K,J)=-AREY(K,NON)
   40 AREX(K,J)= AREX(K,NON)
      DO 45 K=2,LS1
      DO 45 J=1,MTHET
      CPBAR= CP(K,J)+(CP(K+1,J)-CP(K,J))*(0.5*(X(K-1)+X(K)) +0 25*
     1   (X(K+1)-X(K-1))-X(K))/(X(K+1)-X(K))
      FX(K,J)= AREX(K,J)*CPBAR
      FY(K,J)=-AREY(K,J)*CPBAR
   45 FZ(K,J)= AREZ(K,J)*CPBAR
```

```
      DO 50 J=1,MTHET
      CPBAR= CP(1,J)+ (CP(2,J)-CP(1,J))*(0.5*X(1)+0.25*X(2)-X(1))
     1   /(X(2)-X(1))
      FX(1,J)= AREX(1,J)*CPBAR
      FY(1,J)= -AREY(1,J)*CPBAR
      FZ(1,J)= AREZ(1,J)*CPBAR
      CPBAR= CP(LS,J)+ (CP(LS,J)-CP(LS1,J))*(0.5*(X(LS)+X(LS1))+0.25*
     1   (X(LS)-X(LS1))-X(LS))/(X(LS)-X(LS1))
      FX(LS,J)= AREX(LS,J)*CPBAR
      FY(LS,J)= -AREY(LS,J)*CPBAR
   50 FZ(LS,J)= AREZ(LS,J)*CPBAR
      DO 145 K=1,LS
      FXTOT(K)=0.0
      FYTOT(K)=0.0
      FZTOT(K)=0.0
      DO 145 J=1,MTHET
      FXTOT(K)=FXTOT(K)+FX(K,J)
      FYTOT(K)=FYTOT(K)+FY(K,J)
  145 FZTOT(K)=FZTOT(K)+FZ(K,J)
      TRUST= 3.14159*FLOAT(NDJ)*(DIAM/UJ)**2/2.0
      DO 150 K=1,LS
      FXTOT(K)= FXTOT(K)/TRUST
      FYTOT(K)= FYTOT(K)/TRUST
  150 FZTOT(K)= FZTOT(K)/TRUST
      XFORCE=0.0
      YFORCE=0.0
      ZFORCE=0.0
      DO 155 K=1,LS
      XFORCE=XFORCE+FXTOT(K)
      YFORCE=YFORCE+FYTOT(K)
  155 ZFORCE=ZFORCE+FZTOT(K)
      WRITE (6,5)
      WRITE (6,6)
      WRITE (6,9)
      WRITE (6,10) XFORCE,YFORCE,ZFORCE
      YAW=0.0
      PITCH=0.0
      DO  175 K=1,LS
      IF (X(K)-XCG) 175,176,176
  175 CONTINUE
  176 MOMENT=K
      XDIS= X(MOMENT)-XCG
      IF (MOMENT-1) 1111,1111,1180
 1175 DO 160 K=MOMENT,LS
      YAW=YAW+FYTOT(K)*(X(K)-X(MOMENT)+XDIS)
  160 PITCH=PITCH-FZTOT(K)*(X(K)-X(MOMENT)+XDIS)
      GO TO 1185
 1180 MENT=MOMENT-1
      DO 165 K=1,MENT
      YAW=YAW-FYTOT(K)*(X(MOMENT)-X(K)-XDIS)
  165 PITCH=PITCH+FZTOT(K)*(X(MOMENT)-X(K)-XDIS)
      IF (LS-MOMENT) 1111,1111,1175
 1185 DO 170 K=1,LS
      DO 170 J=1,MTHET
      YAW=YAW-FX(K,J)*Y(1,K,J)/TRUST
  170 PITCH= PITCH+FX(K,J)*Z(1,K,J)/TRUST
      YAW= YAW/CHORD
      PITCH= PITCH/CHORD
      WRITE (6,6)
```

```
      WRITE (6,12) PITCH
      WRITE (6,13) YAW
      RETURN
 1111 WRITE (6,601)
 601  FORMAT (1HO,30H**ERROR IN MOMENT DATA INPUT**)
      STOP
      END


      SUBROUTINE WMOD3 (MTHET,IDIS,NBOOL,MEXIT)
C
      DIMENSION UX(16,40),UY(16,40),UZ(16,40),YCOMM(20),RF(7,16)
      DIMENSION X(4,18,40),Z(4,18,40),DNORM(4,16,40),DTANG(4,16,40),
     1          DVOL(4,16,40),FLUX(4,16,40),PHI(4,16,40)
      DIMENSION VX(1,16,40),VY(1,16,40),VZ(1,16,40)
      DIMENSION SI(40,20),CS(40,20),C(30,16),D(30,16)
      DIMENSION E(16),Y(40)
C
      COMMON/BLKHK1/LS,MB,NFOUR,NSYM,ITAPE
      COMMON/BLKHK2/UX,UY,UZ,YCOMM,RF,Z
      COMMON/BLKHK3/SI,CS
      COMMON/BLKHK5/UJ,ALPHA,BETF
      COMMON/BLKHK7/X
      COMMON/BLKH14 /VY
      COMMON/BLKH15 /NDOWN,IREPET
C
      EQUIVALENCE (FLUX(1),DNORM(1)),(PHI(1),DTANG(1))
C
      REWIND ITAPE
      DO 15 K=1,LS
      Y(K) = YCOMM(K)
      READ (ITAPE) DUMMY,E(K),(C(I,K),D(I,K), I=1,NFOUR)
   15 CONTINUE
      BETA= ABS(BETF)
      LS1=LS-1
      MT1=MTHET+1
      DO 60 K=1,LS1
      DO 60 I=1,IDIS
      X(I,K,MT1)=X(I,K,1)
   60 Z(I,K,MT1)=Z(I,K,1)
      DO 65 K=1,LS1
      DO 65 I=2,IDIS
      DO 65 J=1,MTHET
      DNORM(I,K,J)=SQRT((X(I,K,J)-X(I-1,K,J))**2 +(Z(I,K,J)-Z(I-1,K,J))
     1   **2)
   65 DTANG(I,K,J)=SQRT((X(I,K,J+1)-X(I,K,J))**2 +(Z(I,K,J+1)-Z(I,K,J))
     1   **2)
      DO 70 K=1,LS1
      DO 70 I=2,IDIS
      DO 70 J=1,MTHET
      IF (I-IDIS) 1145,1146,1145
 1145 IF (I-2) 1150,1151,1150
 1146 DN=DNORM(IDIS,K,J)
      GO TO 1152
 1151 DN=0.5*DNORM(3,K,J)+DNORM(2,K,J)
      GO TO 1152
 1150 DN=0.5*(DNORM(I+1,K,J)+DNORM(I,K,J))
 1152 IF (J-1) 1155,1156,1155
 1156 DT=0.5*(DTANG(I,K,1)+DTANG(I,K,MTHET))
```

```
      GO TO 1157
 1155 DT=0.5*(DTANG(I,K,J)+DTANG(I,K,J-1))
 1157 IF (K-1) 1159,1158,1159
 1158 DY=Y(2)
      GO TO 1160
 1159 DY=0.5*(Y(K+1)-Y(K-1))
 1160 DVOL(I,K,J)= DN*DT*DY
   70 CONTINUE
      DO 75 K=1,LS
      DO 75 I=2,IDIS
      RL=ALOG(RF(I,K))
      DO 75 J=1,MTHET
      AA=E(K)*RL
      REV=1.0
      DO 80 N=1,NFOUR
      REV=REV*RF(1,K)/RF(I,K)
   80 AA=AA+REV*(-D(N,K)*CS(J,N)+C(N,K)*SI(J,N))
   75 PHI(I,K,J)=AA
      DO 85 K=2,LS1
      DO 85 I=2,IDIS
      DO 85 J=1,MTHET
   85 FLUX(I,K,J)= DVOL(I,K,J)*(PHI(I,K+1,J)-2.0*PHI(I,K,J)+PHI(I,K-1,J
     1 )-(PHI(I,K+1,J)-PHI(I,K,J))*(Y(K+1)-2.0*Y(K)+Y(K-1))/(Y(K+1)
     2  -Y(K)))/(12.566*(Y(K)-Y(K-1))**2)
C     SIGN IN FLUX IS PLUS,DUE TO COMBINATION OF MINUS SIGNS.
      IF (BETA-0.001) 1200,1200,1205
 1205 DO 86 K=1,LS
      DO 86 M=1,MTHET
      VX(1,K,M)=0.
      VY(1,K,M)=0.
   86 VZ(1,K,M)=0.
      LS3=LS-3
      DO 87 K=4,LS3
      IB=MAX0(2,K-4)
      LB=MINO(LS1,K+4)
      DO 87 LKL=IB,LB
      DO 87 M=1,MTHET
      DO 87 I=2,IDIS
      DO 87 J=1,MTHET
      CBS=((X(1,K,M)-X(I,LKL,J))**2+(Z(1,K,M)-Z(I,LKL,J))**2
     1 +(Y(K)-Y(LKL))**2)**1.5
      VX(1,K,M)= VX(1,K,M)+FLUX(I,LKL,J)*(X(1,K,M)-X(I,LKL,J))/CBS
      VY(1,K,M)= VY(1,K,M)+FLUX(I,LKL,J)*(Y(K)-Y(LKL))/CBS
   87 VZ(1,K,M)= VZ(1,K,M)+FLUX(I,LKL,J)*(Z(1,K,M)-Z(1,LKL,J))/CBS
      IF (LS.LE.13) GO TO 1210
      LS4=LS-4
      LS5=LS4-4
      DO 88 KA=4,LS4,LS5
      KB=KA+1
      IF (KA.EQ.4) KC=5
      IF (KA.EQ.LS4) KC=-5
      DO 88 K=KA,KB
      DO 88 M=1,MTHET
      DO 88 I=2,IDIS
      DO 88 J=1,MTHET
      CBS= ((X(1,K,M)-X(I,K+KC,J))**2 +(Z(1,K,M)-Z(I,K+KC,J))**2
     1  +(Y(K)-Y(K+KC))**2)**1.5
      VX(1,K,M)= VX(1,K,M)+FLUX(I,K+KC,J)*(X(1,K,M)-X(I,K+KC,J))/CBS
      VY(1,K,M)= VY(1,K,M)+FLUX(I,K+KC,J)*(Y(K)-Y(K+KC))/CBS
```

```
   88 VZ(1,K,M)= VZ(1,K,M)+FLUX(I,K+KC,J)*(Z(1,K,M)-Z(I,K+KC,J))/CBS
      GO TO 1210
 1200 DO 90 I=2,IDIS
      DO 90 J=1,MTHET
   90 FLUX(I,1,J)= DVOL(I,1,J)*2.0*(PHI(I,2,J)-PHI(I,1,J))/(12.566*Y(2)
     1    *Y(2))
      DO 91 K=1,LS1
      DO 91 I=2,IDIS
      DO 91 J=1,MTHET
   91 PHI(I,K,J)= FLUX(I,K,J)
      DO 92 K=1,LS1
      DO 92 I=2,IDIS
      DO 92 J=1,MTHET
   92 FLUX(I,K+4,J)= PHI(I,K,J)
      LCOMP= LS+4
      DO 95 K=1,LS
      DO 95 I=1,IDIS
      DO 95 J=1,MTHET
      PHI(I,K,J)=X(I,K,J)
   95 DVOL(I,K,J)= Z(I,K,J)
      DO 100 K=1,LS1
      DO 100 I=1,IDIS
      DO 100 J=1,MTHET
      X(I,K+4,J)=PHI(I,K,J)
  100 Z(I,K+4,J)=DVOL(I,K,J)
      DO 105 K=1,4
      N=6-K
      DO 105 I=1,IDIS
      DO 105 J=1,MTHET
      X(I,K,J)= PHI(I,N,J)
  105 Z(I,K,J)= DVOL(I,N,J)
C     FLUX HAVE SAME SIGNS ON BOTH SIDES OF JET,DUE TO SECOND DERIVATIVE
      DO 110 K=1,4
      N=10-K
      DO 110 I=2,IDIS
      DO 110 J=1,MTHET
  110 FLUX(I,K,J)= FLUX(I,N,J)
      DO 115 K=1,LS1
  115 Y(K+20)=Y(K)
      DO 120 K=1,LS1
  120 Y(K+4)=Y(K+20)
      DO 125 K=1,4
      N=10-K
  125 Y(K)=-Y(N)
      DO 130 K=1,LCOMP
      DO 130 M=1,MTHET
      VX(1,K,M)=0.0
      VY(1,K,M)=0.0
  130 VZ(1,K,M)=0.0
      LCOM3=LCOMP-3
      DO 135 K=5,11
      IB=MINO(3,K-4)
      LB=MINO(LCOM3+2,K+4)
      DO 135 LKL=IB,LB
      DO 135 M=1,MTHET
      DO 135 I=2,IDIS
      DO 135 J=1,MTHET
      CBS=((X(1,K,M)-X(I,LKL,J))**2+(Z(1,K,M)-Z(I,LKL,J))**2
     1 +(Y(K)-Y(LKL))**2)**1.5
```

```
      VX(1,K,M)= VX(1,K,M)+FLUX(I,LKL,J)*(X(1,K,M)-X(I,LKL,J))/CBS
      VY(1,K,M)= VY(1,K,M)+FLUX(I,LKL,J)*(Y(K)-Y(LKL))/CBS
  135 VZ(1,K,M)= VZ(1,K,M)+FLUX(I,LKL,J)*(Z(1,K,M)-Z(I,LKL,J))/CBS
      IF (LCOM3.LE.11) GO TO 1210
      DO 140 K=12,LCOM3
      IB=K-4
      LB=MINO(LCOM3+2,K+4)
      DO 140 LKL=IB,LB
      DO 140 M=1,MTHET
      DO 140 I=2,IDIS
      DO 140 J=1,MTHET
      CBS=((X(1,K,M)-X(I,LKL,J))**2+(Z(1,K,M)-Z(I,LKL,J))**2
     1 +(Y(K)-Y(LKL))**2)**1.5
      VX(1,K,M)= VX(1,K,M)+FLUX(I,LKL,J)*(X(1,K,M)-X(I,LKL,J))/CBS
      VY(1,K,M)= VY(1,K,M)+FLUX(I,LKL,J)*(Y(K)-Y(LKL))/CBS
  140 VZ(1,K,M)= VZ(1,K,M)+FLUX(I,LKL,J)*(Z(1,K,M)-Z(I,LKL,J))/CBS
 1210 IF (NBOCL-1) 1181,1180,1181
 1180 IF (BETA-0.001) 1183,1183,1184
 1183 M3=3
      M6=6
      M7=7
      M8=8
      DO 149 J=1,MTHET
      VX(1,3,J)=VX(1,7,J)
      VY(1,3,J)=VY(1,7,J)
  149 VZ(1,3,J)=VZ(1,7,J)
      GO TO 1185
 1184 M2=MEXIT-3
      M3=M2+1
      M4=M2+2
      M5=M2+3
      M6=M2+4
      M7=M2+5
      M8=M2+6
      YN1= (Y(M4)-Y(M3))*(Y(M4)-Y(M7))/(Y(M2)-Y(M3))/(Y(M2)-Y(M7))
      YN2= (Y(M4)-Y(M2))*(Y(M4)-Y(M7))/(Y(M3)-Y(M2))/(Y(M3)-Y(M7))
      YN3= (Y(M4)-Y(M2))*(Y(M4)-Y(M3))/(Y(M7)-Y(M2))/(Y(M7)-Y(M3))
      DO 151 J=1,MTHET
      VX(1,M4,J)= 0.5*(VX(1,M4,J)+YN1*VX(1,M2,J)+YN2*VX(1,M3,J)
     1     +YN3*VX(1,M7,J))
      VY(1,M4,J)= 0.5*(VY(1,M4,J)+YN1*VY(1,M2,J)+YN2*VY(1,M3,J)
     1     +YN3*VY(1,M7,J))
  151 VZ(1,M4,J)= 0.5*(VZ(1,M4,J)+YN1*VZ(1,M2,J)+YN2*VZ(1,M3,J)
     1     +YN3*VZ(1,M7,J))
 1185 YN1= (Y(M6)-Y(M7))*(Y(M6)-Y(M8))/(Y(M3)-Y(M7))/(Y(M3)-Y(M8))
      YN2= (Y(M6)-Y(M3))*(Y(M6)-Y(M8))/(Y(M7)-Y(M3))/(Y(M7)-Y(M8))
      YN3= (Y(M6)-Y(M3))*(Y(M6)-Y(M7))/(Y(M8)-Y(M3))/(Y(M8)-Y(M7))
      DO 152 J=1,MTHET
      VX(1,M6,J)= 0.5*(VX(1,M6,J) +YN1*VX(1,M3,J)+YN2*VX(1,M7,J)
     1     +YN3*VX(1,M8,J))
      VY(1,M6,J)= 0.5*(VY(1,M6,J) +YN1*VY(1,M3,J)+YN2*VY(1,M7,J)
     1     +YN3*VY(1,M8,J))
  152 VZ(1,M6,J)= 0.5*(VZ(1,M6,J) +YN1*VZ(1,M3,J)+YN2*VZ(1,M7,J)
     1     +YN3*VZ(1,M8,J))
 1181 IF (BETA-0.001) 1182,1182,1190
 1182 DO 160 K=5,LCOM3
      N=K-4
      DO 160 L=1,MTHET
      UX(N,L)=UX(N,L)+VX(1,K,L)
```

```
      UY(N,L)=UY(N,L)+VY(1,K,L)
  160 UZ(N,L)= UZ(N,L)+VZ(1,K,L)
      DO 153 K=5,LCOM3
      DO 153 J=1,MTHET
      N=K-4
  153 VY(1,N,J)=VY(1,K,J)
      LCOM6=LCOM3-3
      DO 155 K=LCOM6,LCOMP
      DO 155 J=1,MTHET
  155 VY(1,K,J)=0.
      DO 154 K=1,LS
      DO 154 I=1,IDIS
      DO 154 J=1,MTHET
      X(I,K,J)=PHI(I,K,J)
  154 Z(I,K,J)=DVOL(I,K,J)
      GO TO 1195
 1190 DO 161 K= 4,LCOM3
      DO 161 L=1,MTHET
      UX(K,L)= UX(K,L)+VX(1,K,L)
      UY(K,L)= UY(K,L)+VY(1,K,L)
  161 UZ(K,L)= UZ(K,L)+VZ(1,K,L)
 1195 NDOWN=0
      RETURN
      END


      SUBROUTINE BMOD3 (MTHET,IDIS,NJET)
C
      DIMENSION UX(16,40),UY(16,40),UZ(16,40),X(20),RF(7,16),
     1   Y(4,18,40),Z(4,18,40),E(16),DNORM(4,16,40),DTANG(4,16,40),
     2   DVOL(4,16,40),FLUX(4,16,40),PHI(4,16,40)
      DIMENSION VX(1,16,40),VY(1,16,40),VZ(1,16,40)
      DIMENSION SI(40,20),CS(40,20),C(30,16),D(30,16)
C
      COMMON/BLKHK1/LS,MB,NFOUR,NSYM,ITAPE
      COMMON/BLKHK2/UX,UY,UZ,X,RF,Z
      COMMON/BLKHK3/SI,CS
      COMMON/BLKHK5/UJ,ALPHA,BETF
      COMMON/BLKHK8/Y
      COMMON/BLKH13 /VX
C
      EQUIVALENCE (FLUX(1),DNORM(1)),(PHI(1),DTANG(1))
C
      REWIND ITAPE
      DO 20 K=1,LS
      READ (ITAPE) E(K),(C(I,K),D(I,K),I=1,NFOUR)
   20 CONTINUE
    8 LS1=LS-1
      MT1=MTHET+1
      DO 40 K=2,LS1
      DO 40 I=1,IDIS
      Y(I,K,MT1)=Y(I,K,1)
      Z(I,K,MT1)=Z(I,K,1)
      Y(I,K,MT1+1)=Y(I,K,2)
   40 Z(I,K,MT1+1)=Z(I,K,2)
      DO 45 K=2,LS1
      DO 45 I=2,IDIS
      DO 45 J=1,MT1
      DNORM(I,K,J)=SQRT((Y(I,K,J)-Y(I-1,K,J))**2 +(Z(I,K,J)-Z(I-1,K,J))
```

```
     1     **2)
  45 DTANG(I,K,J)=SQRT((Y(I,K,J+1)-Y(I,K,J))**2 +(Z(I,K,J+1)-Z(I,K,J))
     1     **2)
      DO 50 K=2,LS1
      DO 50 I=2,IDIS
      DO 50 J=1,MTHET
      IF (I-IDIS) 1145,1146,1145
 1145 IF (I-2) 1150,1151,1150
 1146 DN=DNORM(IDIS,K,J)
      GO TO 1152
 1151 DN=0.5*DNORM(3,K,J)+DNORM(2,K,J)
      GO TO 1152
 1150 DN=0.5*(DNORM(I+1,K,J)+DNORM(I,K,J))
 1152 IF (J-1) 1155,1156,1155
 1156 DT=0.5*(DTANG(I,K,1)+DTANG(I,K,MTHET))
      GO TO 1157
 1155 DT=0.5*(DTANG(I,K,J)+DTANG(I,K,J-1))
 1157 DX=0.5*(X(K-1)+X(K+1))
   50 DVOL(I,K,J)= DN*DT*DX
      DO 70 K=1,LS
      DO 70 I=2,IDIS
      DO 70 J=1,MTHET
      AA=-E(K)*RF(1,K)/RF(I,K)
      REV=1.0
      DO 75 N=1,NFOUR
      REV=REV*RF(1,K)/RF(I,K)
   75 AA=AA+REV*(-D(N,K)*CS(J,N)+C(N,K)*SI(J,N))
   70 PHI(I,K,J)=AA
      LS2=LS-2
C     SIGN IN FLUX IS PLUX,DUE TO COMBINATION OF TWO MINUS SIGNS.
      DO 80 K=2,LS1
      WX1= X(K-1)-X(K)
      WX2= X(K-1)-X(K+1)
      WX3= X(K)-X(K+1)
      DO 80 I=2,IDIS
      DO 80 J=1,MTHET
   80 FLUX(I,K,J)= (PHI(I,K-1,J)/WX1/WX2 -PHI(I,K,J)/WX1/WX3 +PHI(I,K+1,
     1   J)/WX2/WX3 -0.5*E(K)*RF(1,K)/RF(I,K)**3)*DVOL(I,K,J)/6.2832
      DO 81 K=1,LS
      DO 81 M=1,MTHET
      VX(1,K,M)=0.0
      VY(1,K,M)=0.0
   81 VZ(1,K,M)=0.0
      LS3=LS-3
      NTHE=MTHET/2 +1
      IF (ABS(BETF).GT.0.001) NTHE=MTHET
      NJL=NJET-2
      NJR=NJET+3
      DO 85 K=3,NJL
      IB=MAX0(2,K-4)
      DO 85 LKL=IB,NJR
      DO 85 M=1,NTHE
      DO 85 I=2,IDIS
      DO 85 J=1,MTHET
      CBS= ((X(K)-X(LKL))**2 +(Y(1,K,M)-Y(I,LKL,J))**2 +(Z(1,K,M)
     1  -Z(I,LKL,J))**2)**1.5
      VX(1,K,M)= VX(1,K,M) +FLUX(I,LKL,J)*(X(K)-X(LKL))/CBS
      VY(1,K,M)= VY(1,K,M) +FLUX(I,LKL,J)*(Y(1,K,M)-Y(I,LKL,J))/CBS
   85 VZ(1,K,M)= VZ(1,K,M) +FLUX(I,LKL,J)*(Z(1,K,M)-Z(I,LKL,J))/CBS
```

```
      NJ1=NJET-1
      NJ2=NJET+2
      DO 90 K=NJ1,NJ2
      IB=K-4
      LB=K+4
      DO 90 LKL=IB,LB
      DO 90 M=1,NTHE
      DO 90 I=2,IDIS
      DO 90 J=1,MTHET
      CBS= ((X(K)-X(LKL))**2 +(Y(1,K,M)-Y(I,LKL,J))**2 +(Z(1,K,M)
     1  -Z(I,LKL,J))**2)**1.5
      VX(1,K,M)= VX(1,K,M) +FLUX(I,LKL,J)*(X(K)-X(LKL))/CBS
      VY(1,K,M)= VY(1,K,M) +FLUX(I,LKL,J)*(Y(1,K,M)-Y(I,LKL,J))/CBS
   90 VZ(1,K,M)= VZ(1,K,M) +FLUX(I,LKL,J)*(Z(1,K,M)-Z(I,LKL,J))/CBS
      DO 95 K=NJR,LS2
      LB=MINO(LS1,K+4)
      DO 95 LKL=NJL,LB
      DO 95 M=1,NTHE
      DO 95 I=2,IDIS
      DO 95 J=1,MTHET
      CBS= ((X(K)-X(LKL))**2 +(Y(1,K,M)-Y(I,LKL,J))**2 +(Z(1,K,M)
     1  -Z(I,LKL,J))**2)**1.5
      VX(1,K,M)= VX(1,K,M) +FLUX(I,LKL,J)*(X(K)-X(LKL))/CBS
      VY(1,K,M)= VY(1,K,M) +FLUX(I,LKL,J)*(Y(1,K,M)-Y(I,LKL,J))/CBS
   95 VZ(1,K,M)= VZ(1,K,M) +FLUX(I,LKL,J)*(Z(1,K,M)-Z(I,LKL,J))/CBS
      N=NJET-1
      N2=N-2
      N3=N-1
      N7=N+1
      XN1= (X(N)-X(N3))*(X(N)-X(N7))/(X(N2)-X(N3))/(X(N2)-X(N7))
      XN2= (X(N)-X(N2))*(X(N)-X(N7))/(X(N3)-X(N2))/(X(N3)-X(N7))
      XN3= (X(N)-X(N2))*(X(N)-X(N3))/(X(N7)-X(N2))/(X(N7)-X(N3))
      DO 110 J=1,NTHE
      VX(1,N,J)= 0.5*(VX(1,N,J) +XN1*VX(1,N2,J)+XN2*VX(1,N3,J)
     1     +XN3*VX(1,N7,J))
      VY(1,N,J)= 0.5*(VY(1,N,J) +XN1*VY(1,N2,J)+XN2*VY(1,N3,J)
     1     +XN3*VY(1,N7,J))
  110 VZ(1,N,J)= 0.5*(VZ(1,N,J) +XN1*VZ(1,N2,J)+XN2*VZ(1,N3,J)
     1     +XN3*VZ(1,N7,J))
  180 DO 100 K=1,LS
      DO 100 L=1,NTHE
      UX(K,L) = UX(K,L)+VX(1,K,L)
      UY(K,L) = UY(K,L)+VY(1,K,L)
  100 UZ(K,L)= UZ(K,L)+VZ(1,K,L)
      RETURN
      END
```

```
      PROGRAM LFTSR(INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
     1TAPE7=PUNCH,TAPE2,TAPE3)
C
      READ (5,501) ISTART,ISTOP
      IF (ISTART-2) 10,20,30
   10 CALL CHAIN1
      IF (ISTOP-1) 50,50,20
   20 CALL CHAIN6
      IF (ISTOP-2) 50,50,30
   30 CALL CHAIN7
   50 CONTINUE
      WRITE (6,601)
      STOP
  501 FORMAT (2I5)
  601 FORMAT (1H0,////+8X,24H***END OF COMPUTATION***)
      END


      SUBROUTINE CHAIN1
C
C     THIS PROGRAM CALCULATES THE DOWNWASH CONTROL POINT MATRIX
C
      DIMENSION GAUSS(50),DLDDN(16),DLDDO(16),FROWR(36,50),THETB(20,4),
     1THETAA(30,16),FORR(30,16),NOMB(20,3),NQ(3),THETA(4),ETA(20),YDWASH
     2(150),FLPOS(10),NSEC(20),XDWASH(150),YSTAT(50),NCP(50),
     3ARRAY(12),TITLE(6),GAUFFA(50),Y(10),NSQ(10),AMLE(30),AMTE(30),
     4YLEAD(31),XLEAD(31),YTRAIL(31),XTRAIL(31)
C
      COMMON GAUSS,THETB,THETAA,FORR,NOMB,NQ,THETA,ETA,YDWASH,FLPOS,NSEC
C
      DATA PIE,XLEAD(1),VJ/3.14159265,0.,16./
      DATA Y(1),NSQ(1),NSQ(2),NSQ(3)/-1.0,16,16,7/
      DATA TITLE/6HDOWNWA,6HSH CON,6HTROL P,6HOINT M,6HATRIX,,6H D     /
C
      REWIND 3
      THETA(1)=0.0
      READ (5,123) ARRAY
      READ (5,121) NYSTAT,MSPAN,NCHORD,NEED,NFLAP,NODE1,NODE3,NAY1,NOLED
     1,NOTED
      READ (5,122) SPACE,FMACH,FBO
      READ (5,122) (YSTAT(I),I=1,NYSTAT)
      READ (5,122) (FLPOS(I),I=1,NFLAP)
      NOL=NOLED-1
      NOT=NOTED-1
      READ (5,122) (AMLE(I),I=1,NOL)
      READ (5,122) (AMTE(I),I=1,NOT)
      READ (5,122) (YLEAD(I),I=1,NOLED)
      READ (5,122) (YTRAIL(I),I=1,NOTED)
      XTRAIL(1)=2.0*FBO
      DO 1 I=2,NOLED
      XLEAD(I)=XLEAD(I-1)+AMLE(I-1)*(YLEAD(I)-YLEAD(I-1))
1     CONTINUE
      DO 2 I=2,NOTED
      XTRAIL(I)=XTRAIL(I-1)+AMTE(I-1)*(YTRAIL(I)-YTRAIL(I-1))
2     CONTINUE
      S=1.0/FBO
      MCBS=MSPAN*NCHORD
```

```
      BOF=2.0*FBO
      WRITE (6,124) ARRAY
      WRITE (6,97) MSPAN,NCHORD,NFLAP,NEED
      DO 3 I=1,NFLAP
      WRITE (6,98) I,FLPOS(I)
      FLPOS(I)=ACOS(1.0-2.0*FLPOS(I))
3     CONTINUE
C     SET UP CONTROL POINT LOCATIONS
      IF (SPACE) 6,4,7
4     READ (5,121) (NCP(I),I=1,NYSTAT)
      NDWASH=0
      LC2=0
      DO 5 I=1,NYSTAT
      NDWASH=NDWASH+NCP(I)
      LC1=LC2+1
      LC2=LC2+NCP(I)
      READ (5,99) (XDWASH(L),L=LC1,LC2)
5     CONTINUE
      GO TO 10
6     WRITE (6,100)
      GO TO 96
7     NXSTAT=1.0/SPACE
      IF (NEED.EQ.0) NXSTAT=NXSTAT+1
      DO 9 I=1,NYSTAT
      L=NEED
      DO 8 J=1,NXSTAT
      XL=L
      K=(I-1)*NXSTAT+J
      XDWASH(K)=XL*SPACE
      L=L+1
8     CONTINUE
9     CONTINUE
      NDWASH=NXSTAT*NYSTAT
10    IF (NDWASH-150) 12,12,11
11    WRITE (6,101)
      GO TO 96
12    K=1
      DO 16 I=1,NYSTAT
      IF (SPACE) 14,13,14
13    NXSTAT=NCP(I)
14    DO 15 J=1,NXSTAT
      YDWASH(K)=YSTAT(I)
      K=K+1
15    CONTINUE
16    CONTINUE
      WRITE (6,102) NDWASH,FMACH
      BETA=SQRT(1.0-FMACH*FMACH)
      NAY3=0
      NAY4=0
      NAY5=0
      NAY6=0
      IF (NAY1.NE.0) READ (5,121) NAY3,NAY4,NAY5,NAY6
      N1=1
      N2=NCP(1)
      IF (SPACE.GE..02) N2=NXSTAT
      DO 95 IYSTAT=1,NYSTAT
      NXPTS=N2-N1+1
C
C**** CONVERT XDWASH FROM PERCENT CHORD TO X
```

```
C
      DO 17 J=2,NOLED
      IF (YSTAT(IYSTAT).LE.YLEAD(J)) GO TO 18
   17 CONTINUE
18    XLE=XLEAD(J-1)+(YSTAT(IYSTAT)-YLEAD(J-1))*AMLE(J-1)
      DO 19 J=2,NOTED
      IF (YSTAT(IYSTAT).LE.YTRAIL(J)) GO TO 20
   19 CONTINUE
20    XTE=XTRAIL(J-1)+(YSTAT(IYSTAT)-YTRAIL(J-1))*AMTE(J-1)
      CHORD=XTE-XLE
      DO 21 I=N1,N2
21    XDWASH(I)=XLE+XDWASH(I)*CHORD
      IF (NAY1.NE.0) WRITE (6,104)
      WRITE (6,103) N1,N2,YSTAT(IYSTAT)
C
C**** SET UP SPANWISE INTEGRATION INTERVALS
C
      AULT=YSTAT(IYSTAT)
      NRAS=4
      IF (AULT.LT..69) GO TO 22
      NRAS=3
      H=1.0-AULT
      GO TO 23
22    IF (AULT.GT..85) H=(1.0-AULT)/2.0
      IF (AULT.LE..85) H=.1
      IF (AULT.LT..57) NRAS=5
      IF (AULT.GT..8) NSQ(4)=10
      IF (AULT.LE..8) NSQ(4)=16
      IF (AULT.GE..57) GO TO 23
      Y(5)=AULT+H+.3
      NSQ(5)=10
      IF (AULT.GT..4) NSQ(5)=7
      IF (AULT.LE..3) NSQ(5)=16
23    Y(2)=AULT-H-.3
      Y(3)=AULT-H
      Y(4)=AULT+H
      Y(NRAS+1)=1.0
      IF (NAY3) 24,27,24
24    WRITE (6,105)
      JR2=1+NRAS
      DO 25 JR=1,JR2
      WRITE (6,106) JR,Y(JR)
25    CONTINUE
      DO 26 JR=1,NRAS
      WRITE (6,107) JR,NSQ(JR)
26    CONTINUE
C     START BIG REGION LOOP
C     CLEAR ROWS OF D MATRIX
27    DO 28 K=1,NXPTS
      DO 28 J=1,MCBS
28    FROWR(J,K)=0.0
      LAP=0
      IFL=0
      DO 90 J=1,NRAS
C     NOW SET UP SPANWISE AND CHORDWISE QUADRATURE STATIONS
C     FOR REGULAR AND SINGULAR REGIONS
      NSTAT=1
      IF (J.EQ.3) GO TO 33
C     ESTABLISH SPANWISE QUADRATURE FOR A REGULAR REGION
```

```
      FOPTS=NSQ(J)
      MNUMB=FOPTS
      IF (NAY4) 29,30,29
29    WRITE (6,108) J
      WRITE (6,109)
30    CONTINUE
      NONSNG=1
      INDEX=FOPTS
      GAUSS(1)=FOPTS
      CALL FNUD (FOPTS,GAUSS(2),GAUSS(INDEX+2))
      NCOHW=MNUMB+2
      ETAJL=Y(J)
      ETAJK=Y(J+1)
      PHIJL=ACOS(-ETAJL)
      PHIJK=ACOS(-ETAJK)
      PHI1=.5*(PHIJL+PHIJK)
      PHI2=.5*(PHIJK-PHIJL)
      DO 32 K=1,MNUMB
      PHIJ=PHI1+PHI2*GAUSS(K+1)
      ETA(K)=-COS(PHIJ)
      IF (NAY4) 31,32,31
31    WRITE (6,125) GAUSS(K+1),PHIJ,ETA(K),GAUSS(NCOWW)
      NCOWW=NCOWW+1
32    CONTINUE
      GO TO 39
C     ESTABLISH SPANWISE QUADRATURE FOR THE SINGULAR REGION
33    IF (NAY4) 34,35,34
34    WRITE (6,110)
35    CONTINUE
      MNUMB=NSQ(J)
      DEL=H/3.0
      ETA(1)=Y(J)
      ETA(2)=ETA(1)+DEL
      ETA(3)=ETA(2)+DEL
      ETA(4)=AULT
      ETA(5)=ETA(4)+DEL
      ETA(6)=ETA(5)+DEL
      ETA(7)=Y(J+1)
      IF (NAY4) 36,38,36
36    DO 37 K=1,7
      WRITE (6,111) ETA(K)
37    CONTINUE
38    NONSNG=0
39    CONTINUE
      DO 49 L=1,MNUMB
C     MNUMB = NO OF SPANWISE STATIONS IN A REGION
C     CALC. X ORDINATE AT L.E. AND T.E. FOR ATA
      ATA=ETA(L)
      K2=NOLED-1
      IF (ATA) 40,41,41
40    ATA=ABS(ATA)
41    DO 42 K=1,K2
      IF (YLEAD(K+1)-ATA) 42,43,44
42    CONTINUE
      GO TO 96
43    DLDDN(L)=XLEAD(K+1)
      GO TO 45
44    DLDDN(L)=XLEAD(K)+(XLEAD(K+1)-XLEAD(K))*(ATA-YLEAD(K))/(YLEAD(K+1)
     1-YLEAD(K))
```

```
45     K2=NOTED-1
       DO 46 K=1,K2
       IF (YTRAIL(K+1)-ATA) 46,47,48
46     CONTINUE
       GO TO 96
47     DLDDO(L)=XTRAIL(K+1)
       GO TO 49
48     DLDDO(L)=XTRAIL(K)+(XTRAIL(K+1)-XTRAIL(K))*(ATA-YTRAIL(K))/(YTRAIL
      1(K+1)-YTRAIL(K))
49     CONTINUE
       DO 89 I=N1,N2
       IX=I-N1+1
       IF (NCHORD-NFLAP) 96,83,50
50     DO 82 L=1,MNUMB
C      MNUMB=NUMBER OF SPANWISE STATIONS IN A REGION
       YO=YSTAT(IYSTAT)-ETA(L)
       COMP=ABS(BETA*S*YO)
       DLDN=(DLDDN(L)+DLDDO(L))/BOF
       DLENJ=(DLDDO(L)-DLDDN(L))/BOF
       DLDNJ=DLDN-S*XDWASH(I)
       STEVEN=DLDNJ/DLENJ
       DLFNJ=ABS(STEVEN)
       XSD=XDWASH(I)*S-DLDN
       IF (LAP) 51,52,51
51     THETFL=FLPOS(IFL)
       XFL=COS(THETFL)
       XFLAP=(DLDN-XFL*DLENJ)*FBO
52     IF (NAY4) 53,54,53
53     WRITE (6,112) L,ETA(L),YO
       BODN=FBO*DLDN
       WRITE (6,120) DLDDN(L),DLDDO(L),BODN
54     CONTINUE
       IF (DLENJ) 55,55,56
55     NSEC(L)=0
       GO TO 82
56     IF (COMP-10.0) 57,57,58
57     IF (DLFNJ-1.0) 60,58,58
58     IF (LAP) 59,67,59
59     THETA(2)=THETFL
       GO TO 66
60     IF (LAP) 61,65,61
61     IF (XDWASH(I)-XFLAP) 63,65,62
62     THETA(2)=THETFL
       THETA(3)=ACOS(STEVEN)
       GO TO 64
63     THETA(2)=ACOS(STEVEN)
       THETA(3)=THETFL
64     NQI=3
       GO TO 68
65     THETA(2)=ACOS(STEVEN)
66     NQI=2
       GO TO 69
67     NQI=1
       NQ(1)=VJ
       GO TO 70
68     NQ(3)=10
69     NQ(2)=10
       NQ(1)=10
C      NUMBER OF CHORDWISE SECTIONS, QUADRATURE POINTS, AND
```

```
C     LIMITS HAVE BEEN ESTABLISHED
70    NSEC(L)=NQI
      NOMB(L,1)=NQ(1)
      NOMB(L,2)=NQ(2)
      NOMB(L,3)=NQ(3)
      THETA(NQI+1)=PIE
      THETB(L,1)=THETA(1)
      THETB(L,2)=THETA(2)
      THETB(L,3)=THETA(3)
      THETB(L,4)=THETA(4)
      IF (NAY4) 71,72,71
71    WRITE (6,113) NQI
72    CONTINUE
C     NOW SET UP QUADRATURE POINTS AND INTEGRANDS
C     FOR CHORDWISE QUADRATURE
      DO 81 ICQ=1,NQI
      MQ=NQ(ICQ)
      IF (NAY4) 73,74,73
73    WRITE (6,114) ICQ,THETA(ICQ),THETA(ICQ+1),MQ
      WRITE (6,115)
74    CONTINUE
      NFEL=MQ+2
      FOPTS=NQ(ICQ)
      GAUFFA(1)=FOPTS
      INDEX=FOPTS
      CALL FNUD (FOPTS,GAUFFA(2),GAUFFA(INDEX+2))
      PT1=(THETA(ICQ+1)+THETA(ICQ))/2.0
      PT2=(THETA(ICQ+1)-THETA(ICQ))/2.0
      DO 80 K=1,MQ
      IF (THETA(ICQ)) 96,76,75
75    PHIJ=PT1+PT2*GAUFFA(K+1)
      GO TO 77
76    PHIJ=PT1*(1.0+GAUFFA(K+1))
77    XO=XSD+DLENJ*COS(PHIJ)
      FKER=FKERNL(XO,YO,S,FMACH)
      THETAA(NSTAT,L)=PHIJ
      FORR(NSTAT,L)=FKER*GAUFFA(NFEL)*SIN(PHIJ)
      IF (NAY4) 78,79,78
78    WRITE (6,116) GAUFFA(K+1),GAUFFA(NFEL),PHIJ,XO,FKER,FORR(NSTAT,L)
79    CONTINUE
      NFEL=NFEL+1
      NSTAT=NSTAT+1
80    CONTINUE
81    CONTINUE
      NSTAT=1
82    CONTINUE
      CALL MATROW (MSPAN,NCHORD,NONSNG,H,I,NAY5,NEED,NFLAP,PHIJK,PHIJL,
     1LAP,IFL,IX,FROWR)
83    IF (NFLAP) 87,87,84
84    LAP=1
      IF (IFL-NFLAP) 85,86,96
85    IFL=IFL+1
      GO TO 50
86    IFL=0
      LAP=0
87    IF (NAY6) 88,89,88
88    WRITE (6,117) (FROWR(ND,IX),ND=1,MCBS)
89    CONTINUE
90    CONTINUE
```

```
C     MATRIX ROWS FOR ALL CONTROL POINTS ON A CHORD ARE COMPLETED
      DO 94 IX=1,NXPTS
      WRITE (3) (FROWR(ND,IX),ND=1,MCBS)
      IF (NODE3) 91,92,91
91    WRITE (7,118) (FROWR(ND,IX),ND=1,MCBS)
92    IF (NAY6) 93,94,93
93    WRITE (6,119) (FROWR(ND,IX),ND=1,MCBS)
94    CONTINUE
      IF (IYSTAT.EQ.NYSTAT) GO TO 95
      N1=N2+1
      IF (SPACE.LT..02) N2=N2+NCP(IYSTAT+1)
      IF (SPACE.GE..02) N2=N2+NXSTAT
95    CONTINUE
C     ALL MATRIX ROW CALCULATED
C     GO TO MATRIX PRINT SUBPROGRAM
      IF (NODE1.NE.0) CALL MPRINT (TITLE,6,3,CDWASH,MCBS)
      RETURN
96    STOP
C
97     FORMAT (26H1NO. OF SPANWISE MODES  = I3/26H0NO. OF CHORDWISE MODES
     1 = I3/26H0NO. OF  FLAP MODES      = I3/26H0COTANGENT MODE,   NEED =
     2 I3)
98     FORMAT (17H0POSITION OF FLAPI3,3H = F8.6)
99     FORMAT (12F6.0)
100    FORMAT (25H0THIS OPTION DISCONTINUED)
101    FORMAT (1H150HNUMBER OF DOWNWASH CONTROL POINTS GREATER THAN 150)
102    FORMAT (1H119XI4,1X23HDOWNWASH CONTROL POINTS,5X,9HMACH NO.=E14.8)
103    FORMAT (24H0DOWNWASH CONTROL POINTSI4,5H   TOI4,5X2HY=E15.8)
104    FORMAT (1H1)
105    FORMAT (75H0SPANWISE QUADRATURE INTERVALS AND NUMBER OF QUADRATURE
     1 POINTS PER INTERVAL)
106    FORMAT (3H0Y(I2,4H) = F10.7)
107    FORMAT (5H0NSQ(I2,4H) = I3)
108    FORMAT (1H115X,15HREGULAR REGION I2,12H INTEGRATION)
109    FORMAT (46H0STATIONS AND WEIGHTS FOR SPANWISE INTEGRATION/1H )
110    FORMAT (1H115X,27HSINGULAR REGION INTEGRATION/33H0SPANWISE STATION
     1S FOR QUADRATURE)
111    FORMAT (6H0ETA= E15.8)
112    FORMAT (48H1STATIONS, WEIGHTS, AND INTEGRANDS FOR CHORDWISE/32H QU
     1ADRATURE AT SPANWISE STATION,I5/6H0ETA= E15.8,5X,4HY0= E15.8/1H0)
113    FORMAT (30H0NO. OF CHORDWISE INTERVALS = I3)
114    FORMAT (24H0CHORDWISE INTERVAL NO. I3/13H LIMITS FROM F11.8,5X,3HT
     10 F11.8,8H RADIANS/28H NO. OF QUADRATURE POINTS = I3)
115    FORMAT (1H0,8X,10HGAUSS STA.,10X,9HGAUSS WT.,13X,5HTHETA,16X,2HX0,
     116X,6HKERNEL,13X,9HGAUSS FN./1H0)
116    FORMAT (6E20.8)
117    FORMAT (1H010X,39HPARTIAL ACCUMULATED SUM OF ROW ELEMENTS/1H0
     16E20.8/(1H 6E20.8))
118    FORMAT (1P5E14.7)
119    FORMAT (1H010X,13HCOMPLETED ROW/1H /(1H 6E20.8))
120    FORMAT (25H0LEADING EDGE AT ETA, X= F9.6/26H TRAILING EDGE AT ETA,
     1 X= F9.6/22H MID-CHORD AT ETA, X= F9.6/1H0)
121    FORMAT (14I5)
122    FORMAT (7F10.0)
123    FORMAT (12A6)
124    FORMAT (1H154X,11HCHAIN (1,8)/50H0CALCULATION OF DOWNWASH CONTROL
     1POINT MATRIX FOR ,12A6)
125    FORMAT (1H010X7HGAUSS= F14.8,2X6HPHIJ= F14.8,2X,5HETA= F14.8,2X4HW
     1T= F14.8)
```

```
      END

      SUBROUTINE CHAIN6
C
C     THIS LINK CALCULATES THE LEAST SQUARES INVERSE OF D
C     D MATRIX IS ON TAPE 3 OR READ FROM CARDS
C     INVERSE IS STORED ON TAPE 2, POSITION ZERO
C
      DIMENSION ARRAY(12),TITLE(9)
C
      READ (5,6) ARRAY
      READ (5,5) NROW,NCOL,NODE3,NODE5,NODE6,NAY
      WRITE (6,7) ARRAY
      CALL PINVRS(3,2,NAY,NODE3,NODE6,NROW,NCOL)
      IF (NODE5) 3,4,3
      DATA Q000HL/6HINVERS/
3     TITLE(1)=Q000HL
      DATA Q001HL/6HE OF D/
      TITLE(2)=Q001HL
      DATA Q002HL/6HDOWNWAS/
      TITLE(3)=Q002HL
      DATA Q003HL/6HH CONT/
      TITLE(4)=Q003HL
      DATA Q004HL/6HRUL PO/
      TITLE(5)=Q004HL
      DATA Q005HL/6HINT MA/
      TITLE(6)=Q005HL
      DATA Q006HL/6HTRIX  /
      TITLE(7)=Q006HL
      CALL MPRINT (TITLE,7,2,NCOL,NROW)
4     RETURN
C
5     FORMAT (10I5)
6     FORMAT (12A6)
7     FORMAT (1H150X,11HCHAIN (6,8)/42H0INVERT DOWNWASH CONTROL POINT MA
     1TRIX FOR ,12A6)
      END

      SUBROUTINE CHAIN7
C
C     CALCULATES PRESSURE DISTRIBUTION
C
      DIMENSION W(1,150),ANM(1,75),ETA(50),CNP(75),CLNP(75),GEE(75),BEN(
     150),ARM(50),CLLOC(20),CMLOC(20),ALLOC(20),CDLOC(20),EEDEL(10),
     2EPSLN(10),CK(6,10),CA(12),CKA(12),DINVRS(1,150),CEE(150,36),P(1,
     3150),CHORD(51),WHY(51),FTHETA(20),PSI(50),CP(50,50),DELTA(51),A(50
     4),B(50),C(50),D(50),ALFA(20),DELFL(10),WW(1,150),FLPOS(10),BETA(20
     5),YP(20),NXDP(20),ARRAY(12)
C
      COMMON W,ANM,ETA,CNP,CLNP,GEE,BEN,ARM,CLLOC,CMLOC,ALLOC,CDLOC,
     1EEDEL,EPSLN,CK,CA,CKA,CL,CM,CDL,N,M,NU,NON,NFLAP,PI,PLBA,NETA,BO,
     2BA,BBAR,PIRC,NPSI
C
      READ (5,166) ARRAY
      READ (5,164) N,M,NYP,NROWS,NETA,NCHORD,NFLAP,NAY,NPSI
      READ (5,164) NALFA,NBETA,NEED,NODE6,NODE7,NW
      READ (5,165) BO,SPACE,YF,DPSI
```

```
      READ (5,167) (YP(I),I=1,NYP)
      READ (5,167) (ETA(I),I=1,NETA)
      READ (5,167) (BETA(I),I=1,NBETA)
      READ (5,167) (ALFA(I),I=1,NALFA)
      READ (5,167) (FLPOS(I),I=1,NFLAP)
      READ (5,167) (CHORD(I),I=1,NCHORD)
      READ (5,167) (WHY(I),I=1,NCHORD)
      READ (5,167) (DELTA(I),I=1,NCHORD)
      WRITE (6,168) ARRAY
      IF (YF) 2,3,2
2     WRITE (6,162) YF
      GO TO 4
3     WRITE (6,163)
4     CONTINUE
      IF (SPACE) 5,6,5
5     NXDDP=NROWS/NYP
      GO TO 7
6     READ (5,169) (NXDP(I),I=1,NYP)
7     NCN=N*M
      RAD=57.29578
      PI=3.14159265
      IF (NFLAP) 158,13,8
8     DO 12 I=1,NFLAP
      DELFL(I)=DELFL(I)/RAD
      IF (FLPOS(I)-0.5) 10,9,11
9     FLPOS(I)=0.5*PI
      GO TO 12
10    FLPOS(I)=ACOS(1.0-2.0*FLPOS(I))
      GO TO 12
11    FLPOS(I)=0.5*PI+ASIN(2.0*FLPOS(I)-1.0)
12    CONTINUE
C     CALCULATE CO-ORDINATES OF PRESSURE POINTS
13    IF (DPSI) 14,16,15
14    READ (5,167) (PSI(I),I=1,NPSI)
      GO TO 19
15    NPSI=1.0/DPSI
      IF (50-NPSI) 16,17,17
16    WRITE (6,171)
      GO TO 159
17    J=1
18    XJ=J
      PSI(J)=XJ*DPSI
      J=J+1
      IF (J-NPSI) 18,18,19
C     NOW CALCULATE ELEMENTS OF C MATRIX
19    I=1
20    ETTA=ETA(I)
      ROOT=SQRT(1.0-ETTA**2)
      IF (NCHORD-1) 158,21,22
21    CC=CHORD(1)
      GO TO 27
22    NESS=2
23    IF (ETTA-WHY(NESS)) 26,25,24
24    NESS=NESS+1
      GO TO 23
25    CC=CHORD(NESS)
      GO TO 27
26    CC=CHORD(NESS-1)-(CHORD(NESS-1)-CHORD(NESS))*(ETTA-WHY(NESS-1))/(
     1WHY(NESS)-WHY(NESS-1))
```

```
27    PIRC=(16.0*PI*ROOT)/CC
      J=1
28    PSII=PSI(J)
      KR=(I-1)*NPSI+J
      IF (PSII-0.5) 30,29,31
29    THETA=PI/2.0
      GO TO 32
30    THETA=ACOS(1.0-2.0*PSII)
      GO TO 32
31    THETA=PI/2.0+ASIN(2.0*PSII-1.0)
32    NU=N-NFLAP
      IF (NEED) 33,34,33
33    N1=2
      NX=0
      GO TO 35
34    N1=1
      NX=1
      GO TO 36
35    FTHETA(1)=COS(THETA/2.0)/SIN(THETA/2.0)
36    DO 37 NN=N1,NU
      ANN=NN-1+NX
      FTHETA(NN)=(4.0*SIN(ANN*THETA))/2.0**(ANN*2.0)
37    CONTINUE
      IF (NFLAP) 158,40,38
38    NUU=NU+1
      NFR=1
      DO 39 NN=NUU,N
      AUX=SIN((FLPOS(NFR)+THETA)/2.0)
      AUY=SIN((FLPOS(NFR)-THETA)/2.0)
      AUXY=ABS(AUX/AUY)
      FTHETA(NN)=(ALOG(AUXY))/PI
      NFR=NFR+1
39    CONTINUE
40    EMM=M
      K=1
      NN=1
41    EM=0.0
      IF (ETTA) 158,42,43
42    ETEM=1.0
      GO TO 44
43    ETEM=ETTA**EM
44    CEE(KR,K)=PIRC*FTHETA(NN)*ETEM
      EM=EM+2.0
      K=K+1
      IF (EM/2.0+1.0-EMM) 43,43,45
45    NN=NN+1
      IF (NN-N) 41,41,46
46    J=J+1
      IF (J-NPSI) 28,28,47
47    I=I+1
      IF (I-NETA) 20,20,48
48    NPOINT=NPSI*NETA
      REWIND 2
      IF (NODE6) 49,51,49
49    DO 50 I=1,NON
      READ (5,170) (DINVRS(1,J),J=1,NROWS)
      WRITE (2) (DINVRS(1,J),J=1,NROWS)
50    CONTINUE
      REWIND 2
```

```
51      IF (NAY) 52,55,52
C       PRINT C AND D MATRICES
52      WRITE (6,172)
        DO 53 I=1,NON
        READ (2) (DINVRS(1,J),J=1,NROWS)
        WRITE (6,173) (DINVRS(1,J),J=1,NROWS)
53      CONTINUE
        REWIND 2
        WRITE (6,174)
        DO 54 I=1,NPOINT
        WRITE (6,173) (CEE(I,K),K=1,NON)
54      CONTINUE
55      NI=NCHORD-1
C       NORMALIZE X DIRECTION
        DO 56 I=1,NCHORD
        DELTA(I)=DELTA(I)/BO
56      CONTINUE
C       CALCULATE A AND FOR WING REGIONS
        DO 57 I=1,NI
        ETAA=WHY(I+1)-WHY(I)
        B(I)=0.5*(CHORD(I+1)-CHORD(I))/ETAA
        IF (ABS(B(I))-1.0E-05) 201,201,202
 201    B(I) = 0.0
 202    CONTINUE
        A(I)=0.5*CHORD(I)-B(I)*WHY(I)
57      CONTINUE
C       NOW CALCULATE AVERAGE AND MEAN CHORDS
        BA=0.0
        BAR=0.0
        DO 58 I=1,NI
        BA=BA+A(I)*(WHY(I+1)-WHY(I))+0.5*B(I)*(WHY(I+1)**2-WHY(I)**2)
        BAR=BAR+(A(I)**2)*(WHY(I+1)-WHY(I))+A(I)*B(I)*(WHY(I+1)**2-WHY(I)*
       1*2)+(B(I)**2)*(WHY(I+1)**3-WHY(I)**3)/3.0
58      CONTINUE
        CHA=2.0*BA
        BBAR=BAR/BA
        CBAR=2.0*BBAR
C       CALCULATE LOCATION OF MEAN CHORD AND MOMENT AXIS
        I=1
59      IF (CBAR-CHORD(I+1)) 60,61,61
  60    IF (I+1-NCHORD) 200,61,61
 200    I = I+1
        GO TO 59
  61    CONTINUE
        IF (B(I)) 203,204,203
 204    YBAR = 0.0
        GO TO 205
 203    YBAR = (BBAR-A(I))/B(I)
 205    CONTINUE
        PSIO=DELTA(I)+(DELTA(I+1)-DELTA(I))*(YBAR-WHY(I))/(WHY(I+1)-WHY(I)
       1)+BBAR/(2.0*BO)
        PSIOBO=PSIO*BO
C       NOW CALCULATE C AND D FOR REGIONS
        DO 62 I=1,NI
        ETAA=WHY(I+1)-WHY(I)
        D(I)=(DELTA(I+1)-DELTA(I))/ETAA
        C(I)=DELTA(I)-PSIO-D(I)*WHY(I)
62      CONTINUE
C       CALCULATE LOCAL MOMENT ARMS AND SEMICHORDS
```

```
      I=1
63    J=2
64    IF (ETA(I)-WHY(J)) 66,66,65
65    J=J+1
      GO TO 64
66    J1=J-1
      BEN(I)=A(J1)+B(J1)*ETA(I)
      ARM(I)=C(J1)+D(J1)*ETA(I)
      I=I+1
      IF (NETA-I) 67,63,63
67    WRITE (6,175) CHA,CBAR,PSIOBO,YBAR
      CON=(PI**2)/(BA*BBAR)
      DO 68 I=1,75
      CNP(I)=0.0
68    CLNP(I)=0.0
      L=0
      IF (NEED) 69,73,69
69    L=L+1
      MM=1
70    DO 71 I=1,NI
      ETAO=WHY(I)
      ETA1=WHY(I+1)
      MP=2*(MM-1)
      RMI=FRMI(ETAO,ETA1,MP)
      PMI=FPMI(ETAO,ETA1,MP)
      CNP(L)=CNP(L)+((A(I)+2.0*BO*C(I))*RMI+(B(I)+2.0*BO*D(I))*PMI)*CON
71    CONTINUE
      MM=MM+1
      IF (MM-M) 72,72,73
72    L=L+1
      GO TO 70
73    IF (NU-1) 158,74,75
74    IF (NEED) 85,75,85
75    L=L+1
      MM=1
76    DO 77 I=1,NI
      ETAO=WHY(I)
      ETA1=WHY(I+1)
      MP=2*(MM-1)
      RMI=FRMI(ETAO,ETA1,MP)
      PMI=FPMI(ETAO,ETA1,MP)
      CNP(L)=CNP(L)+((A(I)+BO*C(I))*RMI+(B(I)+BO*D(I))*PMI)*CON
77    CONTINUE
      MM=MM+1
      IF (MM-M) 78,78,79
78    L=L+1
      GO TO 76
79    IF (NU-2) 85,80,81
80    IF (NEED) 85,81,85
81    L=L+1
      MM=1
82    DO 83 I=1,NI
      ETAO=WHY(I)
      ETA1=WHY(I+1)
      MP=2*(MM-1)
      RMI=FRMI(ETAO,ETA1,MP)
      PMI=FPMI(ETAO,ETA1,MP)
      CNP(L)=CNP(L)-0.125*(A(I)*RMI+B(I)*PMI)*CON
83    CONTINUE
```

```
      MM=MM+1
      IF (MM-M) 84,84,85
 84   L=L+1
      GO TO 82
 85   IF (NFLAP) 158,92,86
 86   DO 87 I=1,NFLAP
      SN=SIN(FLPOS(I))
      CSN=COS(FLPOS(I))
      EPSLN(I)=SN
      EEDEL(I)=SN*(1.0-.5*CSN)
 87   CONTINUE
      L1=L+1
      L2=NU*M
      DO 88 L=L1,L2
      CNP(L)=0.0
 88   CONTINUE
      L=L2
      DO 91 IR=1,NFLAP
      DO 90 MM=1,M
      L=L+1
      CNP(L)=0.0
      MP=2*(MM-1)
      DO 89 I=1,NI
      ETA0=WHY(I)
      ETA1=WHY(I+1)
      RMI=FRMI(ETA0,ETA1,MP)
      PMI=FPMI(ETA0,ETA1,MP)
      CNP(L)=CNP(L)+(2.0*CON/PI)*((EEDEL(IR)*A(I)+BO*EPSLN(IR)*C(I))*RMI
     1+(EEDEL(IR)*B(I)+BO*EPSLN(IR)*D(I))*PMI)
 89   CONTINUE
 90   CONTINUE
 91   CONTINUE
C     CNP COEFFICIENTS HAVE BEEN CALCULATED FOR MOMENT EQN
C     NOW CALCULATE COEFFICIENTS OF LIFT EQN - CLNP
 92   CONST=(PI**3)/(4.0*BA)
      L=0
      IF (NEED) 93,98,93
 93   L=L+1
      CLNP(L)=4.0*CONST
      IF (M-1) 98,98,94
 94   L=L+1
      CLNP(L)=CONST
      IF (M-2) 98,98,95
 95   L=L+1
      CLNP(L)=0.5*CONST
      IF (M-3) 98,98,96
 96   DO 97 MM=4,M
      L=L+1
      PM=2*(MM-1)
      CLNP(L)=(PM-1.0)*CLNP(L-1)/(PM+2.0)
 97   CONTINUE
 98   IF (NU-1) 158,99,100
 99   IF (NEED) 105,100,105
 100  L=L+1
      CLNP(L)=2.0*CONST
      IF (M-1) 105,105,101
 101  L=L+1
      CLNP(L)=0.5*CONST
      IF (M-2) 105,105,102
```

```
102     L=L+1
        CLNP(L)=0.5*0.5*CONST
        IF (M-3) 105,105,103
103     DO 104 MM=4,M
        L=L+1
        PM=2*(MM-1)
        CLNP(L)=(PM-1.0)*CLNP(L-1)/(PM+2.0)
104     CONTINUE
105     IF (NFLAP) 158,113,106
106     L1=L+1
        DO 107 L=L1,L2
        CLNP(L)=0.0
107     CONTINUE
        L=L2
        COST=CONST/PI
        DO 112 IR=1,NFLAP
        EPSLON=EPSLN(IR)
        L=L+1
        CLNP(L)=4.0*COST*EPSLON
        IF (M-1) 158,112,108
108     L=L+1
        CLNP(L)=COST*EPSLON
        IF (M-2) 112,112,109
109     L=L+1
        CLNP(L)=0.5*COST*EPSLON
        IF (M-3) 112,112,110
110     DO 111 MM=4,M
        L=L+1
        PM=2*(MM-1)
        CLNP(L)=(PM-1.0)*CLNP(L-1)/(PM+2.0)
111     CONTINUE
112     CONTINUE
C       CLNP HAVE BEEN CALCULATED - NOW PRINT COEFFS
113     IF (NAY) 114,115,114
114     WRITE (6,176)
        WRITE (6,173) (CLNP(L),L=1,NON)
        WRITE (6,177)
        WRITE (6,173) (CNP(L),L=1,NON)
C       SET UP A TABLE OF GEE FOR CD CALCULATION
115     PLBA=(2.0*PI**5)/BA
        GEE(1)=0.5
        GEE(2)=0.125
        J=4*(M-1)
        IF (2-J) 116,126,126
116     DO 117 JJ=4,J,2
        JJJ=(JJ+2)/2
        EJJ=JJ
        COE=(EJJ-1.0)/(EJJ+2.0)
        GEE(JJJ)=COE*GEE(JJJ-1)
117     CONTINUE
C       START CAMBER LOOP
        DO 157 IW=1,NW
        IF (NODE7) 118,123,118
118     IW1=1
        DO 122 IY=1,NYP
        IF (SPACE) 120,119,120
119     IW2=NXDP(IY)+IW1-1
        GO TO 121
120     IW2=NXDDP+IW1-1
```

```
121   READ (5,170) (W(1,IWX),IWX=IW1,IW2)
      IW1=IW2+1
122   CONTINUE
      GO TO 124
 123  CONTINUE
C123  CALL CAMBER (NXDP,NEED,SPACE,NYP)
C     THIS SUBROUTINE WILL CALCULATE W MATRIX
124   WRITE (6,178) IW
      WRITE (6,179)
      WRITE (6,173) (W(1,I),I=1,NROWS)
      DO 125 KW=1,NROWS
      W(1,KW)=ATAN(W(1,KW))
125   CONTINUE
      WRITE (6,180)
      WRITE (6,173) (W(1,I),I=1,NROWS)
C     START BETA LOOP - (INCIDENCE ANGLES)
126   DO 156 KK=1,NBETA
C     NOW START ALFA LOOP
      DO 155 K=1,NALFA
      RALFA=ALFA(K)/RAD
      ANGLE=BETA(KK)+ALFA(K)
      RANGLE=ANGLE/RAD
      IF (YF) 158,127,129
127   DO 128 I=1,NROWS
      ARG=W(1,I)-RANGLE
      WW(1,I)=SIN(ARG)/COS(ARG)
128   CONTINUE
      WRITE (6,181) BETA(KK),ALFA(K)
      WRITE (6,173) (WW(1,J),J=1,NROWS)
      GO TO 138
129   SYL=SIN(2.0*RALFA)/2.0
      L=1
      DO 137 I=1,NYP
      IF (YP(I)-YF) 130,131,131
130   ATSLP=0.0
      GO TO 132
131   SLOOP=SYL*(YF/YP(I))**2
      ATSLP=ATAN(SLOOP)
132   IF (SPACE) 133,134,133
133   NXP=NXDDP
      GO TO 135
134   NXP=NXDP(I)
135   DO 136 J=1,NXP
      ARG=W(1,L)-RANGLE-ATSLP
      WW(1,L)=SIN(ARG)/COS(ARG)
      L=L+1
136   CONTINUE
137   CONTINUE
      WRITE (6,182)
      WRITE (6,173) (WW(1,J),J=1,NROWS)
138   DO 139 I=1,75
      ANM(1,I)=0.0
139   CONTINUE
      DO 140 I=1,150
      P(1,I)=0.0
140   CONTINUE
C     NOW CALCULATE A MATRIX
      DO 142 I=1,NON
      READ (2) (DINVRS(1,J),J=1,NROWS)
```

```
      DO 141 J=1,NROWS
      ANM(1,I)=ANM(1,I)+DINVRS(1,J)*WW(1,J)
141   CONTINUE
142   CONTINUE
      REWIND 2
      IF (NAY) 143,144,143
143   WRITE (6,183)
      WRITE (6,173) (ANM(1,I),I=1,NON)
C     NOW CALCULATE P MATRIX
144   DO 146 I=1,NPOINT
      DO 145 J=1,NON
      P(1,I)=P(1,I)+CEE(I,J)*ANM(1,J)
145   CONTINUE
146   CONTINUE
C     NOW STORE P IN A TWO DIMENSIONAL ARRAY
      DO 147 L=1,NPOINT
      I=(L-1)/NPSI+1
      J=L-(I-1)*NPSI
      CP(I,J)=P(1,L)
147   CONTINUE
      CALL AERO (NEED)
C     NOW PRINT CL, CM AND PRESSURE DISTRIBUTION
      WRITE (6,184) ALFA(K),BETA(KK)
      WRITE (6,185) CL,CM,CDL
      L=1
148   WRITE (6,186)
      IF (NETA-11*L) 149,149,150
149   NCOL1=1+(L-1)*11
      NCOL2=NETA
      GO TO 151
150   NCOL1=1+(L-1)*11
      NCOL2=L*11
151   WRITE (6,187) (ETA(I),I=NCOL1,NCOL2)
      WRITE (6,188)
      DO 152 J=1,NPSI
      WRITE (6,194) PSI(J),(CP(I,J),I=NCOL1,NCOL2)
152   CONTINUE
      WRITE (6,189)
      WRITE (6,193) (BEN(I),I=NCOL1,NCOL2)
      WRITE (6,190)
      WRITE (6,193) (CLLOC(I),I=NCOL1,NCOL2)
      WRITE (6,192)
      WRITE (6,193) (CMLOC(I),I=NCOL1,NCOL2)
      WRITE (6,160)
      WRITE (6,193) (CDLOC(I),I=NCOL1,NCOL2)
      IF (NAY) 206,207,206
206   WRITE (6,161)
      WRITE (6,193) (ALLOC(I),I=NCOL1,NCOL2)
      DO 153 JC=1,N
      WRITE (6,191) JC,(CK(JC,I),I=NCOL1,NCOL2)
153   CONTINUE
207   CONTINUE
      IF (NETA-11*L) 155,155,154
154   L=L+1
      GO TO 148
C     NOW CONSIDER NEXT ALFA
155   CONTINUE
156   CONTINUE
157   CONTINUE
```

```
      GO TO 159
158   WRITE (6,195)
 159  RETURN
C
160   FORMAT (1H0,20X,10HCD*C/CAVE )
161   FORMAT (1H0,20X,23HALPHA INDUCED (DEGREES))
162   FORMAT (1H0/24H FUSELAGE EDGE AT ETA = F5.4)
163   FORMAT (1H0/8H NO BODY)
164   FORMAT (10I5)
165   FORMAT (4F10.0)
166   FORMAT (12A6)
167   FORMAT (10F7.0)
168   FORMAT (1H154X,11HCHAIN (7,8)/50H0CALCULATION OF PRESSURE LOADING
     1DISTRIBUTION FOR ,12A6)
169   FORMAT (20I2)
 170  FORMAT(5E14.7)
171   FORMAT (1H110X,26H ERROR-FLAG LESS THAN 0.02)
172   FORMAT (1H120X,43HINVERSE OF DOWNWASH CONTROL POINT MATRIX, D)
173   FORMAT (1H06E20.8/(1H 6E20.8))
174   FORMAT (1H120X,32HPRESSURE CONTROL POINT MATRIX, C)
175   FORMAT (1H010X,20HGEOMETRIC PARAMETERS/1H022HAVERAGE CHORD, CAVE =
     1 F10.6/1H031HMEAN AERODYNAMIC CHORD, CBAR = F10.6/1H029HLOCATION O
     2F 1/4 CBAR, XBAR = F10.6/1H034HSPANWISE LOCATION OF CBAR, YBAR =
     3F10.6)
176   FORMAT (1H110X,27HCOEFFICIENTS OF CL EQUATION)
177   FORMAT (1H0/1H010X,27HCOEFFICIENTS OF CM EQUATION)
178   FORMAT (1H131X,20HCAMBER SHAPE NUMBER ,I2)
179   FORMAT (1H025X,46HSPECIFIED DOWNWASH OR SLOPE (DZ/DX) MATRIX,  W)
180   FORMAT (1H0/40H0SPECIFIED SLOPE DISTRIBUTION IN RADIANS)
181   FORMAT (1H110X,21HW MATRIX WITH BETA = F9.4,12H AND ALFA = F9.4)
182   FORMAT (1H110X,48HTOTAL DOWNWASH MATRIX - INCLUDES THE BODY EFFECT
     1)
183   FORMAT (1H0/1H010X,58HA MATRIX, I.E. COEFFICIENTS OF THE PRESSURE
     1LOADING SERIES)
184   FORMAT (1H110X,18HRESULTS FOR ALFA= F9.4,15H, AND EPSILON= F9.4,9H
     1  DEGREES)
185   FORMAT (1H023HLIFT COEFFICIENT, CL = F10.5/1H025HMOMENT COEFFICIEN
     1T, CM = F10.5/1H032HINDUCED DRAG COEFFICIENT, CDI = F10.5)
186   FORMAT (1H020X,33HPRESSURE LOADING DISTRIBUTION, PR)
187   FORMAT (1H06HSPAN =,11F10.4)
188   FORMAT (9H0FRACTION/9H OF CHORD)
189   FORMAT (1H020X,20HLOCAL SEMICHORD, C/2)
190   FORMAT (1H020X,9HCL C/CAVE)
191   FORMAT (2H0KI1,1H ,1P7E15.7/(4H    1P7E15.7))
192   FORMAT (1H020X,17HCM C**2/CAVE CBAR)
193   FORMAT (1H06X,11F10.4)
194   FORMAT (1H F6.4,11F10.4)
195   FORMAT (1H113HERROR IN DATA)
      END


      SUBROUTINE AERO (NEED)
C
      DIMENSION W(1,150),ANM(1,75),ETA(50),CNP(75),CLNP(75),GEE(75),
     1BEN(50),ARM(50),CLLOC(20),CMLOC(20),ALLOC(20),CDLOC(20),EEDEL(10),
     2EPSLN(10),CK(6,10),CA(12),CKA(12)
C
      COMMON W,ANM,ETA,CNP,CLNP,GEE,BEN,ARM,CLLOC,CMLOC,ALLOC,CDLOC,
     1EEDEL,EPSLN,CK,CA,CKA,CL,CM,CDL,N,M,NU,NON,NFLAP,PI,PLBA,NETA,BO,
```

```
     2BA,BBAR,PIRC,NPSI
C
C      NOW CALCULATE CL AND CM
C
       CL=0.0
       DO 1 I=1,NON
       CL=CL+CLNP(I)*ANM(1,I)
1      CONTINUE
       CM=0.0
       DO 2 I=1,NON
       CM=CM+CNP(I)*ANM(1,I)
2      CONTINUE
       CM=-CM
C
C      CALCULATE INDUCED DRAG
C
       SUM=0.0
       DO 16 IS=1,M
       IM=2*(IS-1)
       DO 15 L=1,IS
       IK=2*(L-1)
       SQM=FSQM(IM,IK)
       DO 14 IR=1,M
       IP=2*(IR-1)
       MRP=(IM-IK+IP+2)/2
       AMP=0.0
       NCA=NFLAP+2
       IF (NEED) 5,3,5
3      CA(1)=0.0
       CKA(1)=0.0
       IF (NU) 54,8,4
4      CA(2)=0.5*ANM(1,IS)
       CKA(2)=0.5*ANM(1,IR)
       GO TO 8
5      CA(1)=ANM(1,IS)
       CKA(1)=ANM(1,IR)
       IF (NU-1) 6,6,7
6      CA(2)=0.0
       CKA(2)=0.0
       GO TO 8
7      MIR=M+IR
       MIS=M+IS
       CA(2)=0.5*ANM(1,MIS)
       CKA(2)=0.5*ANM(1,MIR)
8      IF (NFLAP) 54,11,9
9      DO 10 IFL=1,NFLAP
       MFL=(NU-1+IFL)*M
       MFR=MFL+IR
       MFS=MFL+IS
       CA(IFL+2)=EPSLN(IFL)*ANM(1,MFS)/PI
       CKA(IFL+2)=EPSLN(IFL)*ANM(1,MFR)/PI
10     CONTINUE
11     DO 13 IFL=1,NCA
       CIFL=CA(IFL)
       DO 12 IML=1,NCA
       AMP=AMP+CIFL*CKA(IML)
12     CONTINUE
13     CONTINUE
       SUM=SUM+AMP*GEE(MRP)*SQM
```

```
14    CONTINUE
15    CONTINUE
16    CONTINUE
      CDL=PLBA*SUM
C
C     NOW CALCULATE LOCAL LIFT AND MOMENT COEFFICIENTS
C
      CO=4.0*(PI**2)
      COO=PI**2
      DO 43 I=1,NETA
      ROOT=SQRT(1.0-ETA(I)**2)
      SERES1=0.0
      VERES=0.0
      SERS=0.0
      DO 42 J=1,M
      SERES=0.0
      LP=2*(J-1)
      IF (LP) 54,17,19
17    IF (ETA(I)) 54,18,19
18    ETTA=1.0
      GO TO 20
19    ETTA=ETA(I)**LP
20    IF (NU) 54,27,21
21    IF (NEED) 24,22,24
22    MJ=M+J
      SERES=SERES+0.5*ANM(1,J)
      SERS=SERS+(BEN(I)+BO*ARM(I))*ANM(1,J)*ETTA
      IF (NU-1) 27,27,23
23    SERS=SERS-0.125*BEN(I)*ANM(1,MJ)*ETTA
      GO TO 27
24    MJ=M+J
      MMJ=M+M+J
      SERES=SERES+ANM(1,J)
      SERS=SERS+(BEN(I)+2.0*BO*ARM(I))*ANM(1,J)*ETTA
      IF (NU-1) 27,27,25
25    SERES=SERES+0.5*ANM(1,MJ)
      SERS=SERS+(BEN(I)+BO*ARM(I))*ANM(1,MJ)*ETTA
      IF (NU-2) 27,27,26
26    SERS=SERS-0.125*BEN(I)*ANM(1,MMJ)*ETTA
27    IF (NFLAP) 28,30,28
28    ETPI=ETTA/PI
      DO 29 IFL=1,NFLAP
      MFL=(NU+IFL-1)*M
      MIP=MFL+J
      SERS=SERS+2.0*ETPI*(BEN(I)*EEDEL(IFL)+BO*ARM(I)*EPSLN(IFL))*ANM(1,
     1MIP)
      SERES=SERES+EPSLN(IFL)*ANM(1,MIP)/PI
29    CONTINUE
30    AYE1=0.0
      DO 41 NG=1,J
      NGM=2*(NG-1)
      IF (ETA(I)) 32,31,32
31    ETAG=1.0
      GO TO 33
32    ETAG=ETA(I)**(LP-NGM)
33    IF (NG-2) 34,35,36
34    AYE=LP+1
      GO TO 40
35    AYE=1-LP
```

```
      AYE=0.5*AYE
      GO TO 40
36    NUM=1
      LOW=2
      IF (NGM-4) 40,39,37
37    IG2=NGM-2
      DO 38 IG=4,IG2,2
      NUM=NUM*(IG-1)
38    LOW=LOW*IG
39    UNM=NUM*(NGM-LP-1)
      ELW=LOW*NGM
      AYE=UNM/ELW
40    AYE1=AYE1+AYE*ETAG
41    CONTINUE
      VERES=VERES+SERES*AYE1
      SERES1=SERES1+SERES*ETTA
42    CONTINUE
      CLLOC(I)=CO*ROOT*SERES1/BA
      ALLOC(I)=COO*VERES
      CDLOC(I)=CLLOC(I)*ALLOC(I)
      ALLOC(I)=180.0*ALLOC(I)/PI
      CMLOC(I)=-COO*ROOT*SERS/(BA*BBAR)
43    CONTINUE
C
C     CALCULATE CK(N,ETA)
C
      DO 53 IT=1,NETA
      ETTA=ETA(IT)
      PIRC=8.0*PI*SQRT(1.0-ETTA*ETTA)/BEN(IT)
      DO 52 JC=1,N
      IF (JC-1) 45,44,45
44    EL=1.0
      GO TO 48
45    IF (JC-NU) 46,46,47
46    EL=4.0/(2.0**(2*JC-2))
      GO TO 48
47    EL=1.0/PI
48    SIGMA=0.0
      NEL=(JC-1)*M
      DO 51 JS=1,M
      MEL=NEL+JS
      IF (JS-1) 50,49,50
49    SIGMA=SIGMA+ANM(1,MEL)
      GO TO 51
50    SIGMA=SIGMA+ANM(1,MEL)*ETTA**(2*(JS-1))
51    CONTINUE
      CK(JC,IT)=SIGMA*EL*PIRC
52    CONTINUE
53    CONTINUE
      GO TO 55
54    WRITE (6,56)
55    RETURN
C
56    FORMAT (1H113HERROR IN DATA)
      END

      SUBROUTINE PINVRS(NIN,NOUT,NAY,NODE3,NODE6,NROW,NCOL)
C
```

```
C       CALCULATES THE LEAST SQUARE INVERSE OF D.  A IS EQUIVALENT OF D
C       INVERTED MATRIX IS PLACED ON TAPE 2 FOR CHAIN7
C
        DIMENSION A(120,48),B(48,48),C(1,120),DUM(120)
C
        NOM=1
        JMAX=NROW
        IF (JMAX-120) 1,1,33
1       KMAX=NCOL
        IF (KMAX-48) 2,2,33
2       REWIND NIN
        DO 3 J=1,JMAX
        DO 3 K=1,KMAX
        A(J,K)=0.0
3       CONTINUE
        IF (NAY) 4,5,4
4       WRITE (6,34)
5       DO 11 I=1,JMAX
        IF (NODE3) 7,6,7
6       READ (NIN) (DUM(K),K=1,KMAX)
        GO TO 8
7       READ (5,35) (DUM(K),K=1,KMAX)
8       DO 9 K=1,KMAX
9       A(I,K)=DUM(K)
        IF (NAY) 10,11,10
10      WRITE (6,36) (A(I,K),K=1,KMAX)
11      CONTINUE
C       OBTAIN PRODUCT OF A AND A TRANSPOSE
        IF (NAY) 12,13,12
12      WRITE (6,37)
13      DO 16 J=1,KMAX
        DO 14 K=1,KMAX
        B(J,K)=0.0
        DO 14 I=1,JMAX
        B(J,K)=B(J,K)+A(I,J)*A(I,K)
14      CONTINUE
        IF (NAY) 15,16,15
15      WRITE (6,36) (B(J,K),K=1,KMAX)
16      CONTINUE
        DO 17 J=1,120
        C(1,J)=0.0
17      CONTINUE
        DETER=0.0
        CALL MATINV (B,KMAX,C,0,DETER)
        IF (NAY) 18,20,18
18      WRITE (6,38)
        DO 19 N=1,KMAX
        WRITE (6,36) (B(N,K),K=1,KMAX)
19      CONTINUE
C       CALC. (INERSE OF A TRANSPOSE*A)*A TRANSPOSE
        WRITE (6,39)
20      REWIND NOUT
        REWIND NIN
        DO 27 I=1,KMAX
        DO 22 J=1,JMAX
        C(1,J)=0.0
        DO 21 K=1,KMAX
        C(1,J)=C(1,J)+B(I,K)*A(J,K)
21      CONTINUE
```

```
22     CONTINUE
       DO 23 J=1,JMAX
23     DUM(J)=C(1,J)
       IF (NAY) 24,25,24
24     WRITE (6,36) (C(1,J),J=1,JMAX)
25     WRITE (NOUT) (DUM(J),J=1,JMAX)
       WRITE (NIN) (C(1,J),J=1,JMAX)
       IF (NODE6) 26,27,26
26     WRITE (7,35) (DUM(J),J=1,JMAX)
27     CONTINUE
C      LEAST SQUARES INVERSE COMPLETED
C      EVALUATE DETERMINANT OF (A INVERSE)*(A)
       REWIND NIN
       DO 29 J=1,KMAX
       READ (NIN) (C(1,JN),JN=1,JMAX)
       DO 28 K=1,KMAX
       B(J,K)=0.0
       DO 28 I=1,JMAX
       B(J,K)=B(J,K)+C(1,I)*A(I,K)
28     CONTINUE
29     CONTINUE
       IF (NAY) 30,32,30
30     WRITE (6,40)
       DO 31 I=1,KMAX
       WRITE (6,36) (B(I,J),J=1,KMAX)
31     CONTINUE
32     CALL MATINV (B,KMAX,C,0,DETER)
       WRITE (6,41) DETER
       RETURN
33     WRITE (6,42)
       STOP
C
34     FORMAT (25H0MATRIX TO BE INVERTED, A)
35     FORMAT (1P5E14.7)
  36   FORMAT(1H06E20.8/(1H 6E20.8))
37     FORMAT (1H113HA TRANSPOSE*A)
38     FORMAT (1H125H INVERSE OF A TRANSPOSE*A)
39     FORMAT (1H120HINVERTED MATRIX AINM)
40     FORMAT (1H120X,40HUNIT MATRIX = (INVERTED MATRIX)*(MATRIX))
  41   FORMAT (1H0,29HDETERMINANT OF UNIT MATRIX = ,E15.8)
42     FORMAT (1H116HMATRIX TOO LARGE)
       END


       SUBROUTINE MATROW (MSPAN,NCHORD,NONSNG,H,I,NAY,NEED,NFLAP,PHIK,
      1PHIL,LAP,IFL,IX,FROWR)
C
C      THIS ROUTINE PERFORMS THE QUADRATURE AFTER THE STATIONS
C      AND WEIGHTS HAVE BEEN ESTABLISHED.
C
       DIMENSION GAUSS(50),FROWR(36,50),THETB(20,4),THETAA(30,16),FORR(30
      1,16),NOMB(20,3),NQ(3),THETA(4),ETA(20),YDWASH(150),FLPOS(10),NSEC(
      220),ANSWR(50),SGWT(10),FNNNN(20),FN(20)
C
       COMMON GAUSS,THETB,THETAA,FORR,NOMB,NQ,THETA,ETA,YDWASH,FLPOS,NSEC
C
       IF (LAP) 2,1,2
1      NEL2=NCHORD-NFLAP
       NEWASH=1
```

```
      GO TO 3
2     NEL2=1
      NEWASH=MSPAN*(NCHORD-NFLAP+IFL-1)+1
3     MNUMB=GAUSS(1)
      IF (NONSNG) 5,4,5
4     DELA=1.0/(100.0*H)
      SGWT(1)=13.0*DELA
      SGWT(2)=72.0*DELA
      SGWT(3)=495.0*DELA
      SGWT(4)=-1360.0*DELA
      SGWT(5)=SGWT(3)
      SGWT(6)=SGWT(2)
      SGWT(7)=SGWT(1)
      MNUMB=7
5     PKL=(PHIK-PHIL)/2.0
C     DO CHORDWISE INTEGRATION AT SPAWISE STATIONS
      DO 30 NEL=1,NEL2
      NSTAT=1
      IF (NAY) 6,7,6
6     WRITE (6,31) NEL
7     CONTINUE
      DO 19 L=1,MNUMB
      NQI=NSEC(L)
      FNNNN(L)=0.0
      IF (NQI) 8,11,8
8     DO 10 ICQ=1,NQI
      FN(ICQ)=0.0
      MM=NOMB(L,ICQ)
      CALL PRESSR (MM,NEL,NSTAT,ANSWR,FLPOS,NEED,LAP,IFL,THETAA,L)
      DO 9 LNM=1,MM
      FN(ICQ)=FORR(NSTAT,L)*ANSWR(LNM)+FN(ICQ)
      NSTAT=NSTAT+1
9     CONTINUE
      FN(ICQ)=(THETB(L,ICQ+1)-THETB(L,ICQ))*FN(ICQ)/2.0
      FNNNN(L)=FNNNN(L)+FN(ICQ)
10    CONTINUE
      NSTAT=1
11    SPHI=1.0-ETA(L)*ETA(L)
      IF (NAY) 12,13,12
12    WRITE (6,32) ETA(L),FNNNN(L)
13    CONTINUE
      IF (NONSNG) 15,14,15
14    FNNNN(L)=FNNNN(L)*SGWT(L)*SQRT(SPHI)
      GO TO 16
15    YOO=(YDWASH(I)-ETA(L))
      YOO=YOO*YOO
      NGAUS=L+MNUMB+1
      FNNNN(L)=FNNNN(L)*GAUSS(NGAUS)*SPHI/YOO
16    IF (NAY) 17,18,17
17    WRITE (6,33) FNNNN(L)
18    CONTINUE
19    CONTINUE
      DO 29 MEL=1,MSPAN
      MELL=2*(MEL-1)
      AUX=0.0
      DO 24 K=1,MNUMB
      IF (MELL) 22,20,22
20    IF (ETA(K)) 22,21,22
21    POWER=1.0
```

```
      GO TO 23
22    POWER=ETA(K)**MELL
23    AUX=AUX+FNNNN(K)*POWER
24    CONTINUE
      IF (NONSNG) 25,26,25
25    AUX=AUX*PKL
26    FROWR(NEWASH,IX)=FROWR(NEWASH,IX)+AUX
      IF (NAY) 27,28,27
27    WRITE (6,34) MELL,AUX
28    CONTINUE
      NEWASH=NEWASH+1
29    CONTINUE
30    CONTINUE
      RETURN
C
31    FORMAT (42H1CHORDWISE INTEGRALS FOR PRESSURE MODE, N=I3)
32    FORMAT (7H0ETA = E15.8/1H ,21X,7HIC 1 = E15.8)
33    FORMAT (1H ,21X,7HIC 2 = E15.8)
34    FORMAT (40H0SPANWISE INTEGRAL FOR PRESSURE MODE, M=I3,3H = E15.8)
      END


      SUBROUTINE MATINV (A,N,B,M,DETERM)
C
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C
      DIMENSION IPIVOT(48),INDEX(48,2)
      DIMENSION A(48,48),B(48,1),PIVOT(48)
C
C     INITIALIZATION
C
      DETERM=1.0
      DO 2 J=1,N
2     IPIVOT(J)=0
      DO 21 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
      T=0.0
      DO 7 J=1,N
      IF (IPIVOT(J)-1) 3,7,3
3     DO 6 K=1,N
      IF (IPIVOT(K)-1) 4,6,25
   4  IF (ABS(T)-ABS(A(J,K))) 5,6,6
5     IROW=J
      ICOLUM=K
      T=A(J,K)
6     CONTINUE
7     CONTINUE
      IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
      IF (IROW-ICOLUM) 8,12,8
8     DETERM=-DETERM
      DO 9 L=1,N
      T=A(IROW,L)
      A(IROW,L)=A(ICOLUM,L)
9     A(ICOLUM,L)=T
```

```
      IF (M) 12,12,10
10    DO 11 L=1,M
      T=B(IROW,L)
      B(IROW,L)=B(ICOLUM,L)
11    B(ICOLUM,L)=T
12    INDEX(I,1)=IROW
      INDEX(I,2)=ICOLUM
      PIVOT(I)=A(ICOLUM,ICOLUM)
      DETERM=DETERM*PIVOT(I)
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
      A(ICOLUM,ICOLUM)=1.0
      DO 13 L=1,N
13    A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
      IF (M) 16,16,14
14    DO 15 L=1,M
15    B(ICOLUM,L)=B(ICOLUM,L)/PIVOT(I)
C
C     REDUCE NON-PIVOT ROWS
C
16    DO 21 L1=1,N
      IF (L1-ICOLUM) 17,21,17
17    T=A(L1,ICOLUM)
      A(L1,ICOLUM)=0.0
      DO 18 L=1,N
18    A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
      IF (M) 21,21,19
19    DO 20 L=1,M
20    B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
21    CONTINUE
C
C     INTERCHANGE COLUMNS
C
      DO 24 I=1,N
      L=N+1-I
      IF (INDEX(L,1)-INDEX(L,2)) 22,24,22
22    IROW=INDEX(L,1)
      ICOLUM=INDEX(L,2)
      DO 23 K=1,N
      T=A(K,IROW)
      A(K,IROW)=A(K,ICOLUM)
      A(K,ICOLUM)=T
23    CONTINUE
24    CONTINUE
25    RETURN
      END


      SUBROUTINE PRESSR (MM,NEL,NSTAT,ANSWR,FLPOS,NEED,LAP,IFL,THETT,LL)
C
      DIMENSION THETT(30,1),ANSWR(1),FLPOS(1)
C
      LAC=NSTAT
      IF (LAP) 9,1,9
1     IF (NEED) 2,3,2
2     KEL=NEL-1
      GO TO 4
3     KEL=NEL
```

```
4       IF (KEL) 5,5,7
5       DO 6 LNM=1,MM
        AUY=THETT(LAC,LL)/2.0
        ANSWR(LNM)=COS(AUY)/SIN(AUY)
        LAC=LAC+1
6       CONTINUE
        RETURN
7       FNEL=KEL
        DO 8 LNM=1,MM
        AUY=THETT(LAC,LL)
        ANSWR(LNM)=4.0*SIN(AUY*FNEL)/(2.0**(2*KEL))
        LAC=LAC+1
8       CONTINUE
        RETURN
9       AUFL=FLPOS(IFL)
        DO 10 LNM=1,MM
        AUY=THETT(LAC,LL)
        UNUM=SIN(0.5*(AUFL+AUY))
        DENOM=SIN(0.5*(AUFL-AUY))
        ANSWR(LNM)=(ALOG(ABS(UNUM/DENOM)))/3.14159265
        LAC=LAC+1
10      CONTINUE
        RETURN
        END


        SUBROUTINE MPRINT (TEXTM,NW,MTAPE,MAT2,MAT3)
C
C       THIS ROUTINE IS USED TO PRINT A MATRIX
C
        DIMENSION Q000FL(150),A(5),TEXTM(9)
C
        NROWS=MAT2
        NCOLS=MAT3
        REWIND MTAPE
C       NOW BEGIN PRINT LOOP
        LINES=0
        DO 6 J=1,NROWS
        READ (MTAPE) (Q000FL(I),I=1,NCOLS)
        K=1
1       A(1)=0.0
        A(2)=0.0
        A(3)=0.0
        A(4)=0.0
        A(5)=0.0
        A(1)=Q000FL(K)
        A(2)=Q000FL(K+1)
        A(3)=Q000FL(K+2)
        A(4)=Q000FL(K+3)
        A(5)=Q000FL(K+4)
        N1=K
        N2=K+1
        N3=K+2
        N4=K+3
        N5=K+4
        K=K+5
        IF (LINES) 2,3,2
2       IF (44-LINES) 3,4,4
C       START NEW PAGE
```

```
3     WRITE (6,9) (TEXTM(I),I=1,NW)
      WRITE (6,7) NROWS,NCOLS
      WRITE (6,8)
      LINES=5
4     WRITE (6,11) J,N1,A(1),N2,A(2),N3,A(3),N4,A(4),N5,A(5)
      LINES=LINES+1
      IF (NCOLS-K) 5,1,1
5     WRITE (6,10)
      LINES=LINES+1
6     CONTINUE
      RETURN
C
7     FORMAT (1H030X,I4,9H ROWS BY I4,8H COLUMNS)
8     FORMAT (1H02X8HROW  COL,18X,3HCOL,19X,3HCOL,19X,3HCOL,19X,3HCOL)
9     FORMAT (1H129X,9A6)
10    FORMAT (1H )
11    FORMAT (1H 2X,I3,I5,1X,E15.8,2X,I3,2X,E15.8,2X,I3,2X,E15.8,2X,I3,
     12X,E15.8,2X,I3,2X,E15.8)
      END

      SUBROUTINE FNUD (FEN,GAUSS,WTGS)
C
      DIMENSION NLOC(14),TABLE(70),TWGTS(70),GAUSS(1),WTGS(1)
C
      DATA NLOC/2,4,7,10,14,18,23,28,34,40,47,54,62,70/
      DATA TWGTS/.888888888,.555555555,.652145154,.347854845,.568888888,
     1.478628670,.236926885,.467913934,.360761573,.171324492,.417959183,
     2.381830050,.279705391,.129484966,.362683783,.313706645,.222381034,
     3.101228536,.330239355,.312347077,.260610696,.180648160,.812743884E
     4-1,.295524224,.269266719,.219086362,.149451349,.666713443E-1,
     5.272925086,.262804544,.233193764,.186290210,.125580369,.556685671E
     6-1,.249147045,.233492536,.203167426,.160078328,.106939326,
     7.471753364E-1,.232551553,.226283180,.207816047,.178145980,
     8.138873510,.921214998E-1,.404840048E-1,.215263853,.205198463,
     9.185538397,.157203167,.121518570,.801580872E-1,.351194603E-1,
     A.202578241,.198431485,.186161000,.166269205,.139570677,.107159220,
     B.703660475E-1,.307532420E-1,.189450610,.182603415,.169156519,
     C.149595988,.124628971,.951585117E-1,.622535239E-1,.271524594E-1/
      DATA TABLE/0.0,.774596669,.339981043,.861136311,0.0,.538469310,
     1.906179845,.238619186,.661209386,.932469514,0.0,.405845151,
     2.741531185,.949107912,.183434642,.525532409,.796666477,.960289856,
     30.0,.324253423,.613371432,.836031107,.968160239,.148874339,
     4.433395394,.679409568,.865063366,.973906528,0.0,.269543156,
     5.519096129,.730152005,.887062599,.978228658,.125333408,.367831498,
     6.587317954,.769902674,.904117256,.981560634,0.0,.230458316,
     7.448492751,.642349339,.801578090,.917598399,.984183054,.108054948,
     8.319112368,.515248636,.687292904,.827201315,.928434883,.986283808,
     90.0,.201194094,.394151347,.570972172,.724417731,.848206583,
     A.937273392,.987992518,.950125098E-1,.281603550,.458016777,
     B.617876244,.755404408,.865631202,.944575023,.989400935/
C
C
      N=FEN+1.0
      INDEX=NLOC(N-3)
      N2=N/2
      J=N-1
      DO 1 I=1,N2
      GAUSS(I)=-TABLE(INDEX)
```

```
      GAUSS(J)=TABLE(INDEX)
      WTGS(I)=TWGTS(INDEX)
      WTGS(J)=TWGTS(INDEX)
      J=J-1
1     INDEX=INDEX-1
      RETURN
      END


      FUNCTION FSQM (MM,IR)
C
      GMM=MM
      I=(IR+2)/2
      IF (I-1) 1,1,2
1     FSQM=GMM+1.0
      GO TO 8
2     IF (I-2) 3,3,4
3     FSQM=0.5*(GMM+1.0)-GMM
      GO TO 8
4     II=3
      EM1=0.5*(GMM+1.0)
      EM2=GMM
      ENUM1=3.0
      DEM1=4.0
      ENUN1=1.0
      DEN1=2.0
      FS1=ENUM1/DEM1
      FS2=ENUN1/DEN1
5     IF (I-II) 7,7,6
6     ENUM1=ENUM1+2.0
      DEM1=DEM1+2.0
      ENUN1=ENUN1+2.0
      DEN1=DEN1+2.0
      FS1=FS1*ENUM1/DEM1
      FS2=FS2*ENUN1/DEN1
      II=II+1
      GO TO 5
7     FSQM=EM1*FS1-EM2*FS2
8     CONTINUE
      RETURN
      END


      FUNCTION FKERNL (XO,YO,S,FMACH)
C
      BETASQ=1.0-FMACH*FMACH
      COMP=XO*XO+BETASQ*S*S*YO*YO
      SQCOMP=SQRT(COMP)
      FKERNL=1.0+XO/SQCOMP
      IF (SQCOMP) 1,1,2
    1 WRITE (6,601)
      STOP
    2 CONTINUE
      RETURN
  601 FORMAT (1H0,///10X,32H***SQCOMP=0, EXIT FROM FKERNL***)
      END


      FUNCTION FPMI (ETA0,ETA1,MM)
```

```
C
      PHI=ACOS(ETA0)
      PHI1=ACOS(ETA1)
      FPMI=((SIN(PHI))**3.0-(SIN(PHI1))**3.0)/3.0
      IF (MM-2) 3,1,1
1     IM=2
2     GM=IM
      FPMI=((ETA0**GM)*(SIN(PHI))**3.0-(ETA1**GM)*(SIN(PHI1))**3.0)/(GM+
     13.0)+(GM*FPMI)/(GM+3.0)
      IM=IM+2
      IF (IM-MM) 2,2,3
3     RETURN
      END

      FUNCTION FRMI (ETA0,ETA1,MM)
C
      PHI=ACOS(ETA0)
      PHI1=ACOS(ETA1)
      IF (MM-2) 1,2,2
1     FRMI=0.5*(PHI-PHI1)-0.25*(SIN(2.0*PHI)-SIN(2.0*PHI1))
      GO TO 6
2     FRMI=0.125*((PHI-PHI1)-0.25*(SIN(4.0*PHI)-SIN(4.0*PHI1)))
      IF (MM-2) 3,3,4
3     GO TO 6
4     IM=4
5     GM=IM
      FRMI=(ETA0**(GM-1.0)*(SIN(PHI))**3.0-ETA1**(GM-1.0)*(SIN(PHI1))**
     13.0+(GM-1.0)*FRMI)/(GM+2.0)
      IM=IM+2
      IF (IM-MM) 5,5,6
6     RETURN
      END
```

```
      PROGRAM NLBODY(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
      DIMENSION ALPHA1(18),PHI1(9),Q1(9),R1(9),COMNT(18),C(10)
      DIMENSION CYS(4),CNS(4),CMS(4),CES(4),RLS(10),CYSV(3),CNSV(3)
C
      COMMON DX1,DX,ISTART,NEXIT,LREF,SREF,CG,CYS,CNS,CMS,CES,RLS,CYSV,
     1CNSV,PHI
C
      REAL LREF
C
      CALL DATA
      READ (5,5) COMNT
      WRITE (6,25) COMNT
 5    FORMAT(18A4)
      READ (5,10) LREF,SREF,CG,DX1
 10   FORMAT(4F10.4)
      READ (5,15) NALPHA,NPHI,NQ,NR
 15   FORMAT(5I2)
      READ (5,20) (ALPHA1(I),I=1,NALPHA)
      READ (5,20) (PHI1(I),I=1,NPHI)
      READ (5,20) (Q1(I),I=1,NQ)
      READ (5,20) (R1(I),I=1,NR)
 20   FORMAT(9F8.4)
 25   FORMAT(1H1,18A4)
      DO 50 I=1,NPHI
      PHI=.0174533*PHI1(I)
      CP=COS(PHI)
      SP=SIN(PHI)
      CALL COEFF
      DO 50 J=1,NR
      DO 50 K=1,NQ
      C1=R1(J)*CP+Q1(K)*SP
      C2=Q1(K)*CP-R1(J)*SP
      WRITE (6,30) PHI1(I),Q1(K),R1(J)
 30   FORMAT(5H0PHI=,F8.3,5H    Q=,F7.4,5H    R=,F7.4/
     18H0  ALPHA,30X2HCN,15X2HCM,15X3HCY ,14X3HCEM,14X3HCRM)
      DO 50 L=1,NALPHA
      ALPHA=.0174533*ALPHA1(L)
      CA=COS(ALPHA)
      SA=SIN(ALPHA)
      C(1)=C1*CA
      C(2)=SA*CA
      C(3)=C2*CA
      C(4)=CA**2
      CYSPOT=-(C(1)*CYS(1)+C(2)*CYS(2)+C(3)*CYS(3)+C(4)*CYS(4))/SREF
      CESPOT=-(C(1)*CES(1)+C(2)*CES(2)+C(3)*CES(3)+C(4)*CES(4))/
     1        (SREF*LREF)
      CNSPOT=-(C(1)*CNS(1)+C(2)*CNS(2)+C(3)*CNS(3)+C(4)*CNS(4))/SREF
      CMSPOT=-(C(1)*CMS(1)+C(2)*CMS(2)+C(3)*CMS(3)+C(4)*CMS(4))/
     1        (SREF*LREF)
      CYSP1=CYSPOT
      CYSPOT=CYSP1*CP-CNSPOT*SP
      CNSPOT=CYSP1*SP+CNSPOT*CP
      CYSP1=CESPOT
      CESPOT=CYSP1*CP-CMSPOT*SP
      CMSPOT=CYSP1*SP+CMSPOT*CP
      C(10)=C(4)
```

```
      C(9)=2.*C(1)*CA
      C(8)=2.*C(1)*C1
      C(7)=2.*C(3)*CA
      C(6)=2.*C(3)*C2
      C(5)=C(2)*CA
      C(4)=C(2)*SA
      C(3)=C(3)*C1
      C(2)=C(2)*C2
      C(1)=C(1)*SA
      CLSPOT=0.
      CLSVIS=0.
      DO 35 M=1,10
   35 CLSPOT=CLSPOT+C(M)*RLS(M)
      CLSPOT=CLSPOT/(SREF*LREF)
      WRITE (6,40) ALPHA1(L),CNSPOT,CMSPOT,CYSPOT,CESPOT,CLSPOT
   40 FORMAT(1H ,F7.4,10X9HPOTENTIAL,5X5(3X1PE12.4,2X))
      CALL VISC(SA,Q1(K),R1(J),CNSVIS,CMSVIS,CYSVIS,CESVIS,CLSVIS)
      WRITE (6,45) CNSVIS,CMSVIS,CYSVIS,CESVIS,CLSVIS
   45 FORMAT(1H ,17X9HVISCOUS  ,5X5(3X1PE12.4,2X)/1H )
   50 CONTINUE
      STOP
      END


      SUBROUTINE FORCE
C
      DIMENSION CY(4),CN(4),RL(9),CY0(4),CN0(4),RL0(9),KPLRE(11),
     1 KPLIM(11)
      DIMENSION CYS(4),CNS(4),CMS(4),CES(4),RLS(10),CYSV(3),CNSV(3)
      DIMENSION A1(12),B1(12),APR1(12),BPR1(12),C(2)
C
      COMMON DX1,DX,ISTART,NEXIT,LREF,SREF,CG,CYS,CNS,CMS,CES,RLS,CYSV,
     1CNSV,PHI
      COMMON X,RB,RBPR,RB2,S,DSDX,CDCY,CDCL,N,A1,B1,APR1,BPR1,C
C
      REAL LREF,KPLRE,KPLIM
C
      C1=(X-CG)/LREF
      CY(1)=25.13274*(A1(3)-RB)*RB*C1
      CY(2)=12.56637*B1(3)*RB
      CY(3)=2.*C1*CY(2)
      CN(1)=CY(3)
      CN(2)=-12.56637*(A1(3)+RB)*RB
      CN(3)=2.*C1*CN(2)
      CY(4)=12.56637*C(1)-2.*S*APR1(2)
      CN(4)=12.56637*C(2)-2.*S*BPR1(2)
      RL(1)=CY(1)-CN(3)
      RL(2)=2.*CY(3)
      RL(3)=C1*(CY(1)-CN(3))
      RL(4)=CY(2)
      RL(5)=CY(4)
      RL(6)=C1*CY(3)
      RL(7)=C1*CY(4)
      RL(8)=-C1*CN(1)
      RL(9)=-C1*CN(4)
      CY(1)=CY(1)+4.*C1*S
      CN(3)=CN(3)+4.*C1*S
      CN(2)=CN(2)+2.*S
      IF (ISTART) 200,5,10
```

```
5     DO 6 I=1,4
      CYS(I)=0.
      CNS(I)=0.
      CMS(I)=0.
      CES(I)=0.
      CYO(I)=CY(I)
6     CNO(I)=CN(I)
      DO 7 I=1,9
      RLS(I)=0.
7     RLO(I)=RL(I)
      ISTART=1
      GO TO 200
10    XA=X-.5*DX-CG
      DO 15 I=1,4
      CYS(I)=CYS(I)+CY(I)-CYO(I)
      CNS(I)=CNS(I)+CN(I)-CNO(I)
      CMS(I)=CMS(I)-XA*(CN(I)-CNO(I))
15    CES(I)=CES(I)-XA*(CY(I)-CYO(I))
      DO 20 I=1,9
20    RLS(I)=RLS(I)+(RL(I)+RLO(I))*DX/2.
      IF (NEXIT) 200,25,35
25    DO 27 I=1,4
      CYO(I)=CY(I)
27    CNO(I)=CN(I)
      DO 30 I=1,9
30    RLO(I)=RL(I)
      GO TO 200
35    RLS(5)=RLS(5)+12.56637*(A1(2)*(A1(3)+RB)+B1(2)*B1(3))*RB
      RLS(7)=RLS(7)+12.56637*(A1(2)*(A1(3)+RB)+B1(2)*B1(3))*RB*C1
      RLS(9)=RLS(9)+12.56637*(B1(2)*(A1(3)-RB)+A1(2)*B1(3))*RB*C1
      RLS(10)=-12.56637*(A1(2)*((A1(3)+RB)*BPR1(2)-APR1(2)*B1(3))
     1                  +B1(2)*((A1(3)-RB)*APR1(2)+BPR1(2)*B1(3)))*RB
      N1=N-1
      IF (N1) 200,200,37
37    DO 40 I=1,N1
      KPLRE(I)=0.
40    KPLIM(I)=0.0
      DO 50 M=1,N1
      N3=N1-M+1
      IF (M-2) 42,50,42
42    RBI=1.
      DO 45 I=1,N3
      MI=M+I
      RBI=RBI*RB
      IF (MI-2) 200,45,43
43    D=A1(M)*A1(MI)+B1(M)*B1(MI)
      E=A1(M)*B1(MI)-B1(M)*A1(MI)
      KPLRE(I)=KPLRE(I)+D*RBI
      KPLIM(I)=KPLIM(I)+E*RBI
45    CONTINUE
50    CONTINUE
      M=N1+1
      D=B1(3)*KPLRE(1)+(A1(3)-RB)*KPLIM(1)
      E=B1(3)*KPLIM(1)+(A1(3)-RB)*KPLRE(1)
      IF (N1-3) 65,65,55
55    RBI=RB
      DO 60 I=4,N1
      RBI=RBI*RB
      AI=I-2
```

```
      D=D+AI*(A1(I)*KPLIM(I)-B1(I)*KPLRE(I))/RBI
60    E=E+AI*(A1(I)*KPLRE(I)+B1(I)*KPLIM(I))/RBI
65    RLS(5)=RLS(5)+6.283185*E
      RLS(7)=RLS(7)+6.283185*E*C1
      RLS(9)=RLS(9)+6.283185*D*C1
      RLS(10)=RLS(10)-6.283185*(D*APR1(2)+E*BPR1(2))
200   RETURN
      END


      SUBROUTINE DATA
C
      DIMENSION COMAIN(40),COMFOR(59)
      DIMENSION X1(40),RB1(40),DRDX1(40),S1(40),DSDX1(40),CDCY1(40),CDCL
     11(40),M(40),REAL1(11,40),IMAG1(11,40),REPR1(11,40),IMPR1(11,40)
C
      COMMON COMAIN,COMFOR
      COMMON NX,X1,RB1,DRDX1,S1,DSDX1,CDCY1,CDCL1,M,REAL1,IMAG1,REPR1,
     1IMPR1
C
      REAL IMAG1,IMPR1
C
      READ (5,5) MAXZET,NX
5     FORMAT(24I3)
      DO 7 I=1,NX
      DO 7 J=1,11
      REAL1(J,I)=0.
      IMAG1(J,I)=0.
      REPR1(J,I)=0.
7     IMPR1(J,I)=0.
      READ (5,30) (X1(I),I=1,NX)
      READ (5,30) (RB1(I),I=1,NX)
      READ (5,30) (DRDX1(I),I=1,NX)
      READ (5,30) (S1(I),I=1,NX)
      READ (5,30) (DSDX1(I),I=1,NX)
      READ (5,30) (CDCY1(I),I=1,NX)
      READ (5,30) (CDCL1(I),I=1,NX)
30    FORMAT(6E12.5)
      IF (MAXZET-1) 45,10,45
10    DO 15 I=1,NX
15    M(I)=1
      GO TO 300
45    DO 110 I=1,NX
      READ (5,5) NZETA,ISYM
      IF (NZETA) 55,55,60
55    N1=MAXZET
      M(I)=N1
      GO TO 65
60    N1=NZETA
      M(I)=N1
65    IF (N1-1) 300,110,70
70    N1=N1-1
      IF (ISYM) 300,75,95
75    READ (5,30) (REAL1(J,I),J=1,N1)
      READ (5,30) (REPR1(J,I),J=1,N1)
      DO 90 J=1,N1,2
      IMAG1(J,I)=REAL1(J,I)
      IMPR1(J,I)=REPR1(J,I)
      REAL1(J,I)=0.
```

```
 90   REPRI(J,I)=0.0
      GO TO 110
 95   READ (5,30) (REAL1(J,I),IMAG1(J,I),J=1,N1)
      READ (5,30) (REPR1(J,I),IMPR1(J,I),J=1,N1)
 110  CONTINUE
 300  RETURN
      END

      SUBROUTINE VISC(SA,Q1,R1,CNSVIS,CMSVIS,CYSVIS,CESVIS,CLSVIS)
C
      DIMENSION DUM1(3),DUM2(32),DUM3(59),X1(40),DUM4(160),CDCY1(40),
     1CDCL1(40)
C
      COMMON DX1,DUM1,LREF,SREF,CG,DUM2,PHI,DUM3,NX,X1,DUM4,CDCY1,CDCL1
C
      REAL LREF
C
      SP=SIN(PHI)
      CP=COS(PHI)
      CLSVIS=0.
      CNSVIS=0.
      CMSVIS=0.
      CYSVIS=0.
      CESVIS=0.
      ARM=(X1(1)-CG)/LREF
      V=-SA*SP+2.*R1*ARM
      W=SA*CP+2.*Q1*ARM
      CYVO=CDCY1(1)*V*ABS(V)
      CNVO=CDCL1(1)*W*ABS(W)
      CEVO=-ARM*CYVO
      CMVO=-ARM*CNVO
      X=X1(1)
      XO=X
 10   X=AMIN1(X+DX1,X1(NX))
      CDCY=AINTRP(X1,CDCY1,NX,X,4)
      CDCL=AINTRP(X1,CDCL1,NX,X,4)
      ARM=(X-CG)/LREF
      V=-SA*SP+2.*R1*ARM
      W=SA*CP+2.*Q1*ARM
      CYV=CDCY*V*ABS(V)
      CNV=CDCL*W*ABS(W)
      CEV=-ARM*CYV
      CMV=-ARM*CNV
      DX2=(X-XO)/2.
      CNSVIS=CNSVIS+(CNV+CNVO)*DX2
      CYSVIS=CYSVIS+(CYV+CYVO)*DX2
      CMSVIS=CMSVIS+(CMV+CMVO)*DX2
      CESVIS=CESVIS+(CEV+CEVO)*DX2
      XO=X
      CNVO=CNV
      CYVO=CYV
      CMVO=CMV
      CEVO=CEV
      IF (X-X1(NX)) 10,20,20
 20   CYSVIS=CYSVIS/SREF
      CNSVIS=CNSVIS/SREF
      CESVIS=CESVIS/SREF
      CMSVIS=CMSVIS/SREF
```

```
      RETURN
      END

      SUBROUTINE LOCVAL
C
      DIMENSION FCN(40),COMAIN(39)
      DIMENSION A1(12),B1(12),APR1(12),BPR1(12),C(2)
      DIMENSION X1(40),RB1(40),DRDX1(40),S1(40),DSDX1(40),CDCY1(40),CDCL
     11(40),M(40),REAL1(11,40),IMAG1(11,40),REPR1(11,40),IMPR1(11,40)
C
      COMMON COMAIN,PHI
      COMMON X,RB,RBPR,RB2,S,DSDX,CDCY,CDCL,N1,A1,B1,APR1,BPR1,C
      COMMON NX,X1,RB1,DRDX1,S1,DSDX1,CDCY1,CDCL1,M,REAL1,IMAG1,REPR1,
     1IMPR1
C
      REAL IMAG1,IMPR1,IMAG,IMPR
C
      RB=AINTRP(X1,RB1,NX,X,4)
      RB2=RB**2
      RBPR=AINTRP(X1,DRDX1,NX,X,4)
      S=AINTRP(X1,S1,NX,X,4)
      DSDX=AINTRP(X1,DSDX1,NX,X,4)
      DO 10 IL=1,NX
      IF (X-X1(IL)) 20,15,10
   10 CONTINUE
   15 N1=M(IL)
      GO TO 25
   20 N1=M(IL-1)
   25 A1(1)=RB
      B1(1)=0.
      APR1(1)=RBPR
      BPR1(1)=0.
      C(1)=0.
      C(2)=0.
      A1(2)=0.
      B1(2)=0.
      APR1(2)=0.
      BPR1(2)=0.
      A1(3)=0.
      B1(3)=0.
      IF (N1-1) 100,100,30
   30 DO 55 J=2,N1
      J1=J-1
      AJ=J1
      PHIJ=AJ*PHI
      DO 35 K=1,NX
   35 FCN(K)=REAL1(J1,K)
      REAL=AINTRP(X1,FCN,NX,X,4)
      DO 40 K=1,NX
   40 FCN(K)=IMAG1(J1,K)
      IMAG=AINTRP(X1,FCN,NX,X,4)
      DO 45 K=1,NX
   45 FCN(K)=REPR1(J1,K)
      REPR=AINTRP(X1,FCN,NX,X,4)
      DO 50 K=1,NX
   50 FCN(K)=IMPR1(J1,K)
      IMPR=AINTRP(X1,FCN,NX,X,4)
      SN=SIN(PHIJ)
```

```
      CS=COS(PHI J)
      A1(J)=REAL*CS+IMAG*SN
      B1(J)=IMAG*CS-REAL*SN
      APR1(J)=REPR*CS+IMPR*SN
   55 BPR1(J)=IMPR*CS-REPR*SN
      C(1)=RB2*APR1(2)
      C(2)=RB2*BPR1(2)
      IF (N1-2) 100,100,60
   60 N2=N1-1
      DO 65 N=2,N2
      AN=N-2
      J=N+1
      AJ=J-2
      C(1)=C(1)-(AJ*(A1(J)*APR1(N)+B1(J)*BPR1(N))+
     1             AN*(A1(N)*APR1(J)+B1(N)*BPR1(J)))*RB
   65 C(2)=C(2)+(AJ*(A1(J)*BPR1(N)-B1(J)*APR1(N))+
     1             AN*(B1(N)*APR1(J)-A1(N)*BPR1(J)))*RB
  100 RETURN
      END

      FUNCTION AINTRP (X,Y,N,X1,M)
C
      DIMENSION X(40),Y(40)
C
      I=0
    5 I=I+1
      IF (N-I) 70,10,10
   10 IF (X(I)-X1) 5,20,15
   15 IF (I-1) 100,70,25
   20 AINTRP=Y(I)
      GO TO 100
   25 M2=M/2+1
      IF (I-M2) 30,30,35
   30 I1=1
      I2=M
      GO TO 50
   35 IF (N-I-M2) 40,45,45
   40 I2=N
      I1=I2-M+1
      GO TO 50
   45 I1=I-M2
      I2=I1+M-1
   50 AINTRP=0.0
      DO 65 I=I1,I2
      FCN=Y(I)
      DO 60 J=I1,I2
      IF (J-I) 55,60,55
   55 FCN=FCN*(X1-X(J))/(X(I)-X(J))
   60 CONTINUE
   65 AINTRP=AINTRP+FCN
      GO TO 100
   70 WRITE (6,75) Y(1),Y(N),X1
   75 FORMAT (53H AINTRP OUT OF RANGE FOR FUNCTION WITH END VALUES OF ,
     1E12.5,4H AND,E12.5,5H  X1=,E12.5)
  100 RETURN
      END
```

```
      SUBROUTINE COEFF
C
      DIMENSION COMAIN(36),COMFOR(58),X1(40)
C
      COMMON DX1,DX,ISTART,NEXIT,COMAIN,X,COMFOR,NX,X1
C
      NEXIT=0
      ISTART=0
      DX=0.
      X=X1(1)
   10 X=X+DX
      CALL LOCVAL
      CALL FORCE
      DX=DX1
      IF (NEXIT) 500,12,500
   12 IF (X+DX-X1(NX)) 10,15,15
   15 NEXIT=1
      DX=X1(NX)-X
      GO TO 10
  500 RETURN
      END
```

```
      PROGRAM NLWING(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
      DIMENSION XI1(2),XIO(3),ETA(20),ETADW(80),TN(3),
     1X(40),Y(40),CI(80),CF(80),W(20),
     2C4(20,20),C6(20,20),CIRCLN(20),DWASH(20),TRVQU(20,20), C5(20,20),
     3C470(20,20),C570(20,20),TRV70(20,20)
     4,ALPHEF(20),WT(20),SINAEF(20),GAM(20)
     5,COEF(10,10),CHORD(10),XL1(20),XL2(20),XPMOM(20),CIRCL1(20)
     6,CIRCL2(20),AL(10),WGHT(10),SPAN(20),ALPH(20)
C
  1   READ (5,60) ALPHA,BETA,DALPHA
      READ (5,60) ETAO,ETAB,TR,TNLE
      READ (5,60) P,Q,R
      READ (5,60) REFL,XCG,ZCG
      READ (5,60) CD,CDXPOS
      READ (5,55) NSTA,NDWSH
      READ (5,55) NALPHA,NIT
      READ (5,55) NSYM
      READ (5,60) (ETA(I),I=1,NSTA)
      READ (5,60) (ETADW(I),I=1,NDWSH)
      DO 5 I=1,3
  5   READ (5,60) XIO(I),TN(I)
      READ (5,60) (ALPHEF(I),I=1,NDWSH)
      READ (5,60) (AL(I),I=1,10)
      READ (5,60) (WGHT(I),I=1,10)
      ALPHA=ALPHA*.0174533
      BETA=BETA*.0174533
      DALPHA=DALPHA*.0174533
      DO 7 I=1,10
  7   AL(I)=AL(I)*.0174533
      P=P*2./REFL
      Q=Q*2./REFL
      R=R*2./REFL
      CBETA=COS(BETA)
C
C     CALCULATE COORDINATES OF DOWNWASH CONTROL POINTS
C
      NROW=0
      DO 26 J=1,NDWSH
      ALPHEF(J)=ALPHEF(J)*.0174533
      XI=XIO(3)
      YI=ETAO
      YF=ETAB
      IF (ETADW(J)-YI) 25,10,10
 10   IF (ETADW(J)-YF) 15,15,25
 15   NROW=NROW+1
      Y(NROW)=ETADW(J)
      X(NROW)=XI+(Y(NROW)-YI)*TN(3)
      GO TO 26
 25   WRITE (6,65) ETAO,ETADW(J),ETAB
      STOP
 26   CONTINUE
      N=NSTA
      NCOL=N-1
C
C     NOW CALCULATE LAGRANGIAN COEFFICIENTS
C
```

```
      CALL LGRANG(ETA,COEF,N)
C
C     CALCULATE LOCAL CHORDS
C
      DO 17 I=1,NCOL
      IN=NCOL+I
      ETA(IN)=ETA(I)
      SPAN(I)=ETA(I)/(ETAB-ETAO)
      SPAN(IN)=-SPAN(I)
      CHORD(I)=1.+(TR-1.)*ETA(I)/(ETAB-ETAO)
   17 CHORD(IN)=CHORD(I)
      NROW2=NROW+1
      NROW1=2*NROW
      J1=0
      DO 110 J=NROW2,NROW1
      J1=J1+1
      ALPHEF(J)=ALPHEF(J1)
      X(J)=X(J1)
  110 Y(J)=-Y(J1)
      XI1(1)=XIO(1)+ETAB*TN(1)
      XI1(2)=XIO(2)+ETAB*TN(2)
      DO 172 M=1,NALPHA
      ALPHD=ALPHA*57.2958
      BETD=BETA*57.2958
      WRITE (6,300) ALPHD,BETD
      SALPHA=SIN(ALPHA)
      CALPHA=COS(ALPHA)
      DO 170 L=1,NIT
      NCOL=NSTA-1
      NROW=NDWSH
C
C     DETERMINE DOWNWASH CONTRIBUTION FROM LEADING LIFTING LINE
C
      DO 40 J=1,NROW1
      CALL LLINE(X(J),Y(J),0.0,XIO(1),XI1(1),ETAO,ETAB,TN(1),
     1ALPHEF(J),BETA,COEF,CI,N)
      CALL TRVORT(X(J),Y(J),0.0,XIO(1),XI1(1),ETAO,ETAB,TN(1),
     1ALPHEF(J),BETA,COEF,CF,N)
      DO 29 I=1,N
   29 TRVQU(I,J)=CF(I)
      DO 30 I=1,N
   30 C4(I,J)=CI(I)+CF(I)
   40 CONTINUE
C
C     TEST FOR SYMMETRICAL LOADING(NSYM=0)
      IF(NSYM-1)45,56,45
   45 DO 50 J=1,NROW
      J2=J+NROW
      DO 70 I=1,NCOL
      TRVQU(I,J)=TRVQU(I,J)+TRVQU(I,J2)
   70 C5(I,J)=C4(I,J)+C4(I,J2)
   50 CONTINUE
      GO TO 59
   56 DO 73 I=NROW2,NROW1
      IN=I-NCOL
      DO 73 J=1,NROW
      JN=J+NROW
      TRVQU(I,J)=TRVQU(IN,JN)
      TRVQU(I,JN)=TRVQU(IN,J)
```

```
      C5(I,J)=C4(IN,JN)
  73  C5(I,JN)=C4(IN,J)
      DO 72 I=1,NCOL
      DO 72 J=1,NROW1
      C5(I,J)=C4(I,J)
C
C     DETERMINE [illegible] WASH CONTRIBUTION [illegible] LINE
C
  59  DO 41 J=1,NROW1
      CALL LLINE(X(J),Y(J),0.0,XIO([illegible]),X[illegible] [illegible]
     1ALPHEF(J),BETA,COEF,CI,N)
      CALL TRVORT(X(J),Y(J),0.0,XIO(2),XI1(2),ETAO,ETAB,IN([illegible]),
     1ALPHEF(J),BETA,COEF,CF,N)
      DO 32 I=1,N
  32  TRV70(I,J)=CF(I)
      DO 31 I=1,N
  31  C470(I,J)=CI(I)+CF(I)
  41  CONTINUE
C
C     TEST FOR SYMMETRICAL LOADING(NSYM=0)
C
      IF(NSYM-1)46,81,46
  46  DO 51 J=1,NROW
      J2=J+NROW
      DO 71 I=1,NCOL
      TRV70(I,J)=TRV70(I,J)+TRV70(I,J2)
  71  C570(I,J)=C470(I,J)+C470(I,J2)
  51  CONTINUE
      GO TO 85
  81  DO 82 I=NROW2,NROW1
      IN=I-NCOL
      DO 82 J=1,NROW
      JN=J+NROW
      TRV70(I,J)=TRV70(IN,JN)
      TRV70(I,JN)=TRV70(IN,J)
      C570(I,J)=C470(IN,JN)
  82  C570(I,JN)=C470(IN,J)
      DO 83 I=1,NCOL
      DO 83 J=1,NROW1
  83  C570(I,J)=C470(I,J)
C
C     REDEFINE NUMBER OF ROWS AND COLUMNS OF CIRCULATION MATRIX
C     FOR ASYMMETRICAL CASE
C
      NCOL=NROW1
      NROW=NROW1
C
C     DETERMINE WEIGHTING OF CIRCULATION BETWEEN THE LEADING AND
C     AFT LIFTING LINES
C
  85  CALL WGT(ALPHEF,WT,AL,WGHT,NROW)
      DO 52 I=1,NCOL
      DO 52 J=1,NROW
      TRVQU(I,J)=TRVQU(I,J)*WT(I)+TRV70(I,J)*(1.-WT(I))
  52  C5(I,J)=C5(I,J)*WT(I)+C570(I,J)*(1.-WT(I))
      DO 80 I=1,NCOL
      DO 80 J=1,NCOL
      C6(I,J)=0.
      DO 80 K=1,NROW
```

```
   80 C6(I,J)=C6(I,J)+C5(I,K)*C5(J,K)
      DO 120 I=1,NCOL
      DO 120 J=1,NROW
  120 C4(I,J)=C5(I,J)
C
C     DETERMINE INVERSE OF CIRCULATION MATRIX
C
      CALL MATINV(C6,NCOL,C5)
      DO 100 I=1,NCOL
      DO 100 J=1,NROW
      C6(I,J)=0.
      DO 100 K=1,NCOL
  100 C6(I,J)=C6(I,J)+C5(I,K)*C4(K,J)
      DO 150 I=1,NCOL
      CIRCLN(I)=0.
      DO 150 J=1,NROW
C
C     TEST FOR SYMMETRICAL LOADING(NSYM=0)
C
      IF(NSYM-1)125,130,125
  125 W(J)=SALPHA*CBETA+Q*(X(J)-XCG)
      GO TO 145
  130 W(J)=SALPHA*CBETA+P*Y(J)
  145 CONTINUE
  150 CIRCLN(I)=CIRCLN(I)-C6(I,J)*W(J)
      DO 160 J=1,NROW
      DWASH(J)=0.
      DO 160 K=1,NCOL
  160 DWASH(J)=DWASH(J)-C4(K,J)*CIRCLN(K)
      DO 161 J=1,NROW
      DWASH(J)=0.
      DO 161 K=1,NCOL
  161 DWASH(J)=DWASH(J)-TRVQU(K,J)*CIRCLN(K)
      DO 162 J=1,NROW
      ALPHEF(J)=ATAN((SALPHA*CBETA+Q*(X(J)-XCG)+P*Y(J)-DWASH(J))/
     1(CALPHA*CBETA-ZCG*Q-R*Y(J)))
      IF(ALPHEF(J)-ALPHA)185,185,180
  180 ALPHEF(J)=ALPHA
  185 CONTINUE
  162 SINAEF(J)=SIN(ALPHEF(J))
C
C     CALCULATE SPANWISE LOADING
C
      DO 171 I=1,NCOL
  171 GAM(I)=CIRCLN(I)*2.*(CBETA+R*Y(I))+CD*SINAEF(I)*SINAEF(I)
     1*CHORD(I)
C
C     CALCULATE NORMAL FORCE
C
      CALL FMINT(GAM,COEF,ETAB,N,XINT,NSYM,0)
      CN=(1.+TR)*(ETAB-ETA0)/2.
      CN=XINT/CN
C
C     CALCULATE PITCHING MOMENT
C
      DO 210 I=1,NCOL
      CIRCL1(I)=CIRCLN(I)*WT(I)
      CIRCL2(I)=CIRCLN(I)*(1.-WT(I))
      XL1(I)=XI0(1)+ETA(I)*TN(1)
```

```
      XL2(I)=XIO(2)+ETA(I)*TN(2)
  210 XPMOM(I)=(CIRCL1(I)*XL1(I)+CIRCL2(I)*XL2(I))*2.*(CBETA+R*Y(I))
     1  +CD*SINAEF(I)*SINAEF(I)*CHORD(I)*(ETA(I)*TNLE+CDXPOS*CHORD(I))
      CALL FMINT(XPMOM,COEF,ETAB,N,XINT,NSYM,0)
      CM=(1.+TR)*(ETAB-ETAO)*REFL/2.
      CM=XINT/CM
C
C     TEST FOR SYMMETRICAL LOADING(NSYM=0)
C
      IF(NSYM-1)211,212,211
C
C     CALCULATE ROLLING MOMENT
C
 212  CALL FMINT(GAM,COEF,ETAB,N,XINT,NSYM,1)
      CMX=(1.+TR)*(ETAB-ETAO)*REFL/2.
      CMX=XINT/CMX
      GO TO 213
  211 CMX=0.0
 213  CONTINUE
      DO 214 I=1,NCOL
 214  ALPH(I)=ALPHEF(I)*57.2958
      WRITE (6,174) P,Q,R
      WRITE (6,186)
      WRITE (6,175) (SPAN(I),I=1,NCOL)
      WRITE (6,176) (GAM(I),I=1,NCOL)
      WRITE (6,177) (ALPH(I),I=1,NCOL)
  170 WRITE (6,220) CN,CM,CMX
C
C     NOW ADJUST ALPHEFFECTIVE FOR NEXT ITERATION ON ALPHA
C
      IF(ALPHA-0.01)192,190,190
  190 DO 191 I=1,NCOL
  191 ALPHEF(I)=ALPHEF(I)*(ALPHA+DALPHA)/ALPHA
  192 CONTINUE
  172 ALPHA=ALPHA+DALPHA
  55  FORMAT(12I6)
  60  FORMAT(8F9.5)
  65  FORMAT(47HODOWNWASH CONTROL POINT OUTSIDE OF END POINTS.,3F13.5)
  174 FORMAT (1H05X,2HP=F9.5,2HQ=F9.5,2HR=F9.5)
  175 FORMAT (1H015HSPAN          10F10.4/(16X10F10.4))
  176 FORMAT (1H015HLOADING       10F10.4/(16X10F10.4))
  177 FORMAT (1H015HEFFECTIVE ALPHA10F10.4/(16X10F10.4))
  186 FORMAT (1H020X,36HSPANWISE LOADING AND EFFECTIVE ALPHA)
  300 FORMAT (1H110X,18HRESULTS FOR ALFA= F9.4,12H, AND BETA= F9.4,10H
     1 DEGREES)
  220 FORMAT (1H031HNORMAL FORCE COEFFICIENT, CN = F9.5/1H0
     140HMOMENT COEFFICIENT ABOUT Y-AXIS , CMY = F9.5/1H040HMOMENT COEFF
     2ICIENT ABOUT X-AXIS , CMX = F9.5)
      STOP
      END


      SUBROUTINE WGT(ALPHEF,WT,AL,WGHT,N)
C
      DIMENSION ALPHEF(20),WT(20),AL(10),WGHT(10)
C
      DO 100 I=1,N
      IF (ALPHEF(I)-AL(2)) 5,10,10
  10  IF (ALPHEF(I)-AL(3))15,20,20
```

```
20   IF (ALPHEF(I)-AL(4))25,30,30
30   IF (ALPHEF(I)-AL(5))35,40,40
40   IF (ALPHEF(I)-AL(6))45,50,50
50   IF (ALPHEF(I)-AL(7))55,60,60
60   IF (ALPHEF(I)-AL(8))65,70,70
70   IF (ALPHEF(I)-AL(9))75,80,80
 5   WT(I)=WGHT(1)-(ALPHEF(I)-AL(1))*(WGHT(1)-WGHT(2))/(AL(2)-AL(1))
     GO TO 100
15   WT(I)=WGHT(2)-(ALPHEF(I)-AL(2))*(WGHT(2)-WGHT(3))/(AL(3)-AL(2))
     GO TO 100
25   WT(I)=WGHT(3)-(ALPHEF(I)-AL(3))*(WGHT(3)-WGHT(4))/(AL(4)-AL(3))
     GO TO 100
35   WT(I)=WGHT(4)-(ALPHEF(I)-AL(4))*(WGHT(4)-WGHT(5))/(AL(5)-AL(4))
     GO TO 100
45   WT(I)=WGHT(5)-(ALPHEF(I)-AL(5))*(WGHT(5)-WGHT(6))/(AL(6)-AL(5))
     GO TO 100
55   WT(I)=WGHT(6)-(ALPHEF(I)-AL(6))*(WGHT(6)-WGHT(7))/(AL(7)-AL(6))
     GO TO 100
65   WT(I)=WGHT(7)-(ALPHEF(I)-AL(7))*(WGHT(7)-WGHT(8))/(AL(8)-AL(7))
     GO TO 100
75   WT(I)=WGHT(8)-(ALPHEF(I)-AL(8))*(WGHT(8)-WGHT(9))/(AL(9)-AL(8))
     GO TO 100
80   WT(I)=WGHT(9)-(ALPHEF(I)-AL(9))*(WGHT(9)-WGHT(10))/(AL(10)-AL(9))
100  CONTINUE
     RETURN
     END

     SUBROUTINE GAUSS(FUNCTN,A,B,C,D,E,N,X1,X2,ANTEG)
C
     DIMENSION X(16),W(16)
C
     IF(K-1968)1,2,1
 1   K=1968
     X(1)=0.005299533
     X(2)=0.027712488
     X(3)=0.067184399
     X(4)=0.122297796
     X(5)=0.191061878
     X(6)=0.270991611
     X(7)=0.359198225
     X(8)=0.452493745
     X(9)=0.547506255
     X(10)=0.640801775
     X(11)=0.729008389
     X(12)=0.808938122
     X(13)=0.877702204
     X(14)=0.932815601
     X(15)=0.972287512
     X(16)=0.994700468
     W(1)=0.013576230
     W(2)=0.031126762
     W(3)=0.047579256
     W(4)=0.062314486
     W(5)=0.074797994
     W(6)=0.084578260
     W(7)=0.091301708
     W(8)=0.094725305
     W(9)=0.094725305
```

```
      W(10)=0.091301708
      W(11)=0.084578260
      W(12)=0.074797994
      W(13)=0.062314486
      W(14)=0.047579256
      W(15)=0.031126762
      W(16)=0.013576230
    2 SUM=0.
      DO 3 I=1,16
      CALL FUNCTN((X2-X1)*X(I)+X1,A,B,C,D,E,N,F)
    3 SUM=SUM+W(I)*F
      ANTEG=SUM*(X2-X1)
  500 FORMAT(8F9.9)
      RETURN
      END


      SUBROUTINE FORM1(X,A,B,C,D,E,N,F)
C
      F=(D*X**N+E*X**(N-1))/SQRT(A*X**2+B*X+C)
      RETURN
      END


      SUBROUTINE FORM2(X,A,B,C,D,E,N,F)
C
      F=X**N/((A*X*X+B*X+C)*SQRT(X*X+D*X+E))
      RETURN
      END


      SUBROUTINE FORM3(X,A,B,C,DUMY1,DUMY2,N,F)
C
      F=X**N/(A*X*X+B*X+C)
      RETURN
      END


      SUBROUTINE LGRANG(X,C,N)
C
      DIMENSION X(10),C(10,10),X1(9),C2(10)
C
      DO 35 I=1,N
      DO 5 J=1,N
    5 C2(J)=1.
      C1=1.
      M1=0
      DO 15 J=1,N
      IF (I-J) 10,15,10
   10 M1=M1+1
      C1=C1/(X(I)-X(J))
      X1(M1)=X(J)
   15 CONTINUE
      C(I,1)=C1
      N1=N
      I1=1
   20 N1=N1-1
      IF (N1) 35,35,25
   25 I1=I1+1
```

```
      DO 30 J=1,N1
      C2(J)=0.
      DO 30 K=J,N1
   30 C2(J)=C2(J)-C2(K+1)*X1(K)
      C(I,I1)=C2(1)*C1
      GO TO 20
   35 CONTINUE
      RETURN
      END


      SUBROUTINE LLINE(X,Y,Z,XI1,XI2,ETA1,ETA2,TN,ALPHEF,BETA,COEF,CI,N)
C
      DIMENSION COEF(10,10),CI(80)
C
      EXTERNAL FORM1
C
      A1=ABS(ETA2)
      A1=A1*TN/ETA2
      TN2=TN*TN
      C1=(X-XI1)*A1
      C2=(Y-ETA1)*A1-X+XI1
      C3=(X-XI1)*A1-(ETA2-ETA1)*TN2-ETA2+Y
      C4=(Y-ETA2)*A1-X+XI2
      A=1.+TN2
      B=-2.*(Y+ETA1*TN2+C1)
      C=(X-XI1)**2+Y**2+Z**2+TN2*ETA1**2+2.*ETA1*C1
      DEN=12.56637*(A*Z**2+C2**2)
      UM1=C2*A/DEN
      UM0=-C2*(Y+ETA1*TN2+C1)/DEN
      C1=C1+Y-ETA1
      SQR1=SQRT((X-XI1)**2+(Y-ETA1)**2+Z**2)
      SQR2=SQRT((X-XI2)**2+(Y-ETA2)**2+Z**2)
      V1=-C1*C2/(DEN*SQR1)
      V2=-C3*C4/(DEN*SQR2)
      N2=N-1
      DO 10 I=1,N
      CI(I)=0.
      DO 5 J=1,N2
      J1=N-J
      AJ1=J1
      CALL GAUSS(FORM1,A,B,C,UM1,UM0,J1,ETA1,ETA2,FCN)
    5 CI(I)=CI(I)-AJ1*FCN*COEF(I,J)
      ETA1N=1.
      ETA2N=1.
      N1=N+1
      DO 10 J=1,N
      N1=N1-1
      CI(I)=CI(I)+COEF(I,N1)*(V2*ETA2N-V1*ETA1N)
      ETA1N=ETA1N*ETA1
   10 ETA2N=ETA2N*ETA2
      RETURN
      END


      SUBROUTINE TRVORT(X,Y,Z,XI1,XI2,ETA1,ETA2,TN,ALPHEF,BETA,COEF
     1,CF,N)
C
      DIMENSION COEF(10,10),CF(80),A11(3)
```

```
C
      EXTERNAL FORM2,FORM3
C
      TN2=TN*TN
      AA=ABS(ETA2)
      AA=AA*TN/ETA2
      BEFCOS=COS(BETA)
      BEFSIN=SIN(BETA)
      ALFCOS=COS(ALPHEF)
      ALFSIN=SIN(ALPHEF)
      C1=BEFSIN/(BEFCOS*ALFCOS)
      C2=ALFSIN/ALFCOS
      DO 22 K=1,N
   22 CF(K)=0.
      ETAA=ETA1
      ETAB=ETA2
      A1=1.+C2**2
      A2=C1**2+C2**2
      A3=2.*C1
      A4=-2.*Y*A1-A3*(X+C2*Z)
      A5=-2.*(C1*Y-C2*Z+A2*X)
      A6=A1*Y**2+A2*X**2+(1.+C1**2)*Z**2+A3*Y*(X+C2*Z)-2.*X*Z*C2
      A=A1+A2*TN2+A3*AA
      C3=XI1-ETA1*AA
      B=A4+2.*C3*AA*A2+A3*C3+A5*AA
      C=(A2*C3+A5)*C3+A6
      D=2.*((C3-X)*AA-Y)/(1.+TN2)
      E=(X**2+Y**2+Z**2-(2.*X-C3)*C3)/(1.+TN2)
      F=AA-C1
      G=C3-X+C1*Y-C2*Z
      SQR=SQRT(1.+C1**2+C2**2)/12.56637
      DEN=SQRT(1.+TN2)*12.56637*SQR
      H=-(1.+C1*AA)*SQR
      AI=(Y+C1*(X-C3))*SQR
      A11(1)=AI
      A11(2)=H
      DO 10 I=1,N
      I1=N-I
      ETAA=ETA1
      IF(Y)1,2,3
    2 ETAA=.005
      GO TO 1
    3 ETAB=Y-.005
      CALL GAUSS(FORM3,A,B,C,D,E,I1,ETAA,ETAB,ANTEG)
      ETAA=Y+.005
      GO TO 13
    1 ANTEG=0.
   13 ETAB=ETA2
      CALL GAUSS(FORM3,A,B,C,D,E,I1,ETAA,ETAB,BTEG)
      ANTEG=ANTEG+BTEG
      INO=MAXO(1,3-I)
      IN1=MINO(2,N+1-I)
      DO 10 J=INO,IN1
      J1=I+J-2
      AJ=N-J1
      DO 10 K=1,N
   10 CF(K)=CF(K)+AJ*A11(J)*ANTEG*COEF(K,J1)
      A11(1)=AI*G/DEN
      A11(2)=(H*G+F*AI)/DEN
```

```
      A11(3)=H*F/DEN
      N1=N+1
      DO 20 I=1,N1
      I1=N1-I
      ETAA=ETA1
      IF(Y)4,5,6
    5 ETAA=.005
      GO TO 4
    6 ETAB=Y-.005
      CALL GAUSS(FORM2,A,B,C,D,E,I1,ETAA,ETAB,ANTEG)
      ETAA=Y+.005
      GO TO 16
    4 ANTEG=0.
   16 ETAB=ETA2
      CALL GAUSS(FORM2,A,B,C,D,E,I1,ETAA,ETAB,BTEG)
      ANTEG=ANTEG+BTEG
      INO=MAXO(1,4-I)
      IN1=MINO(3,N1+1-I)
      DO 20 J=INO,IN1
      J1=I+J-3
      AJ=N-J1
      DO 20 K=1,N
   20 CF(K)=CF(K)+AJ*A11(J)*ANTEG*COEF(K,J1)
      RETURN
      END


      SUBROUTINE FMINT(FX,COEF,ETAB,N,XINT,NSYM,IMX)
C
      DIMENSION FX(20),COEF(10,10),C(20)
C
      NCOL=N-1
      DO 10 I=1,NCOL
      C(I)=0.
C
C     TEST WHETHER NORMAL FORCE(0),PITCHING(0) OR ROLLING(1) MOMENT
C
      IF(IMX)15,5,15
    5 X=1.
      GO TO 25
   15 X=ETAB
   25 DO 10 J=1,N
C
C     TEST WHETHER NORMAL FORCE(0),PITCHING(0) OR ROLLING(1) MOMENT
C
      IF(IMX)80,75,80
   75 XN=J
      GO TO 85
   80 XN=J+1
   85 K=N+1-J
      X=X*ETAB
   10 C(I)=C(I)+COEF(I,K)*X/XN
      XINT=0.
C
C     TEST FOR SYMMETRICAL LOADING(NSYM=0)
C
      IF(NSYM-1)40,50,40
   40 DO 20 I=1,NCOL
   20 XINT=XINT+C(I)*FX(I)
```

```
      GO TO 60
   50 CONTINUE
C
C     TEST WHETHER NORMAL FORCE(0),PITCHING(0) OR ROLLING(1) MOMENT
C
      IF(IMX)51,52,51
   51 DO 94 I=1,NCOL
      IN=I+NCOL
   94 XINT=XINT-(FX(I)-FX(IN))*C(I)/2.0
      GO TO 60
   52 DO 95 I=1,NCOL
      IN=I+NCOL
   95 XINT=XINT+(FX(I)+FX(IN))*C(I)/2.0
   60 CONTINUE
      RETURN
      END


      SUBROUTINE MATINV(A,N,B)
C
      DIMENSION A(20,20),B(20,20),C(20,20)
C
  100 FORMAT(19H0MATRIX IS SINGULAR)
      DO 1 J=1,N
      DO 1 I=1,N
    1 B(I,J)=0.0
      DO 2 I=1,N
      B(I,I)=1.0
      DO 2 J=1,N
    2 C(J,I)=A(J,I)
      DO 6 I=1,N
      IF(C(I,I))24,50,24
   50 DO 21 IZ=I,N
      IF(C(IZ,I))22,21,22
   21 CONTINUE
      WRITE(6,100)
      GO TO 7
   22 DO 23 M=1,N
      C(I,M)=C(I,M)+C(IZ,M)
   23 B(I,M)=B(I,M)+B(IZ,M)
   24 TC=C(I,I)
      DO 3 J=1,N
      C(I,J)=C(I,J)/TC
    3 B(I,J)=B(I,J)/TC
      DO 6 K=1,N
      IF(K-I)4,6,4
    4 T=C(K,I)
      DO 5 L=1,N
      C(K,L)=C(K,L)-T*C(I,L)
    5 B(K,L)=B(K,L)-T*B(I,L)
    6 CONTINUE
      RETURN
    7 STOP
      END
```