

AD-A081 596

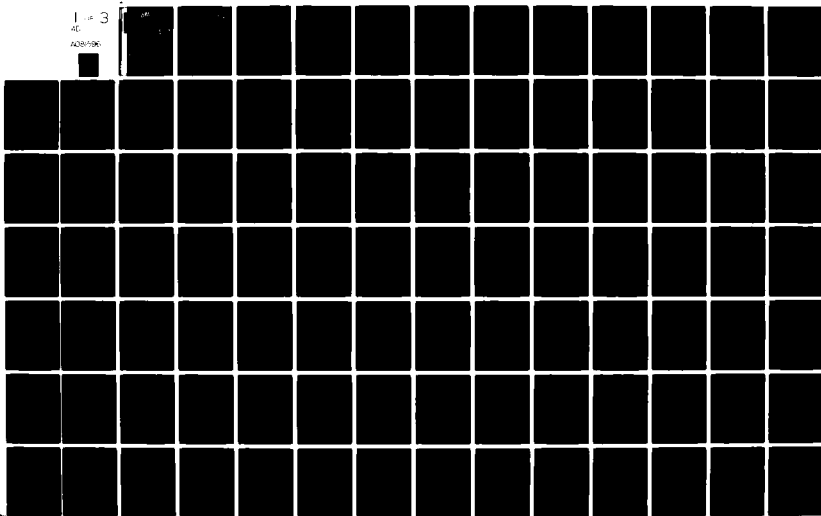
MCDONNELL DOUGLAS ASTRONAUTICS CO HUNTINGTON BEACH CA F/6 9/2
AERODYNAMIC COMPUTER CODE FOR COMPUTING PRESSURE LOADS ON A MIS--ETC(U)
DEC 79 K K WANG N00014-75-C-0462
MDC-68223 NL

UNCLASSIFIED

1 of 3

AC

A081-596



ADA 081 596

LEVEL

12

MDC G8223

**AERODYNAMIC COMPUTER CODE FOR COMPUTING PRESSURE LOADS
ON A MISSILE AT HIGH SUPERSONIC SPEEDS AND GENERALIZED ATTITUDES**

Prepared by Kenneth K. Wang

DTIC
S MAR 6 1980 **D**
C

**MCDONNELL DOUGLAS ASTRONAUTICS COMPANY
5301 BOLSA AVENUE
HUNTINGTON BEACH, CALIFORNIA 92647**

NOVEMBER 1979

**FINAL REPORT
CONTRACT NUMBER N60921-78-C-0203
OCTOBER 1, 1978 TO NOVEMBER 31, 1979**

**Approved for Public Release
Distribution Unlimited**

**Prepared for
NAVAL SURFACE WEAPONS CENTER
WHITE OAK, SILVER SPRING, MARYLAND 20910**

DDC FILE COPY

80 3 6 043

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (6)	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER (9)
4. TITLE (and Subtitle) Aerodynamic Computer Code for Computing Pressure Loads on a Missile at High Supersonic Speeds and Generalized Attitudes.		5. TYPE OF REPORT & PERIOD COVERED Final Report, October 1 827 78-10 November 84, 79, 31
6. AUTHOR(s) (10) Kenneth K. Wang		7. PERFORMING ORG. REPORT NUMBER MDC G8223
8. PERFORMING ORGANIZATION NAME AND ADDRESS McDonnell Douglas Astronautics Company 5301 Bolsa Avenue Huntington Beach, California 92647		9. CONTRACT OR GRANT NUMBER(s) (15) N60921-78-C-0203
10. CONTROLLING OFFICE NAME AND ADDRESS Naval Surface Weapons Center White Oak Silver Spring, Maryland 20910		11. REPORT DATE (11) December 79
12. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 232		13. NUMBER OF PAGES 223
		14. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) For public release, distribution unlimited.		
(14) MDC-G8223		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Aerodynamics Pressure Distribution Structural Analysis High Supersonic Speeds		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) An aerodynamic computer code was developed for calculating the pressure distribution on a missile or its components, e. g., body and wings and subsequently interpolating the results by surface fit at locations as specified for structural analysis using the NASTRAN computer code. The code is valid for high supersonic speeds and generalized attitudes of the missile.		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

389310

PREFACE

This document reports the work performed by McDonnell Douglas Astronautics Company for the Naval Surface Weapons Center, White Oak Laboratory, Silver Spring, Maryland, for the development of an aerodynamic code for computing pressure loads on a missile and/or its components at high supersonic speeds and generalized attitudes. The work was accomplished under contract number N60921-78-C-0203. The Naval Sea Systems Command sponsored the activity.

Capabilities for analyzing a missile traveling at high supersonic speeds ($2.5 < M_\infty < 6$) and generalized attitudes have been incorporated into an existing MDAC aerodynamic code to provide the pressure loads at locations specified by the output of the NSWC computer codes PING and BING for stress analysis by NASTRAN.

The theoretical background of the method for analyzing the high supersonic case is described in this report. The missile attitude has been generalized to include the yaw and roll in addition to the angle of attack. A detailed description of the input parameters is presented together with test cases and operating instructions. A complete listing of the code is included.

Accession For	
NTIS	General
DDC TAB	
Unannounced	
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

CONTENTS

	LIST OF SYMBOLS	vii
	SUMMARY	ix
Section 1	INTRODUCTION	1
Section 2	AERODYNAMIC THEORY—HIGH SUPERSONIC SPEEDS	3
Section 3	GENERALIZED MISSILE ATTITUDE	5
	3.1 Coordinate System Specifying Missile Attitudes	5
	3.2 Effect of Yaw and Roll Angles on Aerodynamic Pressure	5
Section 4	COMPUTER PROGRAM	9
	4.1 Code Capabilities	9
	4.2 Input Description	9
Section 5	TEST CASES	17
	5.1 Combined Angle of Attack and Yaw	19
	5.2 High Supersonic Speeds	21
	5.3 Test-Case Input Cards	28
	5.4 Test-Case Output	31
	5.4.1 Combined Angle of Attack and Yaw	31
	5.4.2 High-Supersonic Flow	50
Section 6	CONCLUSIONS AND RECOMMENDATIONS	61
Section 7	REFERENCES	63
Appendix A	PROGRAM LISTING	65
	DISTRIBUTION LIST	221

LIST OF SYMBOLS

C_p	Pressure coefficient, $(P - P_\infty) \frac{1}{2} \rho U^2$
f_n	Nose finess ratio
M_∞	Free-stream Mach number
x, y, z	Coordinates
α, β, γ	Angle defining missile attitude (see Fig. 1)

SUMMARY

This report represents the work performed under contract number N60921-78-C-0203 for the Naval Surface Weapons Center, White Oak Laboratory, Silver Spring, Maryland, under Naval Sea Systems Command sponsorship for the development of an aerodynamic code for computing pressure loads on a missile at high supersonic speeds and generalized attitudes.

A number of subroutines needed for computing the aerodynamic pressure at high supersonic speeds ($2.5 < M_\infty < 6$) have been developed and incorporated into an existing MDAC aerodynamic code for analyzing the pressure loads on a missile. For treating the missile attitude in the general case, the code was extensively modified to reflect the required changes in the input format, code structure, and output results. After the calculations are made, the code generates the pressure distribution on the missile at the centroid of finite elements as given. Results are produced as punched card output in a NASTRAN bulk-data format.

Section 1 INTRODUCTION

Development of an aerodynamic code (Reference 1) for the calculation of the pressure load on a missile at the centroid of finite elements generated by the NSWC computer code PING and BING (References 2 and 3) was completed in 1978 (contract number N60921-076-C-0164). Because it was based on the numerical solution of a set of linearized governing equations for the fluid flow, its applications are restricted to the subsonic and low-supersonic ($M_\infty < 2.5$) region. With recent improvements in missile technology, it was deemed useful to extend its range of validity to higher supersonic speeds. Additionally, the limitation on missile-attitude changes to angle-of-attack excursions needs to be removed to include the attitude changes in both yaw and roll. The present effort was undertaken to meet these requirements in order to provide the necessary tool for structural analysis.

To minimize the necessary effort, the earlier code was employed as the base into which the high-supersonic and the generalized-missile-attitude capabilities were incorporated. Functions of the original code such as input management, coordinate specification, and transform and output generation were used with a minimum of changes.

Section 2

AERODYNAMIC THEORY - HIGH SUPERSONIC SPEEDS

For high supersonic speeds, the linearized method based on small perturbations for lower Mach numbers ($M_\infty < 2.5$) can no longer be employed for analyzing the pressure loads on missiles. Although exact numerical analyses such as the method of characteristics or finite-difference methods are available, the developmental and computing effort required for their application to the subject problem eliminated them from consideration.

Because it was considered to be the most suitable for the scope of the present effort and for possible future applications, the shock-expansion method was selected for calculating the pressure load on missiles at high supersonic speeds. An explanation of this method follows.

Since the introduction by Epstein (Reference 4) in 1931, the shock-expansion method has been extensively developed by many researchers. The generalized shock-expansion method predicts the pressure distribution for flow about pointed bodies of revolution with reasonable accuracy. In particular, the shock-expansion method (Reference 5) modified for the case where the hypersonic similarity parameter (M_∞ / f_n , ratio of free stream Mach number to nose fineness ratio) is close to 1, suits the present application; its coded version (Reference 6) is used with appropriate modifications.

The basic approach and theoretical derivation of the method are described in detail in Reference 5 and will not be repeated here. However, a brief description of the calculational procedure is presented in the following sections. The method starts with the cone solution for bodies of revolution or the oblique-shock solution for two-dimensional flow. For pointed bodies of revolution, the pressure coefficient C_p after the shock and for each subsequent segment is obtained as the sum of two parts. One represents the pressure coefficient at zero incidence (C_{p0}), and the other represents

the pressure coefficient ($C_{p\alpha}$) due to the incidence α . The zero-incidence pressure coefficient C_{p0} is computed using either the second-order theory or that developed in Reference 7, depending on the value of the parameter M_∞/f_n .

The pressure coefficient $C_{p\alpha}$ for the finite incidence case is calculated using the following relations (References 8 and 9):

$$C_{p\alpha} = \left[A_1 T + A_2 T/M_\infty^2 - A_3/M_\infty^2 \right] \left(\frac{\alpha}{\mu} \right) + \left[A_4 T + A_5 T/M_\infty^2 + A_6/M_\infty^2 \right] \left(\frac{\alpha}{\mu} \right)^2 \quad (1)$$

where

$$A_i = a_{0,i} + a_{1,i} \cos \phi + a_{2,i} \cos 2\phi \quad (i = 1, 2, \dots, 6)$$

$$T = \sin B\mu \tan \mu$$

The constants B and $a_{j,i}$ are those listed in Reference 9. μ and ϕ are the half-cone angle and the meridian angle, respectively.

The pressure coefficient $C_p = C_{p0} + C_{p\alpha}$ obtained at each step of computation is used for calculating the pressure at the end of each segment. As expressed in Reference 5:

$$p_3 = p_c - (p_c - p_2)e^{-\eta} \quad (2)$$

where

p_3 = pressure at the end of segment.

p_c = pressure of equivalent cone = $\frac{1}{2} \gamma M_\infty^2 p_\infty C_p + p_\infty$.

p_2 = pressure result from either 2-D expansion or compression between segments.

η = function of pressure gradient and streamwise distance.

For 2-D flows, either oblique shock or expansion relation is used depending on the flow-deflection angle. The pressure load at the centroid of each panel is calculated by tracing the streamline over the surface of the wing.

Section 3

GENERALIZED MISSILE ATTITUDE

For specifying the attitude of a missile in flight, its pitch, yaw and roll angles are usually required. An aerodynamic analysis limited to pitch-plane motions will be unrealistic and of rather restricted use. Extending the MDAC aerodynamic code to include the effect of yaw and roll expands its range of application.

3.1 COORDINATE SYSTEM SPECIFYING MISSILE ATTITUDES

The coordinate system used for defining the pitch angle α , the yaw angle β , and the roll angle γ of the missile in relation to the freestream-flow vector $\bar{V}$ is shown in Figure 3-1. The successive rotation required to transform from the flow-field coordinate system (X_1, Y_1, Z_1) to that of the missile (X_a, Y_a, Z_a) is described as follows. First, the coordinate (X_1, Y_1, Z_1) is rotated about the Y axis by an angle α and forms the intermediate coordinates (X_2, Y_2, Z_2) . Next, it is rotated about the Z_2 axis by an angle β' to form the coordinates (X_3, Y_3, Z_3) . Finally, a rotation about the X_3 axis by an angle γ leads to the aerodynamic coordinates (X_a, Y_a, Z_a) for aerodynamic analysis of a missile.

As shown in Figure 3-1, either α , β and γ or α , β' and γ can be used for defining missile attitude. For the present case, the first definition was adopted, i.e., the pitch angle α , the yaw angle β , and the roll angle γ . Note that both α and β are projections of angles in the X_1 - Z , and X_1 - Y planes, respectively. In other words, they are defined in terms of angle in the flow or inertial-coordinate system. The reason for their selection will become clear when the method for analyzing the pressure loads is discussed.

3.2 EFFECT OF YAW AND ROLL ANGLES ON AERODYNAMIC PRESSURE

When a missile is considered as a whole rather than as separate components, accurate analysis of the aerodynamic pressure on its surface in the presence

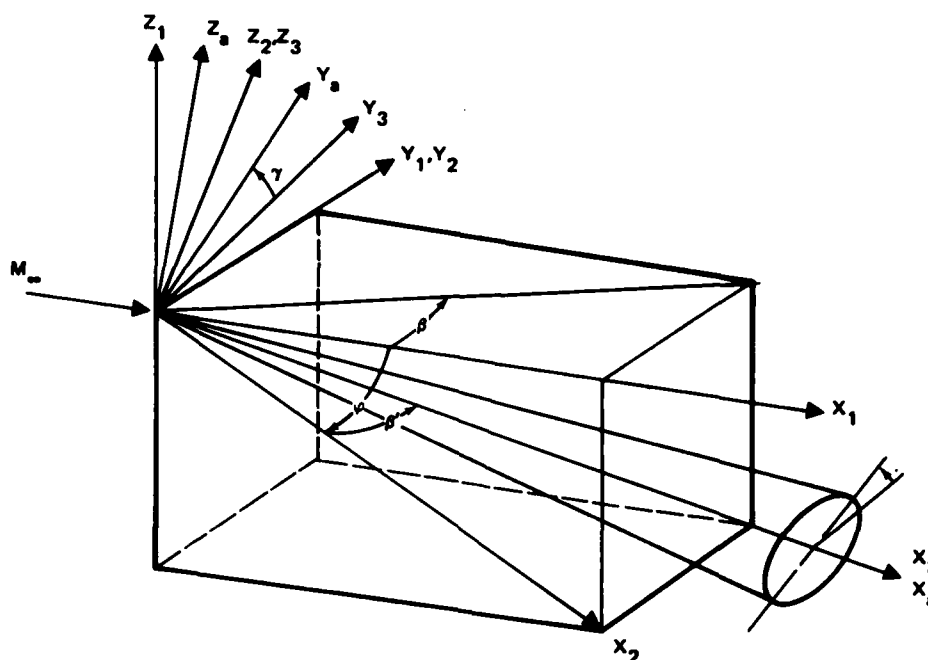


Figure 3-1. Missile Attitude Coordinates

of yaw and/or roll becomes somewhat laborious. A simpler method that provides results adequate for present needs must be developed to reduce the expected effort. For this purpose, a scheme consistent with the basic approach used in analyzing the pressure load was adopted for incorporating the additional effect of yaw and/or roll.

As described in Reference 1, the approach employed is based on the small-perturbation assumption which requires that both angle of attack and yaw be small. This consideration permits the linearization of the governing fluid-dynamic equations and the analysis of the pressure load on the missile as individual components (body and wings). Their interactions, however, are subsequently calculated and added to that of the appropriate component. As described below the effect of yaw and roll on the aerodynamic load on the missile is similarly analyzed taking advantage of this approach.

The combined effect of angle of attack α and yaw β on the body can be viewed as that of an equivalent angle of attack α' with a rotation of the body about its axis by an angle γ' . From the geometry, one has

$$\alpha' = \arccos \left[1.0 / \sqrt{1 + \tan^2 \alpha + \tan^2 \beta} \right] \quad (3)$$

$$\gamma' = \arctan \left[\tan \beta / \tan \alpha \right] \quad (4)$$

As defined, the effect of roll γ represents only a shift in the angular distribution of the pressure load on the body and is incorporated by substituting the algebraic sum of γ and γ' or γ'' for γ' in the analysis.

The combined effect of angle of attack α , yaw β and the roll γ on the wing pressure load is incorporated by an apparent angle of attack α'' which is

$$\alpha'' = \arcsin \left[\sin \gamma'' \sin \beta' \cos \alpha + \cos \gamma'' \sin \alpha \right] \quad (5)$$

where

$$\beta' = \arccos \left[\sqrt{(1 + \tan^2 \alpha)} / (1 + \tan^2 \alpha + \tan^2 \beta) \right] \quad (6)$$

The advantage of the present approach for incorporating the effect of yaw and roll is that it allows maximum use of the output of the existing analysis and consequently minimizes the effort required to develop a more versatile code. The practical necessity of code reorganization, core-storage rearrangement, and appropriate assembly of additional subroutines have, however, been time consuming, and repeated computer runs were required for program checkout.

Section 4

COMPUTER PROGRAM

The MDAC aerodynamic code (Reference 1) previously developed for NSWC under the contract number N60921-76-C-0164 has been extensively modified to incorporate the high-supersonic-speed and the generalized missile-attitude capabilities. Using the output of NSWC computer codes PING and BING which specify the finite-element and grid-point distribution on the missile, the present code calculates the desired pressure load at the centroid of each element. At the user's option, the pressure load may be generated as punched-card output in the bulk-data format of NASTRAN for subsequent structural analysis.

4.1 CODE CAPABILITIES

The present code as modified is capable of calculating the pressure load for a missile or its components (body or wing) at either subsonic or supersonic speeds. The pitch-plane restriction on its attitude has been removed to allow both yaw and roll in addition to pitch-plane motions. Due to the small-disturbance assumption, the code is valid for slender bodies of circular or near circular cross-section at small angle of attack or yaw.

The code requires a field length of 150,000 when used on the CDC CYBER 174 computer with KRONOS 2.1 system. For subsonic or low-supersonic cases, approximately 3 minutes computing time is required. For higher supersonic ($M > 2.5$) cases, much less computing time (< 1.0 min.) is required.

4.2 INPUT DESCRIPTION

Input parameters for numerical computation have been modified as needed, based on the MDAC aerodynamic code previously developed. Only minor changes were required for specifying either the high-supersonic-speed case or the generalized-attitude case. However, for user's convenience, a complete list of input parameters has been included in the following description:

XMACH	Mach number of flow. Higher supersonic routine employed when greater than 2.5, default value = 2.0.
PINF	Free stream pressure, default value = 14.7 psi.
DADEG	Incremental angle of attack of missile in degrees, default value = 0.
SID	Case identification number for each missile component (up to 20 allowed); default values have been set at 1. Also used for selecting the sign convention for pressure output. Code uses the accepted rule for pressure, i.e., positive acting toward the surface and negative acting away from the surface. For opposite-sign convention, set SID(L) as negative for pressure output on the desired component L (in the order of body, wing surfaces, upper or lower).
YAW	Angle of yaw in degrees with respect to the flow.
BRØLL	Missile roll about body axis (degree).

The following input parameters are required for the case where missile body is included:

NBPFL	Number of body profile stations for specifying body geometry inputs $r = r(x)$; up to 51 stations allowed.
$XB \leq 51$	Axial coordinate of body stations in the direction of flow for specifying the body profile.
$RB \leq 51$	Radial coordinate of body profile at corresponding body stations.
$ZD \leq 51$	Body camber at corresponding body stations.
ARB	Angle of attack of missile body in degrees, default value = 0.

As is similar to the overall organization of the code, the input parameters and data have been organized into two main groups in accordance with their assigned functions. The first group contains all the parameters needed for the specification of case selection, missile configuration, and flow conditions. They form the input parameter of the namelist UNIFID. The second group

specifies the finite-element and grid-point system as generated by the NSW C computer codes BING and/or PING. It includes the specification for the coordinate system used for the finite elements.

For the convenience of the user, card format for inputs are described in their sequential order as required by the code.

A. Aerodynamic Control Parameters.

Card 1 Title and case description, format (20A4), alphanumeric.

Card 2 \$UNIFID, begins at Column 2 with the symbol \$.

Card 3 And additional cards if necessary, contain the following as input:

ICASE = 1 Wing only.

= 2 Body only.

= 3 Complete missile.

NTRANS Number of coordinate transformations required to convert the coordinates for the finite element to the coordinates used in the aerodynamic code. Two transformations are allowed for each missile component, in the following order; body, wings, and nacelles.

IPUNCH = 0 No punched card output, default value.

= 1 Punched card output requested.

IRW = 0 Computation results are not saved on tape, default value.

= 1 Aerodynamic computation results are saved on tape 12 for restart, first run only.

= 2 Restart run, bypass all aerodynamic computations and begins computation at the start of pressure interpolation.

POLAR Number of incremental angles of attack, default value = 0.

The following inputs are required for the case where the missile wing is included:

ISOLID	= 0	Wing of built-up construction, pressure load on both upper and lower wing surface generated, default value.
	= 1	Wing of solid section, sum of the pressure loads on wing surface generated.
IFORM	= 0	Wing of built-up construction, both upper and lower surface height (ZWI) are given at identical locations (XWI, YWI).
	= 1	Wing of built-up construction, upper and lower surface height (ZWI) are not given at identical locations.
	= 2	Flat wing surface.
NWPI		Number of coordinate points specify the wing surface contour (≤ 30). Code allows a maximum of 20 wing surfaces (upper and lower). Each wing surface must be defined in the form of equation of a surface, $ZWI = f(XWI, YWI)$. A maximum of 30 points are permitted with the provision that the first four points are restricted to the specification of the corners of the wing only. The coordinates for the corners must be given in the following order. Starting with the innermost point of the leading edge, the remaining corners are to be given in a clockwise order viewing from the above.
NWING		Number of wings or tails. A maximum of 10 is allowed, default value = 0.
XWI(i, j), YWI(i, j), ZWI(i, j)		Wing-surface coordinates where indices i and j designate the surface coordinate point and wing surface, respectively.
DIHED		Dihedral angle, degrees.
PIVOT		Indicates dihedral, 0 for no dihedral, and >0 for dihedral present.
ARW		Angle of attack of wing in degrees, with respect to the body axis if complete missile is considered, default value = 0.

The following input parameters are required for the complete-missile case (ICASE = 3) except for high-supersonic-speed ($M > 2.5$) case:

NWPANL	Number of wings directly connected to the body.
XLG	Axial distance at the intersection of wing leading edge and body for each wing.
XTG	Axial distance at the intersection of wing trailing edge and body for each wing.

Card 4 \$END Begin at column 2 the symbol \$ for ending input.

B. Finite-Element and Grid Inputs. The second group of input cards are arranged immediately following Card 4, as follows:

Cards 5 and 6 Coordinate-system specifications in the format of NASTRAN bulk-data deck [page 2.4-49 - 2.4-54, NASTRAN User's Manual, NAS SP-221(01), 1972]. Three position vectors - A, B, and C - are used to define the coordinate system. The first defines the origin, the second defines the Z axis, and the third defines a point in the XZ plane.

For Card 5 the following format is used:

Col 1-8, Coordinate system CØRDJRbb for rectangular, CØRDJCbb for cylindrical, and CØRDJSbb for spherical, where J identifies the coordinate system numbers.
Col 9-16 Coordinate identification number J, integer.
Col 17-24, Reference coordinate system, integer, optional.
Col 25-32, 33-40, 41-48, Components of vector A (3F8.2).
Col 49-56, 57-64, 65-72, Components of vector B (3F8.2).

For Card 6:

Col 1-8, blank.
Col 9-16, 17-24, 25-32, Components of vector C (3F8.2).

Cards 5 and 6 are to be repeated as many times as there are coordinate systems (NTRANS) required for specifying the finite element. They are to be arranged in the order of body and wings.

The finite element and grid specifications are arranged immediately following the coordinate-system input in the same order, i.e., body, wings.

Card 7: Specifies the number of grid cards (NGRIDP) and element cards (NE), Format (2I10).

The grid and the element cards as generated by the NSWC computer-code BING are sorted into two separate groups. They are then placed immediately after card 7 with the grid cards in front.

Some remarks regarding the input preparation need to be made here to facilitate the use of the aero code.

For the body-only case, inputs of the body profiles, expressed as $r = r(x)$, for aero-panel generation must be in the aero-coordinate system. It is required that x be the axial distance from the tip and r the radius of the body cross section (Figure 4-1). For the finite element and grid point, their relationship to the basic coordinate system and the basic coordinate system to aero-coordinate system are to be specified in the coordinate input card immediately following the namelist input. The grid card and the element card are to be separated and placed after the coordinate system specification, with a card specifying their numbers placed behind the coordinate card.

The input for the wing-only case requires the specification of both the wing planform and its surface contour. The input for the description of the wing planform requires the coordinates of corner points and break points. For each individual wing, inputs for those of quadrilateral shape are their four corner coordinates. The codes have allocated the first four values of the coordinates XWI, YWI and ZWI for each wing or wing region, arranged in a counterclockwise order starting from the leading-edge corner or break point. In the case of triangular wings, there are only three corner points. To satisfy the same requirement, the apex or wing tip is to be considered as two coincident points of the same coordinate. The inputs are to be arranged like those of

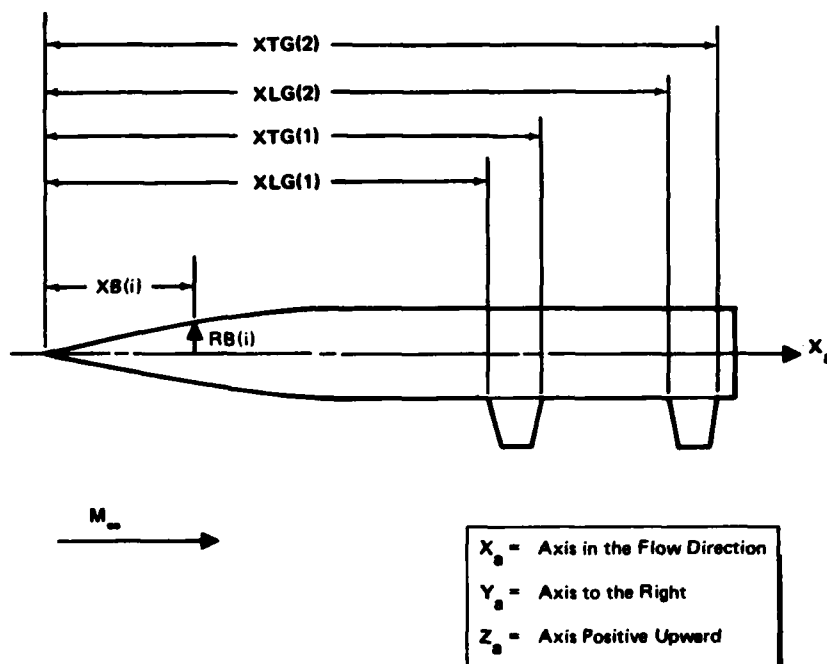


Figure 4-1. Body Profile Input

the quadrilateral shape, with leading-edge body-corner point as the starting point. Wings with complex shapes (see Figure 4-2) must be subdivided into appropriate quadrilateral or triangular regions and the inputs prepared accordingly. The only requirement is that the control chords, apart from either the leading edge or the trailing edge, must be in the streamwise direction.

All inputs for the wing surface are to be given in the form of the equation of surface $Z = f(x, y)$ in terms of XWI, YWI and ZWI. Starting with the fifth value, XWI(5,1), YWI(5,1), ZWI(5,1), their total number may not exceed 26 points. For wings with flat surfaces, only the first four points are necessary.

Inputs defining wing thickness, camber and twist have been eliminated to reduce the input preparation effort. The code generates the required values of camber, thickness slope, and twist angle for the wing using the existing surface-fit scheme from the input.

Similar to the body-only case, the finite-element and grid-point coordinate-system cards are placed immediately behind the namelist input. They are

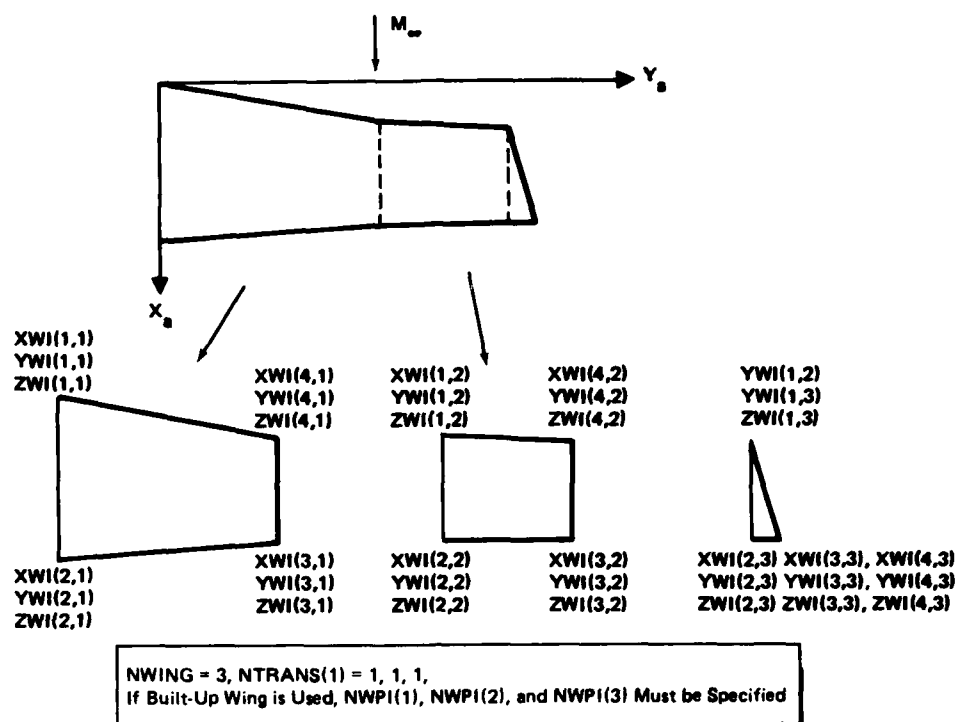


Figure 4-2. Subdivision of Complex Wing

followed by a card specifying the number of grid cards and element cards. The grid-point and element card as generated by PING occupies the last position in the input.

The coordinate system used for the wing can be any one of three systems, i.e., rectilinear (x, y, z) , cylindrical (r, θ, z) and spherical (r, θ, ϕ) . In terms of the input parameters, they are represented by XWI, YWI and ZWI, respectively, for all three systems.

For the complete-missile case, the input can be considered as a combination of these two cases, i.e., the body-only and the wing-only cases. Only the parameters ICASE and NTRANS need be corrected to reflect the changes.

Both coordinate-system cards and finite-element cards are to be grouped together in the order of body, wings, and nacelles.

As a result of the approach used for the analysis, the computations for the high-supersonic-flow case are considerably simpler. Accordingly, the code treats the complete missile as a rule.

Section 5 TEST CASES

Test cases were selected for aerodynamic analysis by the present code for the purpose of verifying the theoretical approach and code modifications. In Reference 10, the experimental results from tests conducted for a model with delta planform wings and cylindrical body at supersonic speeds ($M_\infty = 2.00$ and 2.58) are available and suitable for our purposes. The model configuration with dimensions is shown in Figure 5-1. Location of the pressure taps for wing and body are shown in Figures 5-2 and 5-3, respectively. Tests were made for angles of attack and yaw varying from 0° to 5° .

9CR70

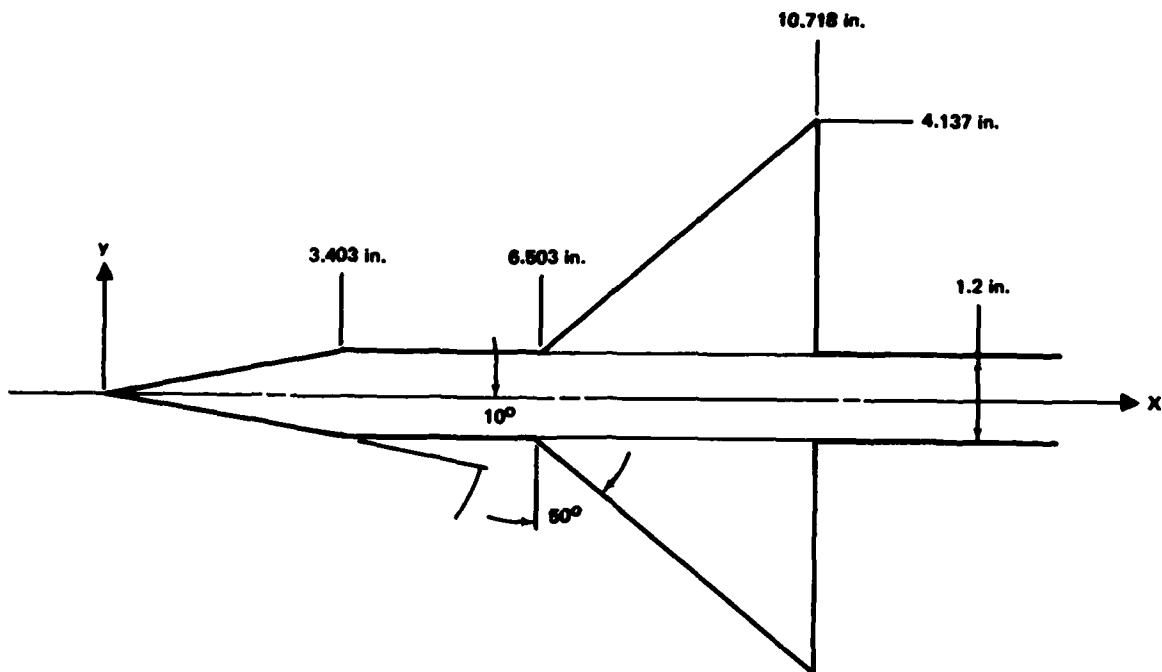
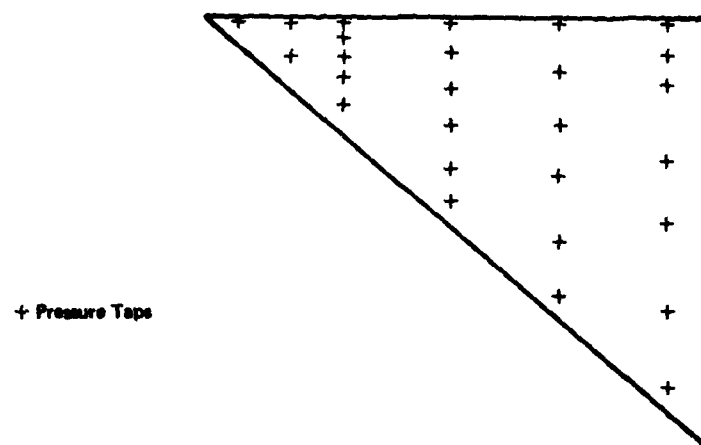


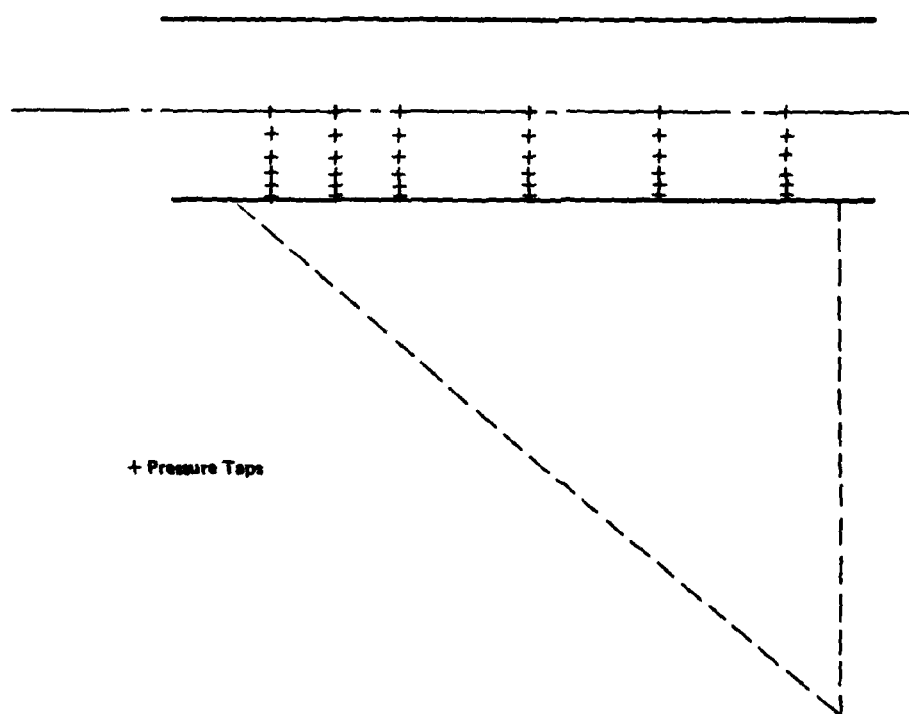
Figure 5-1. Test Model (Reference 10)



9CR70



Fig. 5-2. Wing Plan Form and Pressure Tap Locations



9CR70

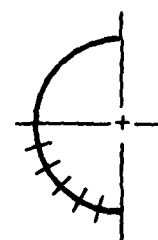


Fig. 5-3. Missile Body and Pressure Tap Locations

MCDONNELL DOUGLAS

To demonstrate the ability of the code in analyzing the pressure load on a missile at high-supersonic speed and generalized attitude, computations were made for missiles flown at (a) combined angle of attack (5°) and yaw (5°) at Mach 2.0 and, (b) high-supersonic case at Mach 2.58 and 5° angle of attack.

5.1 COMBINED ANGLE OF ATTACK AND YAW

The following is a list of input cards, in sequence:

(I) Title Card.

Descriptive title of the case.

FORMAT: (20A4).

(II) Namelist UNIFID input.

Aerodynamic control parameters and geometries of missile.

FORMAT: Real or integer in accordance to the variable.

Test Case:

ICASE = 3	Complete missile.
NTRAN(1)=1, 1	No intermediate coordinate system for wing or body.
ISOLID = 1	Solid wing, sum of upper and lower pressure.
NBPFL = 15	Number of input points for body profile.
NWPI(1)=4	Number of input points for wing surface.
NWING=1	One wing section.
IFORM(1)=2	Flat-wing surface.
ARB = 5.0	Angle of attack of missile.
YAW = 5.0	Angle of yaw of missile.
XMACH = 2.0	Flow Mach number.
NWPANL = 1	Number of wings directly connected to the body.
XLG(1)=6.053	Axial distance of the inner leading edge.
XTG(1)=10.718	Axial distance of the inner trailing edge.

For body profile:

XB(1)=0.0, 0.5, 1.0, 1.5, 2.0, ...

RB(1)=0.0, 0.0881, 0.1763, 0.2664, 0.3526, ...

For the wing surface, only four points are needed for a solid wing:

XWI(1,1)=6.503, 10.718, 10.718, 10.718.

YWI(1,1)=0.6, 0.6, 4.137, 4.137.

ZWI(1,1)=0.0, 0.184, 0.0, 0.184.

(III) Coordinate Card.

Notes: The coordinate cards are needed for input of vectors A, B, and C which are used in defining the relation between the coordinate systems of the finite element (X, Y, Z) and the aero codes (X_a , Y_a , Z_a). In accordance with NASTRAN, vector A defines the origin, vector B defines the Z axis, and vector C defines the X-Z plane.

FORMAT:

NASTRAN bulk-data deck, first card (A8, 2I8, 6F8.2) and second card (A8, 3F8.2).

Test Case:

Two sets of coordinate cards are required, one for the body and one for the wing. The body finite-element coordinate system has its origin coincident with the aero code and its Z axis coincident with the X axis of the aero code as shown in the sketch. Hence, the vector A has component (0.0, 0.0, 0.0), vector B has component (1.0, 0.0, 0.0), and vector C has component (1.0, 0.0, 1.0).

The finite-element coordinate system for the wing, as shown in the sketch, requires the input of vector A with component (0.0, 0.0, 0.0), vector B (0.0, 0.0, 1.0), and vector C (1.0, 0.0, 1.0).

(IV) Finite-Element Cards.

Notes: The finite-element and grid-point cards as generated by BING and PING are used for defining element coordinates. They must be divided into two separate groups with the grid-point card preceding the element card. A card specifies that the numbers of grid card and element card in a format (2I10) must be placed before them.

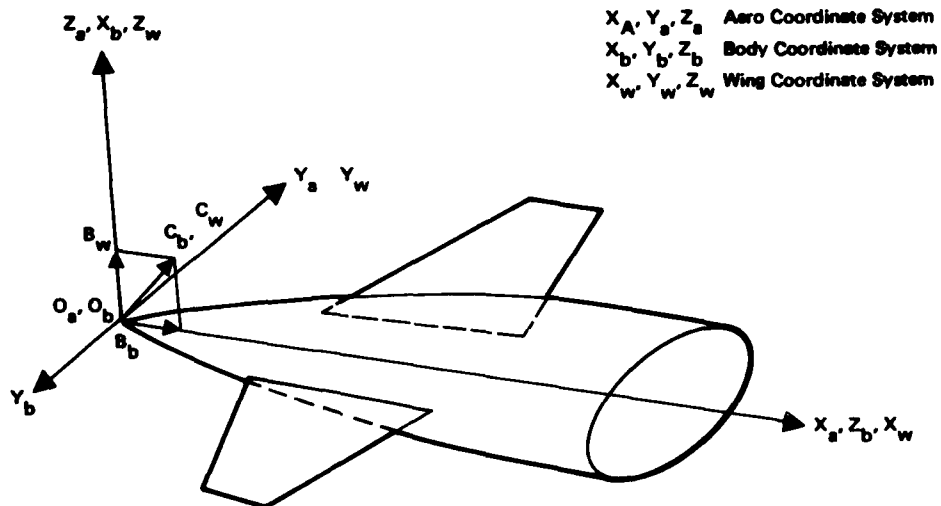
FORMAT:

For grid and element number card (2I10), grid and finite-element cards must be the same as generated by BING or PING.

Test Case:

As shown in the input listing, first set of finite-element grid points and element cards are that of the body and the subsequent set is for the wing surface.

9CR70



Results of Computations

Using the inputs as described, computations were made for the pressure load on the body and wing of the missile tested in Reference 10. In Figure 5-4, the pressure load on the body at the meridian angles of 45° and 135° were compared with that of the experiment. Good correlation with the experimental measurements was obtained. In Figure 5-5, comparisons with the experiment were made for the pressure load on the wing and the results were satisfactory.

It is seen that the code can be used for analyzing the aerodynamic loads on missile at attitudes that may include yaw and roll. Its applicability is therefore considerably extended.

5.2 HIGH SUPERSONIC SPEEDS

The input for this case differs from the previous case only in the Mach number, i.e., 2.58 instead of 2.0. All the remaining input parameters have been retained for this computation and hence will not be repeated here.

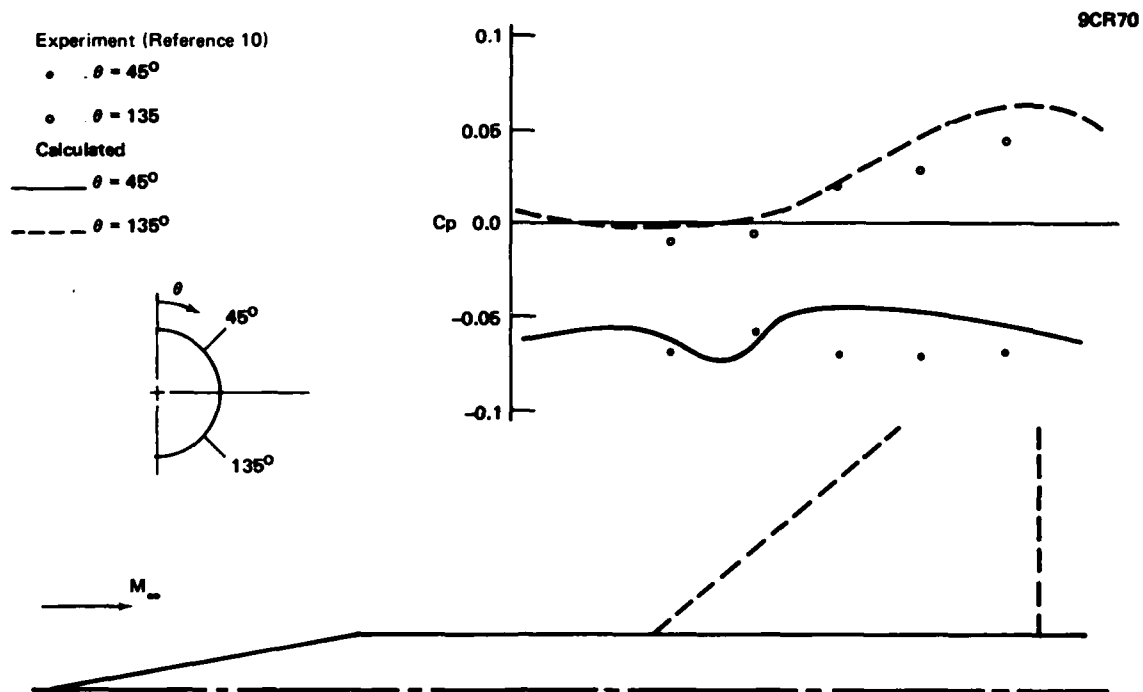


Fig. 5-4. Pressure Distribution on Body at Mach 2.0, 5° Angle of Attack and 5° Yaw

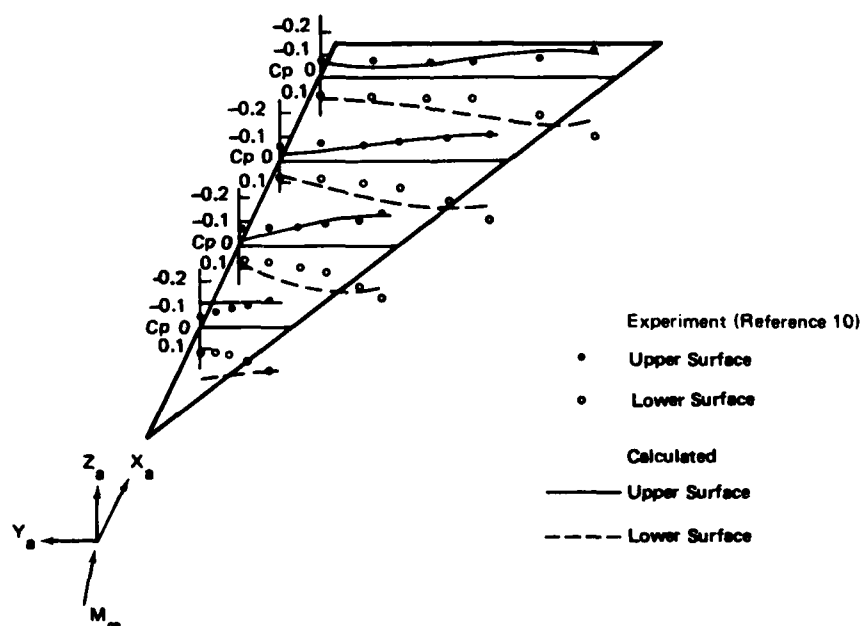


Figure 5-5. Pressure Distribution on Wing at Mach 2.0, 5° Angle of Attack and 5° Yaw

Computational Results

Results of computations for flow at Mach 2.58, illustrated in Figure 5-6 for the body and Figure 5-7 for the wing, show that the pressure distribution on the body correlates poorly with the test data. It is believed that this results from a substantial effect of the wing on the body. This effect should diminish with increasing Mach number. The pressure distribution on the wing surface appears to be acceptable over most of the wing surface.

It should be pointed out that due to the scarcity of experimental data for pressure distributions on complete missiles, the test case described was computed for a Mach number value slightly above the lower limit for the high-supersonic-speed analysis. However, for bodies of simple geometries (circular cones and triangular wing), the test cases were computed at Mach numbers 6.85 and 5.05, and the results correlated rather convincingly. In Figure 5-8, pressure coefficients on circular cone with half-cone angle of 12.5° at angles of attack of 3° , 6° , and 12° were calculated, and the results were very good when compared with available data. In Figures 5-9 and 5-10, pressure coefficients and shock shape for an ogive at an angle of attack were

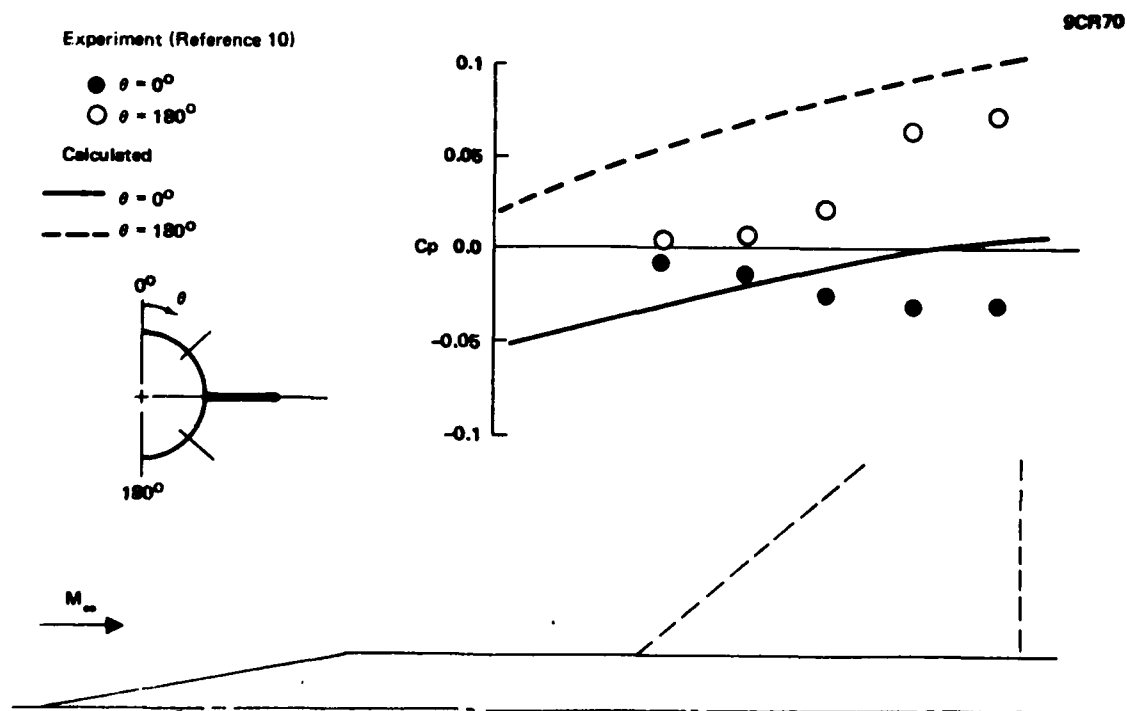


Figure 5-6. Pressure Distribution on Body at Mach 2.58, 5° Angle of Attack

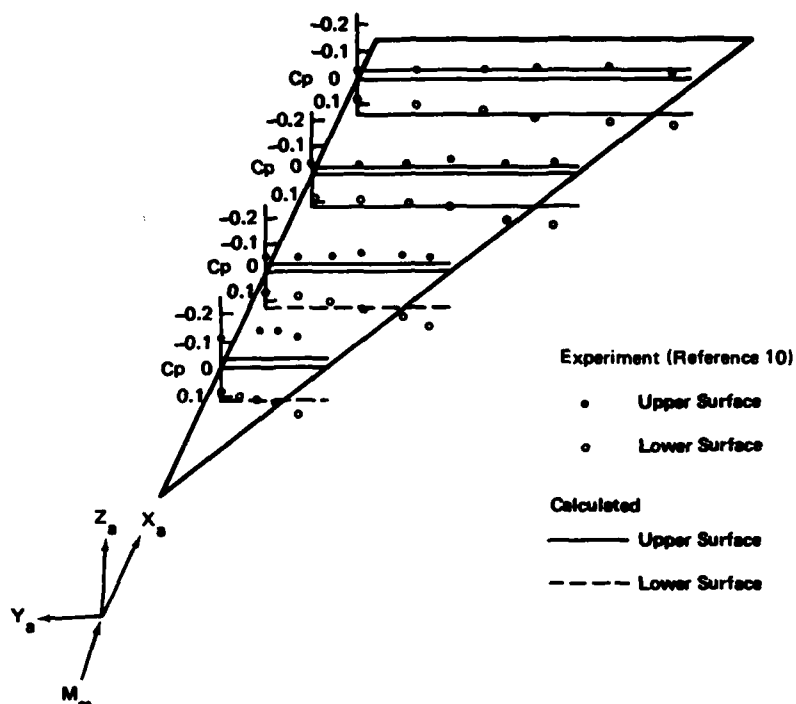


Figure 5-7. Pressure Distribution on Wing at Mach 2.58, 5° Angle of Attack

computed, and the results were satisfactory as shown. In Figure 5-11, the pressure on a 75° sweep, triangular wing was calculated using the shock-expansion method. Comparisons with data show good correlation.

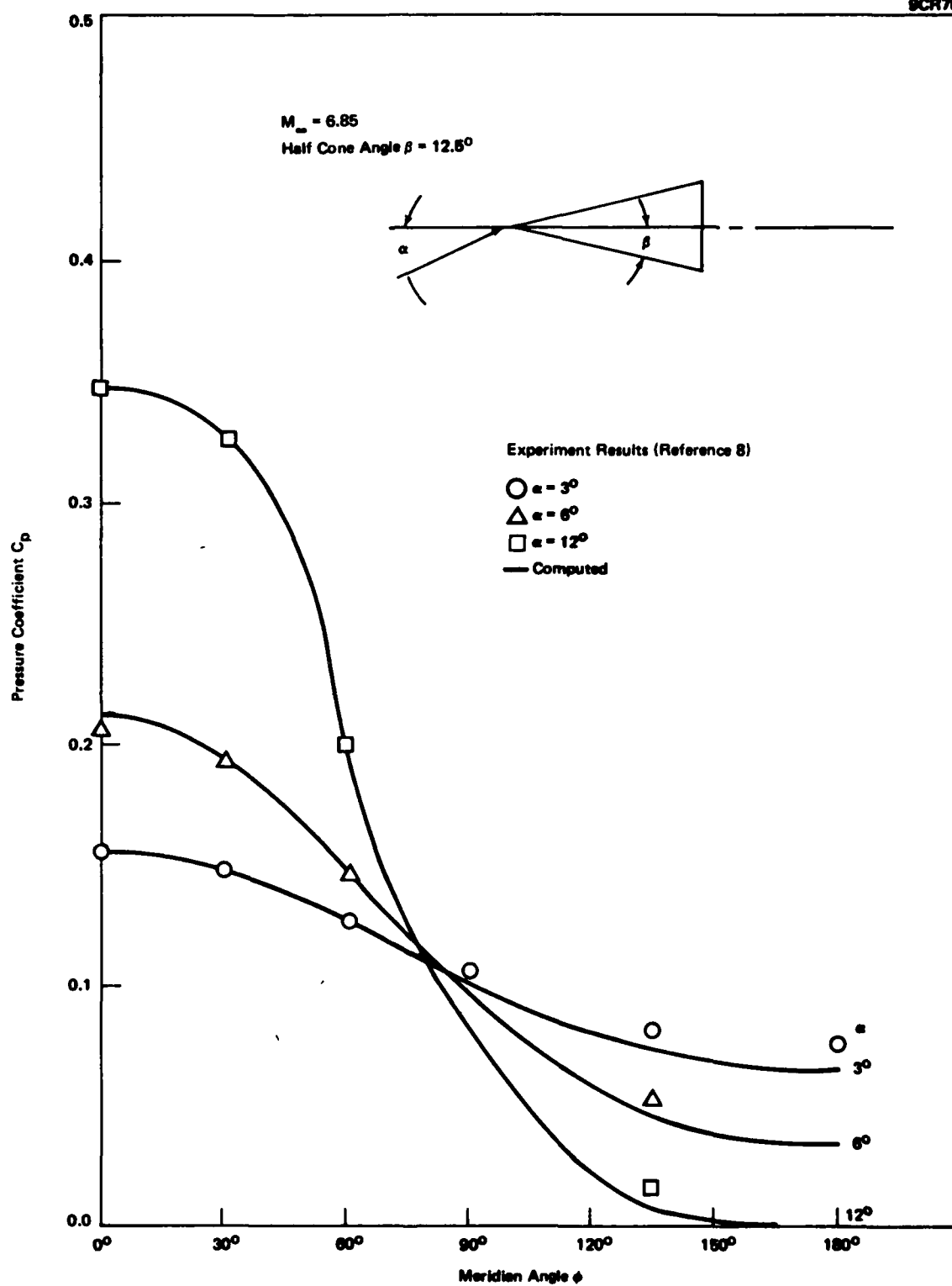


Figure 5-8. Pressure Coefficient for Cone

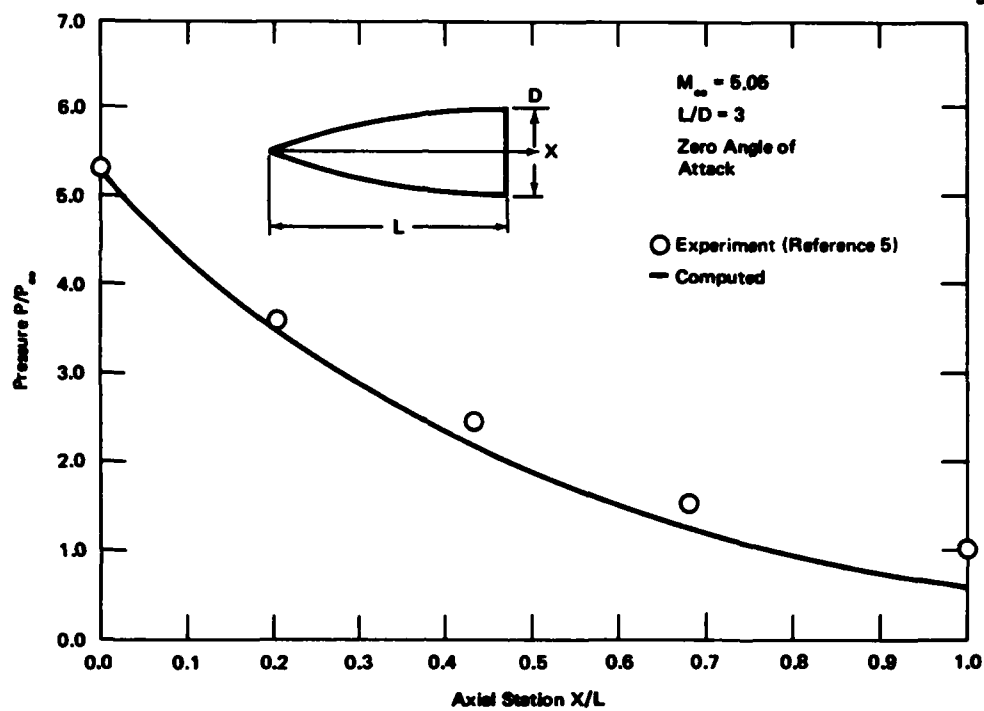


Figure 5-9. Pressure Distribution for Ogive

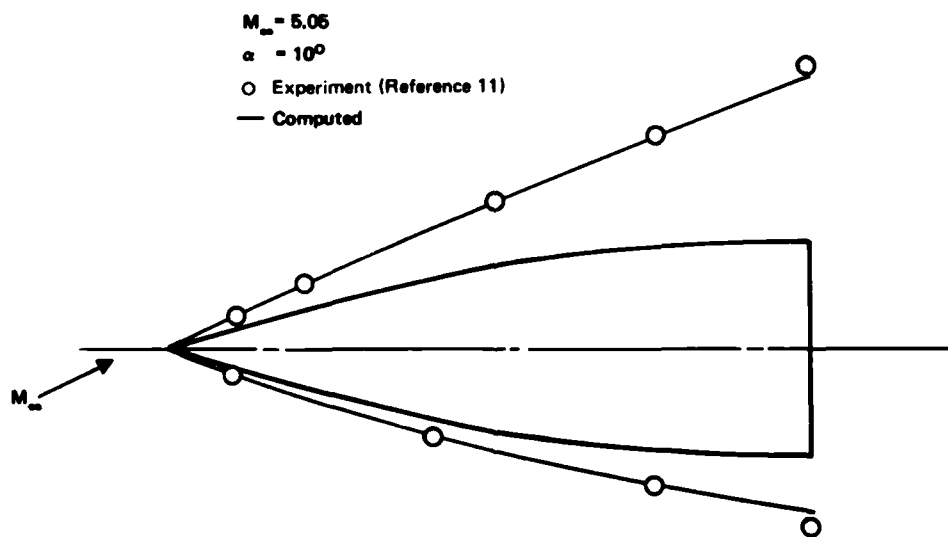


Figure 5-10. Shock-Wave Shape for Ogive

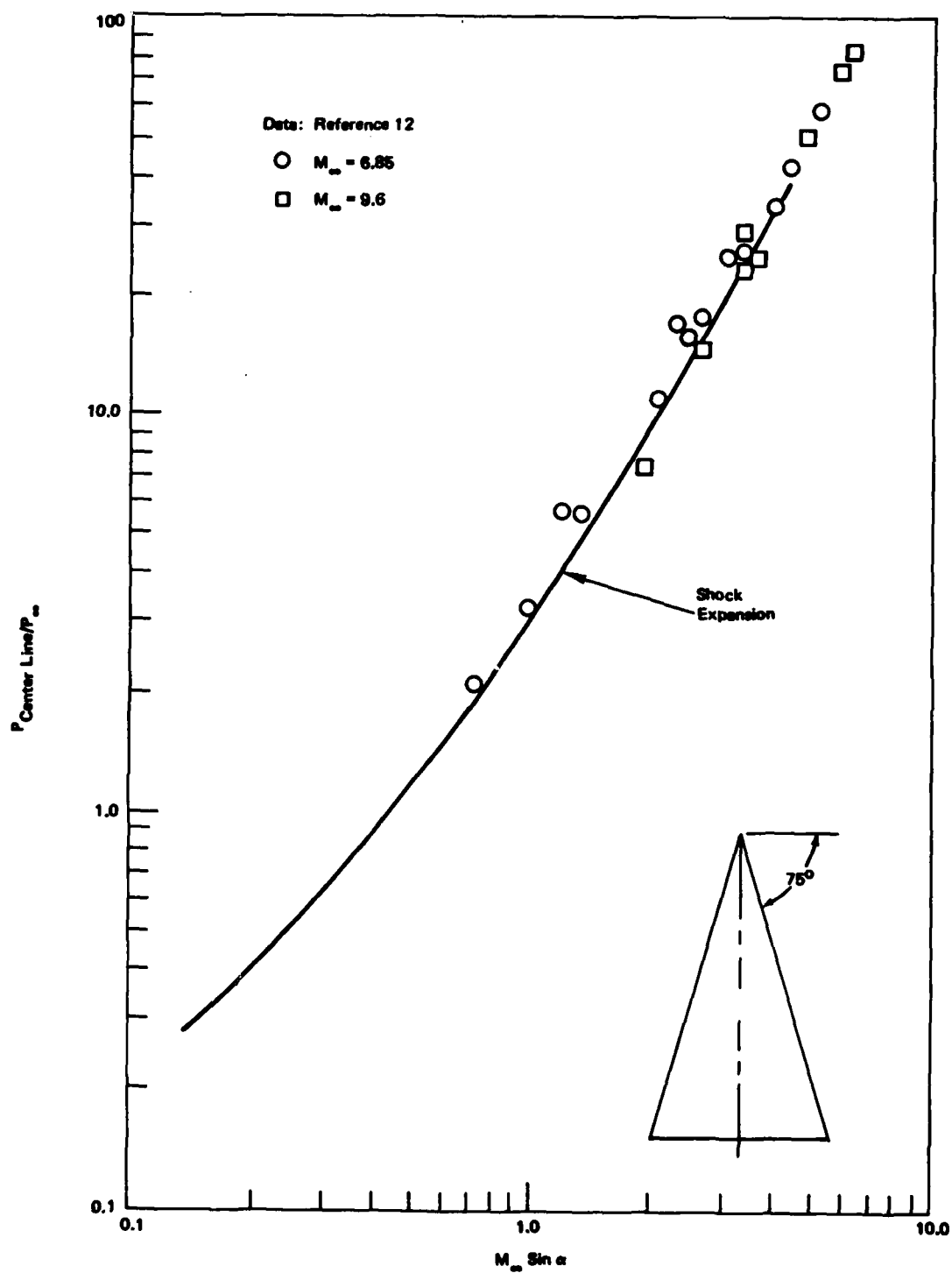


Figure 5-11. Delta Wing Center Line Pressure

5.3 TEST-CASE INPUT CARDS

CUA92	403	119	131	241	231	0.0000
CUA92	404	131	143	251	241	0.0000
CUA92	405	143	155	241	251	0.0000
CTIA2	406	211	221	222	0.0000	0.0000
CUA92	407	221	231	232	222	0.0000
CUA92	408	231	241	242	232	0.0000
CUA92	409	241	251	252	242	0.0000
CUA92	410	251	261	262	252	0.0000

5.4 TEST-CASE OUTPUT

5.4.1 Combined Angle of Attack and Yaw

EXPERIMENT CASE FOR CHECKING GENERALIZED MISSILE ATTITUDE MODE

McDONNELL DOUGLAS



MCDONNELL DOUGLAS

NYTRANS = 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

NYTRANS = 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

1891 - 0.

1 - 617651

74901 - 15

RD	=	0.0,	0.681E-01,	0.1743E+00,	0.2664E+00,	0.3326E+00,	0.4403E+00,	0.5289E+00,	0.5818E+00,	0.6E+00,
		0.6E+00,	0.6E+00,	0.6E+00,	0.6E+00,	0.6E+00,	0.0,	0.0,	0.0,	0.0,

0.0

YAW = 0.5101.

[illegible]

pivot = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

AND

[illegible]

INVEST - 0.00



ZN = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 MUPARL = 1,
 XLC = 0.6503E+01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 XTC = 0.10718E+02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 XNACH = 0.2E+01,
 PIMF = 0.147E+02,
 DADES = 0.0,
 ARB = 0.5E+01,
 ARV = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 ARN = 0.0, 0.0,
 SEND

BODY PANEL CORNER POINT LOCATIONS EXPRESSED IN THE AERO COORDINATE SYSTEM
1 AND 2 INDICATE BODY PANEL LEADING-EDGE POINTS, 3 AND 4 INDICATE TRAILING-EDGE POINTS

PANEL NO	PARTS	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
1	1	5.660	0.000	.600	5.660	.300	.520	6.503	0.000	.600	6.503	.300	.520
2	1	6.503	0.000	.600	6.503	.300	.520	6.925	0.000	.600	6.925	.300	.520
3	1	7.466	0.000	.600	7.466	.300	.520	8.008	0.000	.600	8.008	.300	.520
4	1	8.008	0.000	.600	8.008	.300	.520	8.550	0.000	.600	8.550	.300	.520
5	1	8.550	0.000	.600	8.550	.300	.520	9.092	0.000	.600	9.092	.300	.520
6	1	9.092	0.000	.600	9.092	.300	.520	9.634	0.000	.600	9.634	.300	.520
7	1	9.634	0.000	.600	9.634	.300	.520	10.176	0.000	.600	10.176	.300	.520
8	1	10.176	0.000	.600	10.176	.300	.520	10.718	0.000	.600	10.718	.300	.520
9	1	10.718	0.000	.600	10.718	.300	.520	11.260	0.000	.600	11.260	.300	.520
10	1	11.260	0.000	.600	11.260	.300	.520	6.503	0.000	.600	6.503	.300	.520
11	1	5.660	.300	.520	5.660	.320	.300	6.503	.300	.520	6.503	.320	.300
12	1	6.503	.300	.520	6.503	.320	.300	6.925	.300	.520	6.925	.320	.300
13	1	7.466	.300	.520	7.466	.320	.300	8.008	.300	.520	8.008	.320	.300
14	1	8.008	.300	.520	8.008	.320	.300	8.550	.300	.520	8.550	.320	.300
15	1	8.550	.300	.520	8.550	.320	.300	9.092	.300	.520	9.092	.320	.300
16	1	9.092	.300	.520	9.092	.320	.300	9.634	.300	.520	9.634	.320	.300
17	1	9.634	.300	.520	9.634	.320	.300	10.176	.300	.520	10.176	.320	.300
18	1	10.176	.300	.520	10.176	.320	.300	10.718	.300	.520	10.718	.320	.300
19	1	10.718	.300	.520	10.718	.320	.300	11.260	.300	.520	11.260	.320	.300
20	1	11.260	.300	.520	11.260	.320	.300	6.503	.300	.520	6.503	.320	.300
21	1	5.660	.320	.300	5.660	.300	.520	6.925	.320	.300	6.925	.300	.520
22	1	6.503	.320	.300	6.503	.300	.520	7.466	.320	.300	7.466	.300	.520
23	1	7.466	.320	.300	7.466	.300	.520	8.008	.320	.300	8.008	.300	.520
24	1	8.008	.320	.300	8.008	.300	.520	8.550	.320	.300	8.550	.300	.520
25	1	8.550	.320	.300	8.550	.300	.520	9.092	.320	.300	9.092	.300	.520
26	1	9.092	.320	.300	9.092	.300	.520	9.634	.320	.300	9.634	.300	.520
27	1	9.634	.320	.300	9.634	.300	.520	10.176	.320	.300	10.176	.300	.520
28	1	10.176	.320	.300	10.176	.300	.520	10.718	.320	.300	10.718	.300	.520
29	1	10.718	.320	.300	10.718	.300	.520	11.260	.320	.300	11.260	.300	.520
30	1	11.260	.320	.300	11.260	.300	.520	6.503	.320	.300	6.503	.300	.520
31	1	5.660	.300	.520	5.660	.320	.300	6.925	.300	.520	6.925	.320	.300
32	1	6.503	.300	.520	6.503	.320	.300	7.466	.300	.520	7.466	.320	.300
33	1	7.466	.300	.520	7.466	.320	.300	8.008	.300	.520	8.008	.320	.300
34	1	8.008	.300	.520	8.008	.320	.300	8.550	.300	.520	8.550	.320	.300
35	1	8.550	.300	.520	8.550	.320	.300	9.092	.300	.520	9.092	.320	.300
36	1	9.092	.300	.520	9.092	.320	.300	9.634	.300	.520	9.634	.320	.300
37	1	9.634	.300	.520	9.634	.320	.300	10.176	.300	.520	10.176	.320	.300
38	1	10.176	.300	.520	10.176	.320	.300	10.718	.300	.520	10.718	.320	.300
39	1	10.718	.300	.520	10.718	.320	.300	11.260	.300	.520	11.260	.320	.300
40	1	11.260	.300	.520	11.260	.320	.300	6.503	.300	.520	6.503	.320	.300
41	1	5.660	.320	.300	5.660	.300	.520	6.925	.320	.300	6.925	.300	.520
42	1	6.503	.320	.300	6.503	.300	.520	7.466	.320	.300	7.466	.300	.520
43	1	7.466	.320	.300	7.466	.320	.300	8.008	.320	.300	8.008	.320	.300
44	1	8.008	.320	.300	8.008	.320	.300	8.550	.320	.300	8.550	.320	.300
45	1	8.550	.320	.300	8.550	.320	.300	9.092	.320	.300	9.092	.320	.300
46	1	9.092	.320	.300	9.092	.320	.300	9.634	.320	.300	9.634	.320	.300
47	1	9.634	.320	.300	9.634	.320	.300	10.176	.320	.300	10.176	.320	.300
48	1	10.176	.320	.300	10.176	.320	.300	10.718	.320	.300	10.718	.320	.300
49	1	10.718	.320	.300	10.718	.320	.300	11.260	.320	.300	11.260	.320	.300
50	1	11.260	.320	.300	11.260	.320	.300	6.503	.320	.300	6.503	.320	.300
51	1	5.660	.300	.520	5.660	.320	.300	6.925	.300	.520	6.925	.320	.300
52	1	6.503	.300	.520	6.503	.320	.300	7.466	.300	.520	7.466	.320	.300
53	1	7.466	.300	.520	7.466	.320	.300	8.008	.300	.520	8.008	.320	.300
54	1	8.008	.300	.520	8.008	.320	.300	8.550	.300	.520	8.550	.320	.300
55	1	8.550	.300	.520	8.550	.320	.300	9.092	.300	.520	9.092	.320	.300
56	1	9.092	.300	.520	9.092	.320	.300	9.634	.300	.520	9.634	.320	.300
57	1	9.634	.300	.520	9.634	.320	.300	10.176	.300	.520	10.176	.320	.300
58	1	10.176	.300	.520	10.176	.320	.300	10.718	.300	.520	10.718	.320	.300

59	1	10.176	.300	-.520	10.176	-.000	-.600	10.718	-.000	-.600
60	1	10.718	.500	-.520	10.718	-.000	-.600	11.260	-.000	-.600

BODY PANEL CENTROID AND CONTROL POINT LOCATIONS EXPRESSED IN THE AERO COORDINATE SYSTEM

PANEL	X _C	Y _C	Z _C	X _{CP}	Y _{CP}	Z _{CP}	AREA	THETA- INCLIN	ALPHA- INCLIN	CHORD
1	6.082	.150	.560	6.461	.150	.560	.262	-.262	0.000	.843
2	6.714	.150	.560	6.903	.150	.560	.131	-.262	0.000	.422
3	7.195	.150	.560	7.439	.150	.560	.168	-.262	0.000	.542
4	7.737	.150	.560	7.981	.150	.560	.168	-.262	0.000	.542
5	8.279	.150	.560	8.523	.150	.560	.168	-.262	0.000	.542
6	8.821	.150	.560	9.065	.150	.560	.168	-.262	0.000	.542
7	9.363	.150	.560	9.607	.150	.560	.168	-.262	0.000	.542
8	9.905	.150	.560	10.149	.150	.560	.168	-.262	0.000	.542
9	10.447	.150	.560	10.691	.150	.560	.168	-.262	0.000	.542
10	10.989	.150	.560	11.233	.150	.560	.168	-.262	0.000	.542
11	6.082	.410	.410	6.461	.410	.410	.262	-.785	0.000	.843
12	6.714	.410	.410	6.903	.410	.410	.131	-.785	0.000	.422
13	7.195	.410	.410	7.439	.410	.410	.168	-.785	0.000	.542
14	7.737	.410	.410	7.981	.410	.410	.168	-.785	0.000	.542
15	8.279	.410	.410	8.523	.410	.410	.168	-.785	0.000	.542
16	8.821	.410	.410	9.065	.410	.410	.168	-.785	0.000	.542
17	9.363	.410	.410	9.607	.410	.410	.168	-.785	0.000	.542
18	9.905	.410	.410	10.149	.410	.410	.168	-.785	0.000	.542
19	10.447	.410	.410	10.691	.410	.410	.168	-.785	0.000	.542
20	10.989	.410	.410	11.233	.410	.410	.168	-.785	0.000	.542
21	6.082	.560	.150	6.461	.560	.150	.262	-1.309	0.000	.843
22	6.714	.560	.150	6.903	.560	.150	.131	-1.309	0.000	.422
23	7.195	.560	.150	7.439	.560	.150	.168	-1.309	0.000	.542
24	7.737	.560	.150	7.981	.560	.150	.168	-1.309	0.000	.542
25	8.279	.560	.150	8.523	.560	.150	.168	-1.309	0.000	.542
26	8.821	.560	.150	9.065	.560	.150	.168	-1.309	0.000	.542
27	9.363	.560	.150	9.607	.560	.150	.168	-1.309	0.000	.542
28	9.905	.560	.150	10.149	.560	.150	.168	-1.309	0.000	.542
29	10.447	.560	.150	10.691	.560	.150	.168	-1.309	0.000	.542
30	10.989	.560	.150	11.233	.560	.150	.168	-1.309	0.000	.542
31	6.082	.560	.150	6.461	.560	.150	.262	-1.833	0.000	.843
32	6.714	.560	.150	6.903	.560	.150	.131	-1.833	0.000	.422
33	7.195	.560	.150	7.439	.560	.150	.168	-1.833	0.000	.542
34	7.737	.560	.150	7.981	.560	.150	.168	-1.833	0.000	.542
35	8.279	.560	.150	8.523	.560	.150	.168	-1.833	0.000	.542
36	8.821	.560	.150	9.065	.560	.150	.168	-1.833	0.000	.542
37	9.363	.560	.150	9.607	.560	.150	.168	-1.833	0.000	.542
38	9.905	.560	.150	10.149	.560	.150	.168	-1.833	0.000	.542
39	10.447	.560	.150	10.691	.560	.150	.168	-1.833	0.000	.542
40	10.989	.560	.150	11.233	.560	.150	.168	-1.833	0.000	.542
41	6.082	.410	.410	6.461	.410	.410	.262	-2.336	0.000	.843
42	6.714	.410	.410	6.903	.410	.410	.131	-2.336	0.000	.422
43	7.195	.410	.410	7.439	.410	.410	.168	-2.336	0.000	.542
44	7.737	.410	.410	7.981	.410	.410	.168	-2.336	0.000	.542
45	8.279	.410	.410	8.523	.410	.410	.168	-2.336	0.000	.542
46	8.821	.410	.410	9.065	.410	.410	.168	-2.336	0.000	.542
47	9.363	.410	.410	9.607	.410	.410	.168	-2.336	0.000	.542
48	9.905	.410	.410	10.149	.410	.410	.168	-2.336	0.000	.542
49	10.447	.410	.410	10.691	.410	.410	.168	-2.336	0.000	.542
50	10.989	.410	.410	11.233	.410	.410	.168	-2.336	0.000	.542
51	6.082	.150	.560	6.461	.150	.560	.262	-2.800	0.000	.843
52	6.714	.150	.560	6.903	.150	.560	.131	-2.800	0.000	.422
53	7.195	.150	.560	7.439	.150	.560	.168	-2.800	0.000	.542
54	7.737	.150	.560	7.981	.150	.560	.168	-2.800	0.000	.542
55	8.279	.150	.560	8.523	.150	.560	.168	-2.800	0.000	.542
56	8.821	.150	.560	9.065	.150	.560	.168	-2.800	0.000	.542
57	9.363	.150	.560	9.607	.150	.560	.168	-2.800	0.000	.542
58	9.905	.150	.560	10.149	.150	.560	.168	-2.800	0.000	.542
59	10.447	.150	.560	10.691	.150	.560	.168	-2.800	0.000	.542

40 10.989 .150 -.560 11.233 .150 -.560 .168 -2.880 0.000 .542

WING PANEL CORNER POINT LOCATIONS EXPRESSED IN THE AERO COORDINATE SYSTEM
1 AND 2 INDICATE WING PANEL LEADING-EDGE POINTS, 3 AND 4 INDICATE TRAILING-EDGE POINTS

PANEL NO	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
WING NO.	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
1	6.503	.600	0.000	7.346	1.307	0.000	6.925	.600	0.000	7.483	1.307	0.000	7.483	1.307	0.000
2	4.925	.600	0.000	7.683	1.307	0.000	7.466	.600	0.000	8.117	1.307	0.000	8.117	1.307	0.000
3	8.008	.600	0.000	8.117	1.307	0.000	8.008	.600	0.000	8.550	1.307	0.000	8.550	1.307	0.000
4	8.550	.600	0.000	8.550	1.307	0.000	8.550	.600	0.000	8.984	1.307	0.000	8.984	1.307	0.000
5	9.092	.600	0.000	9.092	1.307	0.000	9.092	.600	0.000	9.526	1.307	0.000	9.526	1.307	0.000
6	9.526	.600	0.000	9.526	1.307	0.000	9.526	.600	0.000	9.969	1.307	0.000	9.969	1.307	0.000
7	10.176	.600	0.000	10.176	1.307	0.000	10.176	.600	0.000	10.618	1.307	0.000	10.618	1.307	0.000
8	10.618	.600	0.000	10.618	1.307	0.000	10.618	.600	0.000	11.061	1.307	0.000	11.061	1.307	0.000
9	11.061	.600	0.000	11.061	1.307	0.000	11.061	.600	0.000	11.504	1.307	0.000	11.504	1.307	0.000
10	11.504	.600	0.000	11.504	1.307	0.000	11.504	.600	0.000	11.947	1.307	0.000	11.947	1.307	0.000
11	11.947	.600	0.000	11.947	1.307	0.000	11.947	.600	0.000	12.390	1.307	0.000	12.390	1.307	0.000
12	12.390	.600	0.000	12.390	1.307	0.000	12.390	.600	0.000	12.833	1.307	0.000	12.833	1.307	0.000
13	12.833	.600	0.000	12.833	1.307	0.000	12.833	.600	0.000	13.276	1.307	0.000	13.276	1.307	0.000
14	13.276	.600	0.000	13.276	1.307	0.000	13.276	.600	0.000	13.719	1.307	0.000	13.719	1.307	0.000
15	13.719	.600	0.000	13.719	1.307	0.000	13.719	.600	0.000	14.162	1.307	0.000	14.162	1.307	0.000
16	14.162	.600	0.000	14.162	1.307	0.000	14.162	.600	0.000	14.605	1.307	0.000	14.605	1.307	0.000
17	14.605	.600	0.000	14.605	1.307	0.000	14.605	.600	0.000	15.048	1.307	0.000	15.048	1.307	0.000
18	15.048	.600	0.000	15.048	1.307	0.000	15.048	.600	0.000	15.491	1.307	0.000	15.491	1.307	0.000
19	15.491	.600	0.000	15.491	1.307	0.000	15.491	.600	0.000	15.934	1.307	0.000	15.934	1.307	0.000
20	15.934	.600	0.000	15.934	1.307	0.000	15.934	.600	0.000	16.377	1.307	0.000	16.377	1.307	0.000
21	16.377	.600	0.000	16.377	1.307	0.000	16.377	.600	0.000	16.820	1.307	0.000	16.820	1.307	0.000
22	16.820	.600	0.000	16.820	1.307	0.000	16.820	.600	0.000	17.263	1.307	0.000	17.263	1.307	0.000
23	17.263	.600	0.000	17.263	1.307	0.000	17.263	.600	0.000	17.706	1.307	0.000	17.706	1.307	0.000
24	17.706	.600	0.000	17.706	1.307	0.000	17.706	.600	0.000	18.149	1.307	0.000	18.149	1.307	0.000
25	18.149	.600	0.000	18.149	1.307	0.000	18.149	.600	0.000	18.592	1.307	0.000	18.592	1.307	0.000
26	18.592	.600	0.000	18.592	1.307	0.000	18.592	.600	0.000	19.035	1.307	0.000	19.035	1.307	0.000
27	19.035	.600	0.000	19.035	1.307	0.000	19.035	.600	0.000	19.478	1.307	0.000	19.478	1.307	0.000
28	19.478	.600	0.000	19.478	1.307	0.000	19.478	.600	0.000	19.921	1.307	0.000	19.921	1.307	0.000
29	19.921	.600	0.000	19.921	1.307	0.000	19.921	.600	0.000	20.364	1.307	0.000	20.364	1.307	0.000
30	20.364	.600	0.000	20.364	1.307	0.000	20.364	.600	0.000	20.807	1.307	0.000	20.807	1.307	0.000
31	20.807	.600	0.000	20.807	1.307	0.000	20.807	.600	0.000	21.250	1.307	0.000	21.250	1.307	0.000
32	21.250	.600	0.000	21.250	1.307	0.000	21.250	.600	0.000	21.693	1.307	0.000	21.693	1.307	0.000
33	21.693	.600	0.000	21.693	1.307	0.000	21.693	.600	0.000	22.136	1.307	0.000	22.136	1.307	0.000
34	22.136	.600	0.000	22.136	1.307	0.000	22.136	.600	0.000	22.579	1.307	0.000	22.579	1.307	0.000
35	22.579	.600	0.000	22.579	1.307	0.000	22.579	.600	0.000	23.022	1.307	0.000	23.022	1.307	0.000
36	23.022	.600	0.000	23.022	1.307	0.000	23.022	.600	0.000	23.465	1.307	0.000	23.465	1.307	0.000
37	23.465	.600	0.000	23.465	1.307	0.000	23.465	.600	0.000	23.908	1.307	0.000	23.908	1.307	0.000
38	23.908	.600	0.000	23.908	1.307	0.000	23.908	.600	0.000	24.351	1.307	0.000	24.351	1.307	0.000
39	24.351	.600	0.000	24.351	1.307	0.000	24.351	.600	0.000	24.794	1.307	0.000	24.794	1.307	0.000
40	24.794	.600	0.000	24.794	1.307	0.000	24.794	.600	0.000	25.237	1.307	0.000	25.237	1.307	0.000

WING PANEL CENTROID AND CONTROL POINT LOCATIONS EXPRESSED IN THE AERO COORDINATE SYSTEM

PANEL	X _C	Y _C	Z _C	X _{CP}	Y _{CP}	Z _{CP}	AREA	THETA- DINED	ALPHA- CAMBER	CHORD
1	7.099	.941	0.000	7.271	.941	0.000	.268	0.000	0.000	.381
2	7.535	.941	0.000	7.735	.941	0.000	.345	0.000	0.000	.490
3	8.024	.941	0.000	8.265	.941	0.000	.345	0.000	0.000	.490
4	8.514	.941	0.000	8.735	.941	0.000	.345	0.000	0.000	.490
5	9.004	.941	0.000	9.224	.941	0.000	.345	0.000	0.000	.490
6	9.494	.941	0.000	9.714	.941	0.000	.345	0.000	0.000	.490
7	9.983	.941	0.000	10.204	.941	0.000	.345	0.000	0.000	.490
8	10.473	.941	0.000	10.694	.941	0.000	.345	0.000	0.000	.490
9	10.963	1.644	0.000	11.444	1.644	0.000	.268	0.000	0.000	.297
10	8.235	1.644	0.000	8.407	1.644	0.000	.268	0.000	0.000	.382
11	8.617	1.644	0.000	8.789	1.644	0.000	.268	0.000	0.000	.382
12	8.999	1.644	0.000	9.171	1.644	0.000	.268	0.000	0.000	.382
13	9.381	1.644	0.000	9.553	1.644	0.000	.268	0.000	0.000	.382
14	9.763	1.644	0.000	9.935	1.644	0.000	.268	0.000	0.000	.382
15	10.145	1.644	0.000	10.317	1.644	0.000	.268	0.000	0.000	.382
16	10.527	1.644	0.000	10.699	1.644	0.000	.268	0.000	0.000	.382
17	8.689	2.345	0.000	8.785	2.345	0.000	.192	0.000	0.000	.274
18	8.933	2.345	0.000	9.037	2.345	0.000	.192	0.000	0.000	.275
19	9.208	2.345	0.000	9.331	2.345	0.000	.192	0.000	0.000	.275
20	9.482	2.345	0.000	9.604	2.345	0.000	.192	0.000	0.000	.275
21	9.757	2.345	0.000	9.881	2.345	0.000	.192	0.000	0.000	.275
22	10.032	2.345	0.000	10.135	2.345	0.000	.192	0.000	0.000	.275
23	10.306	2.345	0.000	10.430	2.345	0.000	.192	0.000	0.000	.275
24	10.581	2.345	0.000	10.704	2.345	0.000	.192	0.000	0.000	.275
25	9.472	3.037	0.000	9.531	3.037	0.000	.089	0.000	0.000	.131
26	9.622	3.037	0.000	9.698	3.037	0.000	.115	0.000	0.000	.169
27	9.791	3.037	0.000	9.867	3.037	0.000	.115	0.000	0.000	.169
28	9.959	3.037	0.000	10.035	3.037	0.000	.115	0.000	0.000	.169
29	10.128	3.037	0.000	10.204	3.037	0.000	.115	0.000	0.000	.169
30	10.297	3.037	0.000	10.372	3.037	0.000	.115	0.000	0.000	.169
31	10.465	3.037	0.000	10.541	3.037	0.000	.115	0.000	0.000	.169
32	10.634	3.037	0.000	10.710	3.037	0.000	.115	0.000	0.000	.169
33	10.184	3.665	0.000	10.209	3.665	0.000	.030	0.000	0.000	.056
34	10.248	3.665	0.000	10.281	3.665	0.000	.038	0.000	0.000	.072
35	10.321	3.665	0.000	10.353	3.665	0.000	.038	0.000	0.000	.072
36	10.393	3.665	0.000	10.425	3.665	0.000	.038	0.000	0.000	.072
37	10.465	3.665	0.000	10.498	3.665	0.000	.038	0.000	0.000	.072
38	10.537	3.665	0.000	10.570	3.665	0.000	.038	0.000	0.000	.072
39	10.610	3.665	0.000	10.642	3.665	0.000	.038	0.000	0.000	.072
40	10.682	3.665	0.000	10.714	3.665	0.000	.038	0.000	0.000	.072

DESCRIPTION OF CASE REQUESTED

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

CASE = 2. CALCULATE CL, GIVEN SHAPE

CPCALC = 1. NON-LINEAR CP

POLAR = 1. POLARS REQUESTED

THICK = 1. WING THICKNESS PRESSURES TO BE ADDED

VOUT = 0. VELOCITY COMPONENTS NOT TO BE PRINTED

MACH NUMBER = 2.0000

POINT ABOUT WHICH THE MOMENTS ARE TO BE COMPUTED

X-COORDINATE = 0.0000
Z-COORDINATE = 0.0000

REFERENCE CHORD LENGTH = 1.0000

WING REFERENCE AREA = 7.4342

WING SEMI-SPAN = 1.0000

HEIGHT OF WING PLANE ABOVE BODY AXIS = 0.0000

INCLINATION OF BODY AXIS WITH RESPECT TO DEFINING AXIS = 0.0000 DEG.

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

BODY YAW = 5.00 DEG

BODY ROLL = 0.00 DEG

PRESSURES, FORCES, AND MOMENTS ON ISOLATED BODY

CP =	.0194	CL =	.0311	CB =	-.0313	CPM =	-.0398	CRM =	0.0000	CYM =	.0386	
BODY PRESSURE COEFFICIENTS(CP)												
THETA(DEG.)	0.00	30.00	60.00	90.00	120.00	150.00	180.00	210.00	240.00	270.00	300.00	330.00
0.0000	.0303	.0460	.0917	.1619	.2188	.2190	.1625	.0922	.0462	.0303	.0298	.0298
.3000	.0299	.0456	.0914	.1613	.2183	.2185	.1620	.0918	.0458	.0299	.0294	.0294
.6000	.0300	.0454	.0909	.1608	.2176	.2178	.1613	.0913	.0456	.0301	.0297	.0297
.9000	.0325	.0482	.0943	.1649	.2221	.2224	.1654	.0947	.0485	.0325	.0320	.0320
1.2000	.0323	.0486	.0951	.1661	.2235	.2237	.1666	.0956	.0488	.0324	.0315	.0315
1.5000	.0283	.0445	.0906	.1608	.2175	.2178	.1613	.0910	.0447	.0283	.0273	.0274
1.8000	.0248	.0403	.0854	.1546	.2106	.2108	.1551	.0858	.0405	.0248	.0242	.0243
2.1000	.0281	.0431	.0879	.1572	.2135	.2138	.1577	.0884	.0433	.0282	.0281	.0281
2.4000	.0533	.0698	.1186	.1935	.2541	.2544	.1941	.1191	.0701	.0534	.0530	.0530
2.7000	.0124	.0288	.0735	.1410	.1956	.1958	.1415	.0738	.0289	.0125	.0110	.0110
3.0000	-.1156	-.1009	-.0671	-.0186	.0204	.0205	-.0184	-.0671	-.1010	-.1158	-.1188	-.1187
3.8571	-.0687	-.0674	-.0425	.0030	.0419	.0420	.0032	-.0424	-.0675	-.0689	-.0613	-.0612
4.7143	-.0518	-.0692	-.0597	-.0242	.0100	.0101	-.0240	-.0596	-.0693	-.0520	-.0302	-.0301
5.5714	-.0342	-.0608	-.0591	-.0292	.0018	.0019	-.0290	-.0590	-.0609	-.0344	-.0056	-.0055
6.4286	-.0269	-.0570	-.0589	-.0324	-.0036	-.0035	-.0321	-.0588	-.0571	-.0271	.0039	.0041
7.2857	-.0215	-.0519	-.0548	-.0297	-.0023	-.0021	-.0295	-.0547	-.0520	-.0217	.0094	.0095
8.1429	-.0188	-.0483	-.0510	-.0264	.0005	.0006	-.0262	-.0509	-.0484	-.0191	.0111	.0112
9.0000	-.0171	-.0453	-.0472	-.0225	.0043	.0044	-.0223	-.0471	-.0454	-.0173	.0118	.0119
9.8571	-.0168	-.0438	-.0449	-.0199	.0070	.0071	-.0196	-.0448	-.0439	-.0170	.0111	.0112
10.7143	-.0182	-.0443	-.0448	-.0195	.0073	.0074	-.0193	-.0447	-.0444	-.0184	.0089	.0090
11.5714	-.0210	-.0468	-.0471	-.0219	.0048	.0049	-.0217	-.0470	-.0469	-.0212	.0059	.0060
12.4286	-.0246	-.0508	-.0515	-.0269	-.0007	-.0006	-.0267	-.0515	-.0509	-.0248	.0026	.0027
13.2857	-.0287	-.0560	-.0579	-.0346	-.0089	-.0088	-.0342	-.0578	-.0561	-.0289	-.0006	-.0004
14.1429	-.0353	-.0645	-.0681	-.0463	-.0222	-.0221	-.0462	-.0681	-.0646	-.0356	-.0058	-.0057
15.0000	.0087	-.0267	-.0344	-.0134	.0115	.0116	-.0132	-.0343	-.0268	.0085	.0431	.0432

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

CD = .0006 CL = .0265 CB = .0197 CPM = .1628 CRM = 0.0000 CYM = .2180

BODY PANEL PRESSURE COEFFICIENT(CP)

THETA(DEG.)	15.00	45.00	75.00	105.00	135.00	165.00	195.00	225.00	255.00	285.00	315.00	345.00
ROW NO.												
1	-.0000	-.0000	.0000	-.0000	.0000	-.0000	-.0000	.0000	-.0000	.0000	-.0000	-.0000
2	.0001	.0013	-.0521	.0747	.0001	.0000	.0000	.0001	.0747	-.0521	.0013	.0001
3	.0021	-.0209	-.0991	.1335	.0386	.0007	.0007	.0386	.1335	-.0991	-.0209	.0021
4	-.0064	-.0566	-.0748	.0831	.0853	.0204	.0204	.0853	.0831	-.0742	-.0566	-.0064
5	-.0390	-.0486	-.0695	.0573	.0773	.0548	.0548	.0773	.0573	-.0695	-.0486	-.0390
6	-.0504	-.0424	-.0675	.0352	.0706	.0735	.0735	.0706	.0352	-.0675	-.0424	-.0504
7	-.0497	-.0411	-.0731	.0194	.0898	.0760	.0760	.0898	.0194	-.0731	-.0411	-.0497
8	-.0454	-.0410	-.0840	.0033	.0702	.0738	.0738	.0702	.0033	-.0840	-.0410	-.0454
9	-.0427	-.0399	-.0977	-.0110	.0894	.0736	.0736	.0894	-.0110	-.0977	-.0399	-.0427
10	-.0408	-.0370	-.0913	-.0635	.0652	.0731	.0731	.0652	-.0635	-.0913	-.0370	-.0408

BODY PANEL SLOPE(PR/DR)

THETA(DEG.)	15.00	45.00	75.00	105.00	135.00	165.00	195.00	225.00	255.00	285.00	315.00	345.00
ROW NO.												
1	-.0618	-.0003	.0613	.1066	.1233	.1068	.0616	.0001	-.0615	-.1068	-.1235	-.1070
2	-.0621	-.0006	.0610	.1063	.1230	.1065	.0614	.0001	-.0617	-.1070	-.1237	-.1072
3	-.0622	-.0007	.0609	.1062	.1229	.1064	.0613	-.0002	-.0618	-.1071	-.1238	-.1073
4	-.0621	-.0006	.0610	.1063	.1230	.1065	.0614	-.0002	-.0617	-.1070	-.1237	-.1072
5	-.0618	-.0003	.0613	.1066	.1233	.1068	.0617	.0001	-.0615	-.1067	-.1234	-.1070
6	-.0614	.0002	.0618	.1070	.1237	.1072	.0621	.0006	-.0610	-.1063	-.1230	-.1065
7	-.0608	.0008	.0626	.1076	.1243	.1078	.0627	.0012	-.0604	-.1037	-.1224	-.1059
8	-.0606	.0010	.0626	.1078	.1245	.1081	.0629	.0014	-.0602	-.1035	-.1222	-.1057
9	-.0610	.0005	.0621	.1074	.1241	.1076	.0625	.0010	-.0606	-.1039	-.1226	-.1061
10	-.0621	-.0006	.0610	.1063	.1230	.1065	.0613	-.0002	-.0618	-.1071	-.1238	-.1073

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

CD = .0479 CL = .2515 CG = .0023 CPM = -2.3062 CPM = .4876 CPM = -.0706

SECTION CD DISTRIBUTION

SPANWISE STATION 1 .04004 2 .04598 3 .03590 4 .06010 5 .05655

SECTION CL DISTRIBUTION

SPANWISE STATION 1 .20650 2 .24106 3 .29594 4 .31941 5 .30264

SECTION CD DISTRIBUTION

SPANWISE STATION 1 .00191 2 .00216 3 .00261 4 .00279 5 .00261

UPPER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION	1	2	3	4	5
CHORDWISE STATION 1	-.16063	-.13427	-.11961	-.11893	-.11694
2	-.11364	-.13776	-.12237	-.11714	-.11716
3	-.07038	-.11434	-.12689	-.11376	-.11729
4	-.04761	-.08883	-.12671	-.11619	-.11727
5	-.03603	-.06433	-.11590	-.11779	-.11710
6	-.03249	-.05036	-.10098	-.11948	-.11629
7	-.03245	-.04132	-.08329	-.12143	-.11637
8	-.03477	-.03960	-.06671	-.12199	-.11588

LOWER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

SPANWISE STATION	1	2	3	4	5
CHORDWISE STATION 1	-.27185	-.23140	-.21016	-.19453	-.18930
2	-.20897	-.23461	-.21571	-.19830	-.18910
3	-.15329	-.20306	-.22058	-.20354	-.18933
4	-.12559	-.17036	-.21891	-.20480	-.18999
5	-.11323	-.13837	-.20232	-.20988	-.19094
6	-.11096	-.12083	-.18607	-.21216	-.19218
7	-.11044	-.11093	-.16210	-.21332	-.19367
8	-.11115	-.11066	-.13960	-.21221	-.19544

WING PANEL PRESSURE DIFFERENCE(CD)

SPANWISE STATION	1	2	3	4	5
CHORDWISE STATION 1	-.41652	-.35227	-.31724	-.30407	-.29760
2	-.30228	-.35879	-.32503	-.30500	-.29735
3	-.20839	-.30408	-.33450	-.30728	-.29789
4	-.16033	-.24714	-.33269	-.31038	-.29840
5	-.13721	-.19364	-.30627	-.31677	-.29893
6	-.13106	-.16311	-.27440	-.31891	-.29936
7	-.12963	-.14514	-.23457	-.32201	-.30023
8	-.13159	-.14278	-.19736	-.32184	-.30104

UPPER SURFACE WING PANEL SLOPE(DZ/DX)					
SPANWISE STATION	1	2	3	4	5
CHORDWISE STATION					
1	-.04367	-.04367	-.04367	-.04367	-.04367
2	-.04367	-.04367	-.04367	-.04367	-.04367
3	-.04367	-.04367	-.04367	-.04367	-.04367
4	-.04367	-.04367	-.04367	-.04367	-.04367
5	-.04367	-.04367	-.04367	-.04367	-.04367
6	-.04367	-.04367	-.04367	-.04367	-.04367
7	-.04367	-.04367	-.04367	-.04367	-.04367
8	-.04367	-.04367	-.04367	-.04367	-.04367
LOWER SURFACE WING PANEL SLOPE(DZ/DX)					
SPANWISE STATION	1	2	3	4	5
CHORDWISE STATION					
1	-.13087	-.13087	-.13087	-.13087	-.13087
2	-.13087	-.13087	-.13087	-.13087	-.13087
3	-.13087	-.13087	-.13087	-.13087	-.13087
4	-.13087	-.13087	-.13087	-.13087	-.13087
5	-.13087	-.13087	-.13087	-.13087	-.13087
6	-.13087	-.13087	-.13087	-.13087	-.13087
7	-.13087	-.13087	-.13087	-.13087	-.13087
8	-.13087	-.13087	-.13087	-.13087	-.13087
WING CAMBER SLOPE(DZ/DX)					
SPANWISE STATION	1	2	3	4	5
CHORDWISE STATION					
1	-.08727	-.08727	-.08727	-.08727	-.08727
2	-.08727	-.08727	-.08727	-.08727	-.08727
3	-.08727	-.08727	-.08727	-.08727	-.08727
4	-.08727	-.08727	-.08727	-.08727	-.08727
5	-.08727	-.08727	-.08727	-.08727	-.08727
6	-.08727	-.08727	-.08727	-.08727	-.08727
7	-.08727	-.08727	-.08727	-.08727	-.08727
8	-.08727	-.08727	-.08727	-.08727	-.08727
WING CHORD LENGTHS(C)					
	3.80911	2.97057	2.13560	1.31133	.56200

MCDONNELL DOUGLAS

BODY PRESSURE COEFFICIENTS (CP), INCLUDING THE EFFECT OF WING

TMETA (DEC)	0.0000 300.0000	30.0000 330.0000	60.0000	90.0000	120.0000	150.0000	180.0000	210.0000	240.0000	270.0000
X										
0.0000	-0.3026 -0.2973 -0.2989	-0.4598 -0.2976 -0.4563	-0.9175 -0.9137 -0.9137	-1.6193 -1.6150 -1.6150	-2.1881 -2.1829 -2.1853	-2.1904 -2.1853 -2.1853	-1.6246 -1.6200 -1.6200	-0.9279 -0.9181 -0.9181	-0.6619 -0.6584 -0.6584	-0.3030 -0.2993 -0.2993
.3000	-0.2935 -0.3002 -0.2971	-0.2937 -0.4543 -0.2972	-0.9089 -0.9089 -0.9089	-1.6083 -1.6083 -1.6083	-2.1756 -2.1756 -2.1756	-2.1779 -2.1779 -2.1779	-1.6134 -1.6134 -1.6134	-0.9133 -0.9133 -0.9133	-0.6564 -0.6564 -0.6564	-0.3005 -0.3005 -0.3005
.6000	-0.2971 -0.3267 -0.3198	-0.4823 -0.4823 -0.4823	-0.9429 -0.9429 -0.9429	-1.6493 -1.6493 -1.6493	-2.2214 -2.2214 -2.2214	-2.2238 -2.2238 -2.2238	-1.6545 -1.6545 -1.6545	-0.9474 -0.9474 -0.9474	-0.6847 -0.6847 -0.6847	-0.3251 -0.3251 -0.3251
.9000	-0.3198 -0.3150 -0.3150	-0.4860 -0.4860 -0.4860	-0.9511 -0.9511 -0.9511	-1.6609 -1.6609 -1.6609	-2.2347 -2.2347 -2.2347	-2.2371 -2.2371 -2.2371	-1.6661 -1.6661 -1.6661	-0.9557 -0.9557 -0.9557	-0.6882 -0.6882 -0.6882	-0.3236 -0.3236 -0.3236
1.2000	-0.2825 -0.2734 -0.2734	-0.4447 -0.4447 -0.4447	-0.9053 -0.9053 -0.9053	-1.6078 -1.6078 -1.6078	-2.1755 -2.1755 -2.1755	-2.1778 -2.1778 -2.1778	-1.6126 -1.6126 -1.6126	-0.9099 -0.9099 -0.9099	-0.6468 -0.6468 -0.6468	-0.2828 -0.2828 -0.2828
1.5000	-0.2734 -0.2425 -0.2425	-0.4033 -0.4033 -0.4033	-0.8542 -0.8542 -0.8542	-1.5456 -1.5456 -1.5456	-2.1039 -2.1039 -2.1039	-2.1081 -2.1081 -2.1081	-1.5505 -1.5505 -1.5505	-0.8584 -0.8584 -0.8584	-0.6032 -0.6032 -0.6032	-0.2482 -0.2482 -0.2482
1.8000	-0.2614 -0.2809 -0.2809	-0.4308 -0.4308 -0.4308	-0.8792 -0.8792 -0.8792	-1.5722 -1.5722 -1.5722	-2.1352 -2.1352 -2.1352	-2.1375 -2.1375 -2.1375	-1.5772 -1.5772 -1.5772	-0.8836 -0.8836 -0.8836	-0.6439 -0.6439 -0.6439	-0.2617 -0.2617 -0.2617
2.1000	-0.3331 -0.3302 -0.3302	-0.6980 -0.6980 -0.6980	-1.1857 -1.1857 -1.1857	-1.9349 -1.9349 -1.9349	-2.5410 -2.5410 -2.5410	-2.5437 -2.5437 -2.5437	-1.9408 -1.9408 -1.9408	-1.1911 -1.1911 -1.1911	-0.7008 -0.7008 -0.7008	-0.3338 -0.3338 -0.3338
2.4000	-0.1245 -0.1100 -0.1100	-0.2875 -0.2875 -0.2875	-0.7346 -0.7346 -0.7346	-1.4103 -1.4103 -1.4103	-1.9559 -1.9559 -1.9559	-1.9580 -1.9580 -1.9580	-1.4150 -1.4150 -1.4150	-0.7385 -0.7385 -0.7385	-0.2892 -0.2892 -0.2892	-0.1246 -0.1246 -0.1246
2.7000	-0.1150 -0.1181 -0.1181	-0.1009 -0.1009 -0.1009	-0.6711 -0.6711 -0.6711	-0.8155 -0.8155 -0.8155	-0.2037 -0.2037 -0.2037	-0.2045 -0.2045 -0.2045	-0.1840 -0.1840 -0.1840	-0.6706 -0.6706 -0.6706	-0.1009 -0.1009 -0.1009	-0.1157 -0.1157 -0.1157
3.0000	-0.0874 -0.0612 -0.0612	-0.6737 -0.6737 -0.6737	-0.4246 -0.4246 -0.4246	-0.0299 -0.0299 -0.0299	-0.4191 -0.4191 -0.4191	-0.4201 -0.4201 -0.4201	-0.0318 -0.0318 -0.0318	-0.4239 -0.4239 -0.4239	-0.6717 -0.6717 -0.6717	-0.0892 -0.0892 -0.0892
3.8571	-0.5179 -0.5016 -0.5016	-0.6921 -0.6921 -0.6921	-0.3971 -0.3971 -0.3971	-0.2420 -0.2420 -0.2420	-0.0997 -0.0997 -0.0997	-0.1008 -0.1008 -0.1008	-0.2400 -0.2400 -0.2400	-0.0596 -0.0596 -0.0596	-0.6931 -0.6931 -0.6931	-0.5199 -0.5199 -0.5199
4.7143	-0.3416 -0.0563 -0.0563	-0.6079 -0.6079 -0.6079	-0.0591 -0.0591 -0.0591	-0.0221 -0.0221 -0.0221	-0.0193 -0.0193 -0.0193	-0.0195 -0.0195 -0.0195	-0.0289 -0.0289 -0.0289	-0.0502 -0.0502		

AERODYNAMIC PRESSURE ON THE BODY - - CENTROID OF FINITE ELEMENT AS GENERATED BY MSNC CODE IN THE
SAME LOCAL COORDINATE SYSTEM

ELEMENT	X	THETA	PRESSURE	ELEMENT	X	THETA	PRESSURE
1	.269E+00	.345E+03	.145E+01	2	.269E+00	.315E+03	.264E+01
3	.269E+00	.285E+03	.516E+01	4	.269E+00	.255E+03	.796E+01
5	.269E+00	.225E+03	.937E+01	6	.269E+00	.195E+03	.797E+01
7	.269E+00	.165E+03	.797E+01	8	.269E+00	.135E+03	.937E+01
9	.269E+00	.105E+03	.796E+01	10	.269E+00	.750E+02	.516E+01
11	.269E+00	.450E+02	.264E+01				

TOTAL LIFTING PRESSURE ON THE WING AT THE CONTROL POINT

X	Y	PRESSURE
7.2708	.9406	.1780E+02
7.7551	.9406	.1327E+02
8.2448	.9406	.9206E+01
8.7345	.9406	.7129E+01
9.2243	.9406	.6144E+01
9.7140	.9406	.5913E+01
10.2038	.9406	.5882E+01
10.6935	.9406	.6008E+01
8.0296	1.6443	.1505E+02
8.4073	1.6443	.1533E+02
8.7893	1.6443	.1307E+02
9.1712	1.6443	.1062E+02
9.5531	1.6443	.8352E+01
9.9350	1.6443	.7046E+01
10.3170	1.6443	.6275E+01
10.6989	1.6443	.6185E+01
8.7853	2.3449	.1357E+02
9.0568	2.3449	.1392E+02
9.3314	2.3449	.1430E+02
9.6060	2.3449	.1423E+02
9.8805	2.3449	.1310E+02
10.1551	2.3449	.1182E+02
10.4297	2.3449	.1010E+02
10.7043	2.3449	.8492E+01
9.2312	3.0366	.1290E+02
9.6980	3.0366	.1299E+02
9.8664	3.0366	.1314E+02
10.0352	3.0366	.1329E+02
10.2038	3.0366	.1349E+02
10.3724	3.0366	.1366E+02
10.5410	3.0366	.1378E+02
10.7096	3.0366	.1376E+02
10.2094	3.6654	.1261E+02
10.2808	3.6654	.1261E+02
10.3531	3.6654	.1262E+02
10.4254	3.6654	.1263E+02
10.4976	3.6654	.1268E+02
10.5699	3.6654	.1272E+02
10.6421	3.6654	.1276E+02
10.7164	3.6654	.1281E+02

AERODYNAMIC PRESSURE ON THE WING - - CENTROID OF FINITE ELEMENT AS GENERATED BY NSUC CODE IN THE SAME LOCAL COORDINATE SYSTEM

TOTAL NUMBER OF WINGS 1

WING NO. 1

ELEMENT	X	Y	PRESSURE	ELEMENT	X	Y	PRESSURE
400	.697E+01	.797E+00	.194E+02	401	.756E+01	.895E+00	.150E+02
402	.826E+01	.895E+00	.884E+01	403	.896E+01	.895E+00	.633E+01
404	.966E+01	.895E+00	.583E+01	405	.104E+02	.895E+00	.591E+01
406	.767E+01	.139E+01	.136E+02	407	.826E+01	.148E+01	.142E+02
408	.896E+01	.148E+01	.108E+02	409	.966E+01	.148E+01	.706E+01
410	.104E+02	.148E+01	.592E+01				

5.4.2 High-Supersonic Flow



EXPERIMENT CASE FOR CHECKING GENERALIZED MISSILE ATTITUDE MODE

MCDONNELL DOUGLAS

PRESSURES, FORCES AND MOMENTS ON ISOLATED BODY

CD = .0212 CL = .0987 CB = .0000 CPM = .3613 CM = 0.0000 CYM = 0.0000

BODY PRESSURE COEFFICIENTS (CP)

THETA(DEG)	0.0000	30.0000	60.0000	90.0000	120.0000	150.0000	180.0000
0.0000	.0362	-.0379	-.0465	-.0675	-.0991	-.1291	-.1515
1.3125	-.0391	-.0408	-.0495	-.0708	-.1029	-.1332	-.1497
2.6250	.0410	-.0427	-.0515	-.0728	-.1052	-.1357	-.1484
3.9375	-.0396	-.0413	-.0500	-.0713	-.1035	-.1338	-.1464
5.2500	-.0346	-.0362	-.0448	-.0637	-.0973	-.1272	-.1395
6.5625	-.0327	-.0344	-.0431	-.0620	-.0954	-.1251	-.1374
7.8750	-.0294	-.0311	-.0398	-.0585	-.0922	-.1220	-.1343
9.1875	-.0285	-.0302	-.0389	-.0576	-.0913	-.1211	-.1334
10.5000	-.0270	-.0287	-.0374	-.0561	-.0904	-.1202	-.1325
11.8125	-.0255	-.0272	-.0359	-.0546	-.0895	-.1193	-.1316
13.1250	-.0240	-.0257	-.0344	-.0531	-.0886	-.1184	-.1307
14.4375	-.0225	-.0242	-.0329	-.0516	-.0877	-.1175	-.1298
15.7500	-.0210	-.0227	-.0314	-.0501	-.0868	-.1166	-.1289
17.0625	-.0195	-.0212	-.0300	-.0486	-.0859	-.1157	-.1280
18.3750	-.0180	-.0197	-.0285	-.0471	-.0850	-.1148	-.1271
19.6875	-.0165	-.0182	-.0270	-.0456	-.0841	-.1139	-.1262
21.0000	-.0150	-.0167	-.0255	-.0441	-.0832	-.1130	-.1253
22.3125	-.0135	-.0152	-.0240	-.0426	-.0823	-.1121	-.1244
23.6250	-.0120	-.0137	-.0225	-.0411	-.0814	-.1112	-.1235
24.9375	-.0105	-.0122	-.0210	-.0396	-.0805	-.1103	-.1226
26.2500	-.0090	-.0107	-.0195	-.0381	-.0796	-.1094	-.1217
27.5625	-.0075	-.0092	-.0180	-.0366	-.0787	-.1085	-.1208
28.8750	-.0060	-.0077	-.0165	-.0351	-.0778	-.1076	-.1199
30.1875	-.0045	-.0062	-.0150	-.0336	-.0769	-.1067	-.1190
31.5000	-.0030	-.0047	-.0135	-.0321	-.0760	-.1058	-.1181
32.8125	-.0015	-.0032	-.0120	-.0306	-.0751	-.1049	-.1172
34.1250	0.0000	-.0017	-.0105	-.0291	-.0742	-.1040	-.1163
35.4375	.0015	0.0002	-.0090	-.0276	-.0733	-.1031	-.1154
36.7500	.0030	.0017	-.0075	-.0261	-.0724	-.1022	-.1145
38.0625	.0045	.0032	-.0060	-.0246	-.0715	-.1013	-.1136
39.3750	.0060	.0047	-.0045	-.0231	-.0706	-.1004	-.1127
40.6875	.0075	.0062	-.0030	-.0216	-.0697	-.0995	-.1118
42.0000	.0090	.0077	-.0015	-.0201	-.0688	-.0986	-.1109
43.3125	.0105	.0092	0.0000	-.0186	-.0679	-.0977	-.1100
44.6250	.0120	.0107	.0015	-.0171	-.0670	-.0968	-.1091
45.9375	.0135	.0122	.0030	-.0156	-.0661	-.0959	-.1082
47.2500	.0150	.0137	.0045	-.0141	-.0652	-.0950	-.1073
48.5625	.0165	.0152	.0060	-.0126	-.0643	-.0941	-.1064
49.8750	.0180	.0167	.0075	-.0111	-.0634	-.0932	-.1055
51.1875	.0195	.0182	.0090	-.0096	-.0625	-.0923	-.1046
52.5000	.0210	.0197	.0105	-.0081	-.0616	-.0914	-.1037
53.8125	.0225	.0212	.0120	-.0066	-.0607	-.0905	-.1028
55.1250	.0240	.0227	.0135	-.0051	-.0598	-.0896	-.1019
56.4375	.0255	.0242	.0150	-.0036	-.0589	-.0887	-.1010
57.7500	.0270	.0257	.0165	-.0021	-.0580	-.0878	-.1001
59.0625	.0285	.0272	.0180	-.0006	-.0571	-.0869	-.0992
60.3750	.0300	.0287	.0195	0.0009	-.0562	-.0860	-.0983
61.6875	.0315	.0302	.0210	.0024	-.0553	-.0851	-.0974
63.0000	.0330	.0317	.0225	.0039	-.0544	-.0842	-.0965
64.3125	.0345	.0332	.0240	.0054	-.0535	-.0833	-.0956
65.6250	.0360	.0347	.0255	.0069	-.0526	-.0824	-.0947
66.9375	.0375	.0362	.0270	.0084	-.0517	-.0815	-.0938
68.2500	.0390	.0377	.0285	.0099	-.0508	-.0806	-.0929
69.5625	.0405	.0392	.0300	.0114	-.0499	-.0797	-.0920
70.8750	.0420	.0407	.0315	.0129	-.0490	-.0788	-.0911
72.1875	.0435	.0422	.0330	.0144	-.0481	-.0779	-.0902
73.5000	.0450	.0437	.0345	.0159	-.0472	-.0770	-.0893
74.8125	.0465	.0452	.0360	.0174	-.0463	-.0761	-.0884
76.1250	.0480	.0467	.0375	.0189	-.0454	-.0752	-.0875
77.4375	.0495	.0482	.0390	.0204	-.0445	-.0743	-.0866
78.7500	.0510	.0497	.0405	.0219	-.0436	-.0734	-.0857
80.0625	.0525	.0512	.0420	.0234	-.0427	-.0725	-.0848
81.3750	.0540	.0527	.0435	.0249	-.0418	-.0716	-.0839
82.6875	.0555	.0542	.0450	.0264	-.0409	-.0707	-.0830
84.0000	.0570	.0557	.0465	.0279	-.0400	-.0698	-.0821
85.3125	.0585	.0572	.0480	.0294	-.0391	-.0689	-.0812
86.6250	.0600	.0587	.0495	.0309	-.0382	-.0680	-.0803
87.9375	.0615	.0602	.0510	.0324	-.0373	-.0671	-.0794
89.2500	.0630	.0617	.0525	.0339	-.0364	-.0662	-.0785
90.5625	.0645	.0632	.0540	.0354	-.0355	-.0653	-.0776
91.8750	.0660	.0647	.0555	.0369	-.0346	-.0644	-.0767
93.1875	.0675	.0662	.0570	.0384	-.0337	-.0635	-.0758
94.5000	.0690	.0677	.0585	.0399	-.0328	-.0626	-.0749
95.8125	.0705	.0692	.0600	.0414	-.0319	-.0617	-.0740
97.1250	.0720	.0707	.0615	.0429	-.0310	-.0608	-.0731
98.4375	.0735	.0722	.0630	.0444	-.0301	-.0599	-.0722
99.7500	.0750	.0737	.0645	.0459	-.0292	-.0590	-.0713
101.0625	.0765	.0752	.0660	.0474	-.0283	-.0581	-.0704
102.3750	.0780	.0767	.0675	.0489	-.0274	-.0572	-.0695
103.6875	.0795	.0782	.0690	.0504	-.0265	-.0563	-.0686
105.0000	.0810	.0797	.0705	.0519	-.0256	-.0554	-.0677
106.3125	.0825	.0812	.0720	.0534	-.0247	-.0545	-.0668
107.6250	.0840	.0827	.0735	.0549	-.0238	-.0536	-.0659
108.9375	.0855	.0842	.0750	.0564	-.0229	-.0527	-.0650
110.2500	.0870	.0857	.0765	.0579	-.0220	-.0518	-.0641
111.5625	.0885	.0872	.0780	.0594	-.0211	-.0509	-.0632
112.8750	.0900	.0887	.0795	.0609	-.0202	-.0500	-.0623
114.1875	.0915	.0902	.0810	.0624	-.0193	-.0491	-.0614
115.5000	.0930	.0917	.0825	.0639	-.0184	-.0482	-.0605
116.8125	.0945	.0932	.0840	.0654	-.0175	-.0473	-.0596
118.1250	.0960	.0947	.0855	.0669	-.0166	-.0464	-.0587
119.4375	.0975	.0962	.0870	.0684	-.0157	-.0455	-.0578
120.7500	.0990	.0977	.0885	.0699	-.0148	-.0446	-.0569
122.0625	.1005	.0992	.0900	.0714	-.0139	-.0437	-.0560
123.3750	.1020	.1007	.0915	.0729	-.0130	-.0428	-.0551
124.6875	.1035	.1022	.0930	.0744	-.0121	-.0419	-.0542
125.9375	.1050	.1037	.0945	.0759	-.0112	-.0410	-.0533
127.2500	.1065	.1052	.0960	.0774	-.0103	-.0401	-.0524
128.5625	.1080	.1067	.0975	.0789	-.0094	-.0392	-.0515
129.8750	.1095	.1082	.0990	.0804	-.0085	-.0383	-.0506
131.1875	.1110	.1097	.1005	.0819	-.0076	-.0374	-.0497
132.5000	.1125	.1112	.1020	.0834	-.0067	-.0365	-.0488
133.8125	.1140	.1127	.1035	.0849	-.0058	-.0356	-.0479
135.1250	.1155	.1142	.1050	.0864	-.0049	-.0347	-.0470
136.4375	.1170	.1157	.1065	.0879	-.0040	-.0338	-.0461
137.7500	.1185	.1172	.1080	.0894	-.0031	-.0329	-.0452
139.0625	.1200	.1187	.1095	.0909	-.0022	-.0320	-.0443
140.3750	.1215	.1202	.1110	.0924	-.0013	-.0311	-.0434
141.6875	.1230	.1217	.1125	.0939	0.0000	-.0302	-.0425
143.0000	.1245	.1232	.1140	.0954	.0009	-.0293	-.0416
144.3125	.1260	.1247	.1155	.0969	.0018	-.0284	-.0407
145.6250	.1275	.1262	.1170	.0984	.0027	-.0275	-.0398
146.9375	.1290	.1277	.1185	.0999	.0036	-.0266	-.0389
148.2500	.1305	.1292	.1200	.1014	.0045	-.0257	-.0380
149.5625	.1320	.1307	.1215	.1029	.0054	-.0248	-.0371
150.8750	.1335	.1322	.1230	.1044	.0063	-.0239	-.0362
152.1875	.1350	.1337	.1245	.1059	.0072	-.0230	-.0353
153.5000	.1365	.1352	.1260	.1074	.0081	-.0221	-.0344
154.8125	.1380	.1367	.1275	.1089	.0090	-.0212	-.0335
156.1250	.1395	.1382	.1290	.1104	.0099	-.0203	-.0326
157.4375	.1410	.1397	.1305	.1119	.0108	-.0194	-.0317
158.7500	.1425	.1412	.1320	.1134	.0117	-.0185	-.0308
160.0625	.1440	.1427	.1335	.1149	.0126	-.0176	-.0299
161.3750	.1455	.1442	.1350	.1164	.0135	-.0167	-.0290
162.6875	.1470	.1457	.1365	.1179	.0144	-.0158	-.0281
164.0000	.1485	.1472	.1380	.1194	.0153	-.0149	-.0272
165.3125	.1500	.1487	.1395	.1209	.0162	-.0140	-.0263
166.6250	.1515	.1502	.1410	.1224	.0171	-.0131	-.0254
167.9375	.1530	.1517	.1425	.1239	.0180	-.0122	-.0245
169.2500	.1545	.1532	.1440	.1254	.0189	-.0113	-.0236
170.5625	.1560	.1547	.1455	.1269	.0198	-.0104	-.0227
171.8750	.1575	.1562	.1470	.1284	.0207	-.0095	-.0218
173.1875	.1590	.1577	.1485	.1299	.0216	-.0086	-.0209
174.5000	.1605	.1592	.1500	.1314	.0225	-.0077	-.0200
175.8125	.1620	.1607	.1515	.1329	.0234	-.0068	-.0191
177.1250	.1635	.1622	.1530	.1344	.0243	-.0059	-.0182
178.4375	.1650	.1637	.1545	.1359	.0252	-.0050	-.0173
179.7500	.1665	.1652	.1560	.1374	.0261	-.0041	-.0164
181.0625	.1680	.1667	.1575	.1389	.0270	-.0032	-.0155
182.3750	.1695	.1682	.1590	.1404	.0279	-.0023	-.0146
183.6875	.1710	.1697	.1605	.1419	.0288	-.0014	-.0137
185.0000	.1725	.1712	.1620				

FORCES AND MOMENTS ON WING											
CD =	.0191	CL =	.1677	CG =	.0000	CPH =	-1.5614	CRM =	.2992	CYM =	-.0339
UPPER SURFACE WING PANEL PRESSURE COEFF. (CP)											
SPANWISE STATION											
CHORDWISE STATION											
	1	2	3	4	5						
1	-.03429	-.03429	-.03429	-.03429	-.03429						
2	-.03429	-.03429	-.03429	-.03429	-.03429						
3	-.03429	-.03429	-.03429	-.03429	-.03429						
4	-.03429	-.03429	-.03429	-.03429	-.03429						
5	-.03429	-.03429	-.03429	-.03429	-.03429						
6	-.03429	-.03429	-.03429	-.03429	-.03429						
7	-.03429	-.03429	-.03429	-.03429	-.03429						
8	-.03429	-.03429	-.03429	-.03429	-.03429						
LOWER SURFACE WING PANEL PRESSURE COEFF. (CP)											
SPANWISE STATION											
CHORDWISE STATION											
	1	2	3	4	5						
1	.13455	.13455	.13455	.13455	.13455						
2	.13455	.13455	.13455	.13455	.13455						
3	.13455	.13455	.13455	.13455	.13455						
4	.13455	.13455	.13455	.13455	.13455						
5	.13455	.13455	.13455	.13455	.13455						
6	.13455	.13455	.13455	.13455	.13455						
7	.13455	.13455	.13455	.13455	.13455						
8	.13455	.13455	.13455	.13455	.13455						

AERODYNAMIC PRESSURE ON THE BODY -- CENTROID OF FINITE ELEMENT AS GENERATED BY NSMC CODE IN THE
 SAME LOCAL COORDINATE SYSTEM

ELEMENT	X	THETA	PRESSURE	ELEMENT	X	THETA	PRESSURE
1	.269E+00	.345E+03	.269E+01	2	.269E+00	.315E+03	.296E+01
3	.269E+00	.285E+03	.403E+01	4	.269E+00	.255E+03	.589E+01
5	.269E+00	.225E+03	.816E+01	6	.269E+00	.195E+03	.962E+01
7	.269E+00	.165E+03	.962E+01	8	.269E+00	.135E+03	.816E+01
9	.269E+00	.105E+03	.589E+01	10	.269E+00	.750E+02	.403E+01
11	.269E+00	.450E+02	.296E+01				

FORCES AND MOMENTS ON WING											
CD =	.0191	CL =	.1677	CB =	.0000	CPM =	-1.5614	CRM =	.2992	CYM =	-.0339
UPPER SURFACE WING PANEL PRESSURE COEFF. (CP)											
SPANWISE STATION											
CHORDWISE STATION											
	1	2	3	4	5						
1	-.03429	-.03429	-.03429	-.03429	-.03429						
2	-.03429	-.03429	-.03429	-.03429	-.03429						
3	-.03429	-.03429	-.03429	-.03429	-.03429						
4	-.03429	-.03429	-.03429	-.03429	-.03429						
5	-.03429	-.03429	-.03429	-.03429	-.03429						
6	-.03429	-.03429	-.03429	-.03429	-.03429						
7	-.03429	-.03429	-.03429	-.03429	-.03429						
8	-.03429	-.03429	-.03429	-.03429	-.03429						
LOWER SURFACE WING PANEL PRESSURE COEFF. (CP)											
SPANWISE STATION											
CHORDWISE STATION											
	1	2	3	4	5						
1	.13455	.13455	.13455	.13455	.13455						
2	.13455	.13455	.13455	.13455	.13455						
3	.13455	.13455	.13455	.13455	.13455						
4	.13455	.13455	.13455	.13455	.13455						
5	.13455	.13455	.13455	.13455	.13455						
6	.13455	.13455	.13455	.13455	.13455						
7	.13455	.13455	.13455	.13455	.13455						
8	.13455	.13455	.13455	.13455	.13455						

TOTAL LIFTING PRESSURE ON THE WING AT THE CONTROL POINT

X	Y	PRESSURE
7.2708	.9406	.1156E+02
7.7551	.9406	.1156E+02
8.2448	.9406	.1156E+02
8.7345	.9406	.1156E+02
9.2243	.9406	.1156E+02
9.7140	.9406	.1156E+02
10.2038	.9406	.1156E+02
10.6935	.9406	.1156E+02
8.0296	1.6443	.1156E+02
8.4073	1.6443	.1156E+02
8.7893	1.6443	.1156E+02
9.1712	1.6443	.1156E+02
9.5531	1.6443	.1156E+02
9.9350	1.6443	.1156E+02
10.3170	1.6443	.1156E+02
10.6989	1.6443	.1156E+02
8.7853	2.3449	.1156E+02
9.0568	2.3449	.1156E+02
9.3316	2.3449	.1156E+02
9.6060	2.3449	.1156E+02
9.8805	2.3449	.1156E+02
10.1551	2.3449	.1156E+02
10.4297	2.3449	.1156E+02
10.7043	2.3449	.1156E+02
9.5312	3.0366	.1156E+02
9.6980	3.0366	.1156E+02
9.8666	3.0366	.1156E+02
10.0352	3.0366	.1156E+02
10.2038	3.0366	.1156E+02
10.3724	3.0366	.1156E+02
10.5410	3.0366	.1156E+02
10.7096	3.0366	.1156E+02
10.2094	3.6654	.1156E+02
10.2808	3.6654	.1156E+02
10.3531	3.6654	.1156E+02
10.4256	3.6654	.1156E+02
10.4976	3.6654	.1156E+02
10.5699	3.6654	.1156E+02
10.6421	3.6654	.1156E+02
10.7146	3.6654	.1156E+02

AERODYNAMIC PRESSURE ON THE WING - - CENTROID OF FINITE ELEMENT AS GENERATED BY NSMC CODE IN THE SAME LOCAL COORDINATE SYSTEM

TOTAL NUMBER OF WINGS		1							
WING NO.		1							
ELEMENT	X	Y	PRESSURE	ELEMENT	X	Y	PRESSURE	ELEMENT	PRESSURE
400	.697E+01	.797E+00	.116E+02	401	.756E+01	.893E+00	.116E+02	401	.116E+02
402	.826E+01	.893E+00	.116E+02	403	.896E+01	.893E+00	.116E+02	403	.116E+02
404	.966E+01	.893E+00	.116E+02	405	.104E+02	.893E+00	.116E+02	405	.116E+02
406	.747E+01	.139E+01	.116E+02	407	.826E+01	.148E+01	.116E+02	407	.116E+02
408	.896E+01	.148E+01	.116E+02	409	.966E+01	.148E+01	.116E+02	409	.116E+02
410	.104E+02	.148E+01	.116E+02						

.....END OF COMPUTATIONS

Section 6
CONCLUSIONS AND RECOMMENDATIONS

The MDAC aerodynamic code has been extended to treat (a) high supersonic speeds ($M_\infty > 2.5$) and (b) generalized missile attitudes. As a result it is capable of analyzing the missile under more realistic flight conditions. Effort for input preparation remains the same for all practical purposes because only a few inputs were added for the specification of attitudes. Computing time for high supersonic flow analysis will be only a small fraction of that for the low-speed case. The original flexibility in the coordinate system has been maintained for the convenience of users. Outputs are produced as computer printout and punch cards (if desired) in the format of NASTRAN bulk data.

With these additional capabilities incorporated, the code represents a versatile tool for aerodynamic analysis of missiles and forms a unique link between the flight dynamics and structural analysis of the missile under a wide variety of flight conditions that it may encounter in actual operation. Furthermore, the large effort necessary for preparing the input to the NASTRAN is reduced when it is used in combination with the NSWC computer codes PING or BING.

For missile flight at high supersonic speeds ($2 \leq M_\infty \leq 8$) the problem of aerodynamic heating becomes important and needs to be analyzed. As an example, for an insulated flat plate aligned parallel to the flow, the surface temperature can reach anywhere from 1000°R to 7000°R for a speed range of $M_\infty = 2$ to 8 . Additionally, during maneuvering, severe heating may occur as a result of shock impingement on the missile, e.g., nose bow shock-wing interaction. Thus, the induced thermal stress can seriously affect the structural integrity of the missile, and failure of a missile component can result. Consequently, development of capability for analyzing the aerodynamic heating to complement the pressure-load calculation will be an appropriate task and is recommended as a future effort. The required flow-field quantities such as the pressure, velocity, and density can be derived conveniently from the present code as constructed. This feature can be expected to minimize the necessary work compared to that of an independent effort.

Section 7
REFERENCES

1. K. K. Wang. Aerodynamic Computer Code for Computing Pressure Loading on Complete Missile for Structural Analysis. MDC G7215, MDAC, Huntington Beach, California, January 1978.
2. P. C. Huang and J. P. Matra, Jr. Planform Input Generator (PING), A NASTRAN Preprocessor for Lifting Surfaces-Theoretical Development, User's Manual, and Program Listing. NOLTR-73-199, Naval Ordnance Laboratory, White Oak, Silver Spring, Maryland, December 1973.
3. P. C. Huang and J. P. Matra, Jr. Missile Body Input Generator (BING) A NASTRAN Preprocessor; Theoretical Development, User's Manual, and Program Listing. NSWC-NOL-TR-75-9, Naval Surface Weapons Center, White Oak, Silver Spring, Maryland, March 1975.
4. P. S. Epstein. On the Air Resistance of Projectiles. Proc. Nat. Acad. Sci., U.S.A. 17, 1931; pp. 532-547.
5. C. A. Syvertson and D. H. Dennis. A Second-Order Shock-Expansion Method Applicable to Bodies of Revolution Near Zero Lift. NACA Report 1328, 1957.
6. A. E. Gentry, D. N. Smyth, and W. R. Oliver. The Mark IV Supersonic-Hypersonic Arbitrary Body Program. AFFDL-TR-73-159, 2, AFFDL, Wright-Patterson Air Force Base, Ohio, November 1973.
7. A. G. Hammitt and K. R. Murphy. Approximate Solutions for Supersonic Flow Over Wedges and Cones. JAS, January 1960.
8. D. J. Jones. Tables of Inviscid Supersonic Flow About Circular Cones at Incidence, $\gamma = 1.4$. AGARDograph 137, November 1969.
9. D. J. Jones. A Pressure Formula for an Inclined Circular Cone in Supersonic Flow, $\gamma = 1.4$. AIAA Journal, February 1972; pp. 234-236.
10. E. R. A. Landers. Pressure Distribution on Planar and Cruciform Delta Wings Attached to Cylindrical Bodies in Supersonic Flow, Part II. W. R. E. Report HSA 107, Australian Defense Scientific Service, Weapons Research Establishment, Adelaide, Australia.
11. R. C. Savin. Application of the Generalized Shock-Expansion Method to Include Bodies of Revolution Traveling at High Supersonic Speeds. NACA TN 3345, April 1955.
12. M. H. Bertram and A. Henderson, Jr. Recent Hypersonic Studies of Wings and Bodies. ARS Journal, Vol. 31, No. 8, August 1961.

Appendix A

PROGRAM LISTING

MCDONNELL DOUGLAS

65

PRECEDING PAGE NOT FILMED
BLANK


```

C      ICH = 1 , DATA ARE SAVED ON TAPE 12 FOR RESTART
C      ICH = 2 , RESTART RUN, BYPASS ALL AERO CALCULATIONS
C
C      IF ( ICH .LT. 2 ) GO TO 100
C      CALL OVERLAY(4HANS, 7, 0)
C      GO TO 1100
C      100 CONTINUE
C
C      IOVR10 = 0
C      CALL OVERLAY(4HWOOD, 1, 0)
C
C      CALL AERODYNAMIC LINK
C      1050 CALL AERO
C
C      DUMMY READ AND WRITE ON TAPE 9 IN AN UNINFORMED FORM TO
C      MOVE SYSTEM ROUTINE W,50 INTO OVERLAY( 0, 0 ) TO AVOID
C      LOADING ERRORS.
C      ..... DO NOT REMOVE .....
C
C      ICH = 1
C      IF ( ICH .EQ. 1 ) GO TO 1100
C      READ (9) DUM
C      WRITE (9) DUM
C
C      1100 CONTINUE
C      WRITE (6,901)
C      901 FORMAT (1H0////.....END OF COMPUTATIONS .....*)
C
C      .....
C      STOP
C      END
C
C      SUBROUTINE AERO
C
C      .....
C      CONTROL ROUTINE FOR AERODYNAMIC LINKS
C      .....
C      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPEF,NTAPEG,NTAPEH,NTAPEI,
C      1 NTAPEJ,NTAPEK,NTAPEL,NTAPEM,NTAPEN,NTAPEO,NTAPEP,NTAPEQ,NTAPER,
C      COMMON /TRANSFER/ IOVR10,IOVR60,IOVR70
C      COMMON /ENDJOB/ JOBEED
C      COMMON /CONJOB/ ISAVET
C
C      REWIND NTAPEA
C      REWIND NTAPEB
C      REWIND NTAPEC
C      REWIND NTAPEF
C      REWIND NTAPEG
C      REWIND NTAPER
C
C      CALL OVERLAY(4HWOOD,3,0,4HRECALL)
C      GO TO ( 500, 1070, 2000), KACE
C

```

```

C      WING ALONE, KACE = 1
C      500 CONTINUE
C      IOVR70 = 1
C      CALL OVERLAY(4HWANG, 7, 0)
C      IOVR70 = 4
C      CALL OVERLAY(4HWANG, 7, 0)
C      GO TO 3000
C      BODY ALONE, KACE = 2
C      1070 CONTINUE
C      GO TO 3000
C      WING - BODY COMBINATION, KACE = 3
C      2000 CONTINUE
C      IOVR70 = 2
C      CALL OVERLAY(4HWANG, 7, 0)
C      CALL OVERLAY(4HWOOD, 6, 0, 0HRECALL)
C      IOVR70 = 3
C      CALL OVERLAY(4HWANG, 7, 0)
C      CALL OVERLAY(4HWOOD, 5, 0, 0HRECALL)
C      IOVR70 = 4
C      CALL OVERLAY(4HWANG, 7, 0)
C      3000 CONTINUE
C      IOVR60 = 1
C      CALL OVERLAY(4HWOOD, 6, 0, 0HRECALL)
C      IOVR70 = 0
C      CALL OVERLAY(4HWANG, 7, 0)
C      -----
C      RETURN
C      -----
C      END
C      SUBROUTINE FOR EOF (IT)
C      .....
C      WRITES E.P. ON A TAPE TO INDICATE THE END OF A DATA SET
C      .....
C      INTEGER EOF
C      DATA EOF /4NE.F./
C      WRITE (IT) EOF
C      -----
C      RETURN
C      -----
C      END
C      SUBROUTINE PSF (NF, NTAPE, IRR)
C      .....
C      READS A TAPE UNTIL IT HAS READ THE CHARACTERS E.P.

```

```

C .....
C      INTEGER IW
C      INTEGER EOF
C
C      DATA EOF /4ME.F./
C
C      IRR = 0
C      K = 0
C      DO 1000 I = 1, 10000
C      READ(NTAPE) IW
C      IF (IW .NE. EOF) GO TO 1000
C      K = K + 1
C      IF (K .GE. NF) RETURN
C      1000 CONTINUE
C      IRR = NF - K
C
C      -----
C      RETURN
C      -----
C      END
C
C      FUNCTION INTURP(IDIC,NDIC,LI,LO)
C
C      .....
C      READ AND WRITE A COMMAND CARD, SEE IF COMMAND (IN THE
C      FIRST WORD) IS IN THE DICTIONARY.
C      .....
C
C      INTEGER IDIC(1), COM(20), ICON
C      EQUIVALENCE (COM(1),ICON)
C
C      IDIC (INSTEAD OF DIC) IS USED AS AN ARGUMENT
C      SO THAT A FIXED-POINT COMPARISON CAN BE MADE
C
C      READ (LI,1040) COM
C      IF (LO) 1010, 1010, 1000
C      1000 WRITE (LO,1030) COM
C      1010 DO 1020 INTERP=1,NDIC
C      IF (IDIC(INTERP) .EQ. ICON) GO TO 1030
C      1020 CONTINUE
C
C      INTERP=0
C      1030 INTURP=INTERP
C
C      -----
C      RETURN
C      -----
C
C      1040 FORMAT(20A4)
C      1050 FORMAT(/1H0,20A4)
C      END
C
C      OVERLAY(NDIC,1,0)
C      LIST,NONE
C      PROGRAM CCOF
C      COMMON/ MAIN / NTAPE,NTAPEC,NTAPE,NTAPEE,NTAPEF,NTAPEI,
C      1 NTAPED,MBODY,MNINE,SHACH,STH,KACE,MPOLAR,IRN
C      COMMON /TNSFR/ IOWR10,IOWR60,IOWR70
C      200 CALL PANEL
C
C      END
C      SUBROUTINE PANEL

```

```

C
C
COMMON/ MAIN / LA,NTAPE,NTAPEC,NTAPED,LE,LF,LL,LO,NBODY,MWING,
1  XNACH, SYM, KACE, NPOLAR, IRM
COMMON /XYWING/ XW(13,11),YW(11),ZW(11)
COMMON/DSXB / XBS1, XBS2, XBS1, XBS2
COMMON/DSXB / XBS1, XBS2, XBS1, XBS2
DIMENSION XB(51),RB(51),ZD(51),THETA(11),X(11),Y(11),Z(11),BC(10),
1  XBAR(50),YBAR(10),ZBAR(10),XC(50),THET(11),CHORD(50),AREA(50,10),
2  XW(10),ZBR(110,10),XBR(110,10),YBR(110,10),CW(110,10),THETW(10)
3  ,ARW(110,10),XCN(110,10)
C
C
REVING NTAPEC
XCPT = 0.95
I=1
ZD=0.
ON=1.
IF ( NBODY ) 100, 100, 10
10 READ (LI,1010) BODY, DTNET, PLANE, RP, YAW
NBOYS=BODY
NTNET=DTNET
MPLANE=PLANE
NLS=0
READ (LI,1010) ( X(N), N=1,NBOYS)
READ (LI,1010) ( Y(N), N=1,NBOYS)
READ (LI,1010) ( Z(N), N=1,NBOYS)
READ (LI,1010) ( THETA(N), N=1,NTNET )
READ (LI,1010) XBS1, XBS2, XBS1, XBS2
DO 15 N=1,NTNET
15 THETA(N)=THETA(N)/57.2957795
IF(PLANE) 300,300,20
C
C
BODY-WING CASE, REQUIRES PANELED BODY
C
20 CONTINUE
READ (LI,1010) ( X(N), N=1,MPLANE)
ALPHA=0
MPLT=MPLANE-1
MROU=NTNET-1
C
C
IF(RP) 22,22,28
C
C
NON-CIRCULAR BODY
C
22 CONTINUE
READ (LI,1010) ( Y(N), N=1,NTNET)
READ (LI,1010) ( Z(N), N=1,NTNET)
GO TO 35
C
C
CURCULAR BODY
C
28 DO 30 N=1,NTNET
Y(N)=RP*SIN(THETA(N))
30 Z(N)=RP*COS(THETA(N))
35 DO 40 N=1,NBOU
YBAR(N)=(Y(N)+Y(N+1))/2.
ZBAR(N)=(Z(N)+Z(N+1))/2.
DO 50 M=1,NPLM
XBAR(M)=(X(N+1)+X(N))/2.
CHORD(M)=X(N+1)-X(N)
50 XC(M)=X(N)+XCPT*CHORD(M)
DO 60 M=1,NROU
60 THE(M)=YBAR2((Z(N+1)-Z(N)), (Y(N+1)-Y(N)))
WRITE(60,900)

```

```

WRITE(LO,910)
NP=0
DO 70 M=1,NROW
DO 70 M=1,NPLM1
NP=NP+1
WRITE(LO,920) NP,I,X(M),Y(M),Z(M),X(M),Y(M),Z(M+1),
1X(M+1),Y(M),Z(M),X(M+1),Y(M+1),Z(M+1)
70 WRITE (NTAPE) NP,I,X(M),Y(M),Z(M),X(M),Y(M),Z(M+1),
1X(M+1),Y(M),Z(M),X(M+1),Y(M+1),Z(M+1)
WRITE(LO,940)
WRITE(LO,950)
NP=0
DO 80 M=1,NROW
DO 80 M=1,NPLM1
NP=NP+1
AREA(M,N)=CHORD(M)*BC(M)
WRITE(LO,960) NP,XBAR(M),YBAR(M),ZBAR(M),XC(M),YBAR(M),ZBAR(M),
1AREA(M,N),THET(M),ZC,CHORD(M)
80 WRITE (NTAPE) NP,XBAR(M),YBAR(M),ZBAR(M),XC(M),YBAR(M),ZBAR(M),
1AREA(M,N),THET(M),ZC,CHORD(M)
WRITE (NTAPE) NPLM1,XCPT
NBOBY=NP
IS(MVING)500,500,120
100 IF(MVING)500,500,110
C
C BODY ONLY OR BODY-WING CASE
C
110 CONTINUE
NBOBY=0
PLANE=0.
120 NP=0
WRITE(LO,970)
WRITE(LO,980)
DO 200 L=1,MVING
CALL WINGP(MB,MC,YO,ZA,DIM)
THET(L)=DIN
WRITE(LO,1020) L,YO,ZA,DIN
MB=MB+1
MC=MC+1
XW(L)=MB+MC+1
DO 150 M=1,MB1
DO 150 M=1,MC1
NP=NP+1
WRITE (NTAPE) NP,I,XW(M,N),YW(M,N),ZW(M,N),XW(M,N+1),YW(M,N+1),ZW(M,N+1),
1XW(M+1,N),YW(M+1,N),ZW(M+1,N),XW(M,N),YW(M,N),ZW(M,N),XW(M,N+1),YW(M,N+1),ZW(M,N+1),
1XW(M+1,N),YW(M+1,N),ZW(M+1,N)
K=N*(M+1)+NC1
CX=XW(M+1,N)-XW(M,N)
CY=YW(M+1,N)-YW(M,N)
CT=ZW(M+1,N)-ZW(M,N)
RT=(1.-CT/(CX*CT))/5.0
RT1=1.0-RT
XWL=RT*XW(M,N+1)+RT1*XW(M,N)
YWL=RT*YW(M,N+1)+RT1*YW(M,N)
ZWL=RT*ZW(M,N+1)+RT1*ZW(M,N)
CW(K,L)=XWL-ZWL
TOUN=50RT1*(YWL(M+1)-YW(M))**2+(ZW(M+1)-ZW(M))**2)
XW(K,L)=YOUN+.5*(CX*CT)
YWL(K,L)=(XWL+ZWL)/2.
ZWL(K,L)=YW(M+1)+RT*YW(M)+RT1
1ZWL(K,L)=ZW(M+1)+RT*ZW(M)+RT1
150 XW(K,L)=XCPT,XW(K,L)=1.0-XCPT,XWL
200 CONTINUE
WRITE(LO,990)
WRITE(LO,1000)
NP=0

```



```

NB=NB
NC=NC
NCA=CASE
PERCT = 0.1

CRR = CR - X0
DCR = ( 1.0 - PERCT ) * CRR / ( XNC - 2.0 )
XN(1)=X0
XN(2) = XN(1) + PERCT* CRR
DO 145 M=3,NC
  XN(M) = XN(M-1) + DCR
145 CONTINUE
BY = B - Y0
DBY=BY/(XNB-1.)
YM(1)=Y0
DO 165 M=2,NB
  YM(M) = YM(M-1) + DBY
165 CONTINUE
SLE=(A-X0)/(B-Y0)
STE=(C-CR)/(B-Y0)
IF(SLE-STE)210,230,210
210 CONTINUE
  XZ = ( C-A ) / ( CR - X0 )
  DO 215 M=1,NC
    XY(M,NB) = A + BX*( XN(M) - X0 )
215 CONTINUE
  DO 220 M=1,NB
    DT(M)=(YM(M)-Y0)/(B-Y0)
  DO 220 M=1,NC
    XY(M,M) = XN(M) + ( XY(M,NB) - XN(M) ) * DT(M)
220 CONTINUE
  DO 230 M=1,NB
    DT(M)=YM(M)-Y0
  DO 240 M=1,NC
    XY(M,M) = XN(M) + SLE*DT(M)
240 CONTINUE
250 CONTINUE
CALL WROTAT(NB,Y0,ZA,DIH)
RETURN
1010 FORMAT(7F10.0)
END

C
SUBROUTINE WROTAT(NB,Y0,ZA,UNGDIN)
COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, MRODT, RWING, XNACH,
  STN, KACE, NPOLAR, IAN
1
COMMON /XYWING/ XY(13,11),YM(11),ZW(11)
DIMENSION DPYVT(11)
UNGDIN=0.
READ (11,1020) PIVOT, UNGDIN, WARG
PIVOT=Y0
ZPIVOT=ZA
C PIVOT, BIPEDRAL INDICATOR
C ZPIVOT, ZPIVOT, PIVOT POINT
C UNGDIN, BIPEDRAL ANGLE
  DO 50 I=1,NB
    ZW(I)=ZA
  50 ZW(I)=ZA
C
IF (PIVOT .LT. 0.01 ) RETURN
  DO 100 I=1,NB
    DPYVT(I)=YM(I)-ZPIVOT
  100 CONTINUE
  UNGDIN=UNGDIN/57.2957795
  COSDIH=COS(UNGDIN)
  SINDIH=SIN(UNGDIN)

```



```

30 CONTINUE
XND = XNDB
XNC = XNCC
XND = XND
XNC = XNC
XBODY = 0
XCORD = XC
XCORD = XND

C FIRST RUN INW .LT. 2
C RESTART RUN INW = 2
C
C IF ( INW .EQ. 2 ) GO TO 500
C
C READ IN COORDINATE SYSTEM SPECIFICATION FOR BODY AND WING AND
C COMPUTE TRANSFORMATION MATRICES.
C
C IF ( ICASE .NE. 1 ) XBODY = 1
C AMACEL = ABS( XMACEL )
C MACELS = IFIX(AMACEL)
C MW2 = 0
C IF ( ICASE .EQ. 2 ) GO TO 45
C DO 40 I=1, MWING
C XCORD = 1
C IF ( IFORM(1) .EQ. 1 ) XCORD = 2
C MW2 = MW2 + XCORD
C
C 40 CONTINUE
C 45 CONTINUE
C XN = XBODY + MW2 + MACELS
C DO 50 I=1, NS
C XSYN = 1
C CALL CORDTR
C
C 50 CONTINUE
C
C IF ( ICASE .EQ. 2 ) GO TO 70
C
C TRANSFORM WING COORDINATES TO THAT OF AERO
C
C XSYN = XBODY
C DO 60 I = 1, MWING
C XSYN = XSYN + 1
C MW = 2 + I - 1
C CALL WTRANS( MW, I )
C CALL CORDTR( MW, I )
C IF ( IFORM(1) .NE. 1 ) GO TO 60
C XSYN = XSYN + 1
C MW = MW + 1
C CALL WTRANS( MW, I )
C CALL CORDTR( MW, I )
C
C 60 CONTINUE
C IF ( MACELS .EQ. 0 ) GO TO 70
C
C TRANSFORM THE MACEL AXIS IN TO THAT OF THE BODY AXIS
C
C DO 51 K = 1, MACELS
C XSYN = 1 + MW2 + K
C NT = WTRANS( XSYN )
C DO 51 ITRAN = 1, NT
C DO 51 J = 1, MACELP
C XN(I) = XN(J) + VA( 1, ITRAN, XSYN )
C
C 51 CONTINUE
C 70 CONTINUE
C

```

FOR HIGH SUPERSONIC CASE, BYPASS THE GEOMETRY AND AERO
ANALYSIS SECTION

IF (XNACH .LT. 2.5) GO TO 80

CALL OVERLAY(4NHANG, 2, 1)

GO TO 500

80 CONTINUE

PREPARE INPUTS AND WRITE ON TAPE 11 FOR AERO CODE

ICASE = 1, WING ONLY

ICASE = 2, BODY ONLY

ICASE = 3, COMPLETE MISSILE

GO TO (100, 200, 300) , ICASE

WING ONLY CASE

100 CONTINUE

KACE = 1

CALL WINGIN

GO TO 500

200 CONTINUE

BODY ONLY CASE

KACE = 2

NBODY = 1

XNACH = 0.0

NWING = 0.0

PLANE = 0.0

CALL BODYIN

GO TO 500

300 CONTINUE

COMPLETE MISSILE CASE

KACE = 3

CALL BARNIN

500 CONTINUE

1000 FORMAT(20A4)

1010 FORMAT(1M1,20A4)

1020 FORMAT(7F10.0)

END

SUBROUTINE BODYPW(XLAST)

PANEL BODY SECTION TO CORRESPONDING WITH THE WING PANEL FOR
AERODYNAMIC COMPUTATIONS

COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI, LO, NBODY, NWING, XNACH,

1 STM, KACE, NPOLAR, IRU

COMMON/ BODY2 / NBPFL, NBDOTS, NTHET, XB(51), RB(51), ZB(51),

1 THETA(1), RP, IN(20)

COMMON/ BODYSP / DBOX(51), DBPX(51), DBPC(51)

COMMON/ CORTAN/ NSTH, NTRANS(20), ISTH(20), VAC(3,2,20), STPM(9,2,20)

```

COMMON/DEFAU / BODY, BODYTS, BTHET, CPCALC, NCASE, NRUN, DAGEG, ARB
1  PLANE, POLAR, RPP, XNACEL, SYMR, YTHICK, WING, XNACHH, INB
2  XNC, PINT, IPURCH, CKASE, ARN(10), IRON,
3  YAN
COMMON/WINGSM/ XW(10,20), YW(10,20), ZW(10,20), MWPI(20), NCMORD,
1  WSPAN, ISOLIG, NPLANE, ICAMBR, ITWIST, ITHICK, IFORM(10)
2  PIVOT(10), DIM(10), NUPANL, XLG(10), XTG(10)
COMMON/POINTP/ TY(4), TYP(4), TYP(4)

```

```

XCPT = 0.95
PERCT = 0.1
NC = XNC
NTHET = BTHET
NCK = NUPANL * NTHET * (NC - 1)
IF ( NCK .GT. 100) NC = NC / 2
XNC = FLOAT(NC)
NC2 = NC - 2
IF ( NC2 .LE. 0 ) NC2 = 1
NPC = 1

```

DETERMINE BODY PANEL RADIIUS RP

```

YTHIN = YTH( 1,1)
NW2 = 2 * NWING
XMIN = XW( 1, 1)
XMAX = XW( 2, 1)
IF ( NWING .EQ. 1 ) GO TO 205
DO 200 I=3,NW2,2
J = 1
NW1 = I/2 + 1
IF ( IFORM(NW1) .EQ. 2 ) J = NW1
YTHIN = ANINT( YTH(1,J), YTHIN )
200 CONTINUE
205 CONTINUE
RP = YTHIN
RPP = RP

```

PANEL BODY USING INPUT OF NUPANL, XLG, XTG

```

WSPAN = XTG(1) - XLG(1)
DX1 = 0.2 * WSPAN
DX2 = ( 1.0 - PERCT ) * WSPAN / NC2
XW(1) = XLG(1) - DX1
XW(2) = XLG(1)
XW(3) = XLG(1) + PERCT * WSPAN
IC = 3
DO 300 I=3,NC
IC = IC + 1
XW(IC) = XW(IC-1) + DX2
300 CONTINUE
305 CONTINUE
NPC = NPC + 1
IF ( NPC .GT. NUPANL ) GO TO 700
IF ( ABS( XLG(NPC) - XTG(NPC-1) ) .LE. 1.0E-3 ) GO TO 310
IC = IC + 1
XW(IC) = XW(IC-1) + 0.5 * ( XLG(NPC) - XTG(NPC-1) )
IC = IC + 1
XW(IC) = XLG(NPC)
310 CONTINUE
WSPAN = XTG(NPC) - XLG(NPC)
DX2 = ( 1.0 - PERCT ) * WSPAN / NC2
IC = IC + 1

```



```

250 CONTINUE
C
RETURN
END
C
SUBROUTINE BODYPR( X0, R0, Z0, THETA, NP, NTHET, NBOOTS, X0S1,
1 X0S2, Y0S1, Y0S2, XLAST)
C
C GENERATES BODY PROFILE R = R(X), Z0 = Z0(X) AND MERIDIAN ANGLE
C THETA.
C COMPUTES SLOPES DS01, DS02
C
COMMON/ MAIN / LA, LB, LC, LP, LE, LF, LJ, LO, MBOOT, MWING, IMACHN,
1 STNR, KACE, MPOLAR, INR
COMMON / BODYSP/ B0X(S1), B2DX(S1), B0XC(S1)
COMMON/ INPUT / TX(2, 50)
COMMON/ POINTP/ TY(4), TYP(4), TYP(4)
COMMON/ POINTS/ MPTS, NVAR
C INENSION XB(S1), XB( S1), Z0( S1), THETA( 11)
C
MPTS = NP
XLAST = XB(NP)
C
C SPLINE FIT INPUT PROFILE AFTER COORDINATE TRANSFORM.
C
ICAMBR = 0
DO 100 I=1, MPTS
TX(1, I) = XB(I)
TX(2, I) = -B(I)
IF ( Z0(I) ) -1.1E-10 ) 60 TO 100
ICAMBR = 1
TX(3, I) = Z0(I)
100 CONTINUE
C
NVAR = 3
IF ( ICAMBR .EQ. 0 ) NVAR = 2
CALL SPLFIT
IF ( IMACHN .GE. 2.50 ) 60 TO 260
C
C FOR IMPROVED ACCURACY, FIRST 20 PERCENT OF BODY IS DIVIDED INTO
C 10 EQUAL SECTIONS, THE REMAINING PART IS DIVIDED INTO 15 EQUAL
C SECTIONS.
NT = 11
XMID = XB(NP)
NBO1 = NBOOTS - 1
DX = XMID * 0.2 / 10.0
DO 200 I=2, NT
AI = I - 1
TY(1) = DX * AI
XB(I) = TY(1)
CALL POINT(1)
RB(I) = TY(2)
IF ( ICAMBR .EQ. 0 ) 60 TO 200
Z0(I) = TY(3)
200 CONTINUE
C
NX1 = NT - 1
N2 = NT + 1
DX = ( XLAST - XB(NT) ) / ( NBO1 - NX1 )
DO 250 I=N2, NBOOTS
XB(I) = XB(I-1) + DX
TY(1) = XB(I)
CALL POINT(1)
RB(I) = TY(2)

```

```

IF( ICAHGR .EQ. 0 ) GO TO 250
I2(1) = TV(3)
250 CONTINUE
GO TO 274

C
C
C      FOR HIGH SUPERSONIC CASE, XB, RE EQUALLY DIVIDED
C
260 CONTINUE
DX = ELAST/(NBODYS-1)
DO 265 I = 2, NBODYS
  XB(I) = (I-1)*DX
  TV(I) = XB(I)
  CALL POINT(I)
  RB(I) = TV(2)
  ID(I) = TV(3)
265 CONTINUE
274 CONTINUE

C
C
C      CALCULATE THE SLOPE AND CARRIER OF BODY AT POINT MIDWAY BETWEEN
C      STATION.
C
DO 275 I=2,NBODYS
  TV(I) = ( XB(I-1) + XB(I) ) * 0.5
  CALL POINT(I)
  DBDX(I-1) = TV(2) / TV(1)
  IF ( ICAHGR .EQ. 0 ) GO TO 275
  DBDX(I-1) = TV(3) / TV(1)
275 CONTINUE
  RES1 = NBODYS - 1
  DBDX(NBODYS) = (RB(NBODYS) - RB(NBDS)) / ( XB(NBODYS) - XB(NBDS) )

C
  TV(1) = XB(1)
  CALL POINT(1)
  DBS1 = TV(2) / TV(1)
  TV(1) = RB(51)
  CALL POINT(1)
  DBS2 = TV(2) / TV(1)

C
C      GENERATES THETA FOR BODY PANEL
C
  DTN = 180.0 / FLOAT(DTHET - 1)

C
DO 300 I=1,DTHET
  AI = I - 1
  THETA(I) = DTN * AI
300 CONTINUE

C
  XB51 = 1.0
  XB52 = NBODYS
  RETURN
END

C
C/ LIST NONE
C/ SUBROUTINE DEFAUT
C
COMMON/DEFAUT / BODY, BODYS, DTHET, CPALC,WCASE, MRUN, PADES, ARB,
1 PLANE, POLAR, RP, INACEL, SYN, THICK, WING, INACH,
2 XMB, INC, PINF, IPUNCH, SID(20), CHASE, ARB(10), IMON,
3 TAN
C
  BANK = 0.0
  BODY = 0.0
  BODYS = 25.0

```

```

BTNET = 7.0
CPCALC = 1.0
DAEG = 0.0
BIM = 0.0
NRUN = 1.
IPUNCH = 0
CKASE = 2
PIW = 14.7
PIVOT = 0.0
POLAR = 0.0
RFAREA = 0.0

```

```

C
NP = 0.0
DO 100 I=1,20
  SIO(I) = 1.0
100 CONTINUE
  SYN = 1.0
  THICK = 1.0
  THCOMP = 1.0
  TWIST = 0.0
  WING = 0.0
  WCASE = 3
  WTRICH = 2.0
  XNACEL = 0.0
  XNB = 0.0
  XNC = 9.0
  XNB = 3.0
  XBC = 4.0
  XP = 0.0
  ZA = 0.0
  ZP = 0.0
  YAW = 0.0
  XNOM = 0.0

```

```

RETURN
END

```

```

SUBROUTINE CORDTR

```

```

      REAL IN COORDINATE SPECIFICATIONS VECTORS VA, VB, AND VC
      AND COMPUTES THE TRANSFORMATION MATRICES BTM FOR EACH
      COORDINATE SYSTEM

```

```

COMMON/CORTAN/ NSTN,NTRANS(20),ISTN(20),VA(3,2,20), BTM(9,2,20)

```

```

DIMENSION ICOORD(3), VB( 3, 3), VCC( 3, 3)

```

```

DATA ICOORD/1NR, 1NC, 1NS /

```

```

      COMPUTE COORDINATE SYSTEM TRANSFORM MATRICES

```

```

      NSTN = NTRANS(NSTN)
      DO 100 ITRAN = 1, NSTN

```

```

        READ (5,901) ID, ICHECK,NM, (VA(I,ITRAN,NSTN), I=1,3), (VB(I,ITRAN)
        1      I=1,3)
        READ (5,902) (VC( I, ITRAN), I=1,3)
        IF ( ITRAN .GT. 1 ) GO TO 50
        IF ( ICOORD(1) .EQ. ICHECK ) ISTM( NSTN ) = 0
        IF ( ICOORD(2) .EQ. ICHECK ) ISTM( NSTN ) = 1
        IF ( ICOORD(3) .EQ. ICHECK ) ISTM( NSTN ) = 2

```

```

C 50 CONTINUE

```



```

C      RETURN
C      END
C      SUBROUTINE BANVIN
C
COMMON/MAIN / LA, LB, LC, LD, LE, LF, LI, LO, MBOOY, MWING, XNACHN,
1 SYM, KACE, MPOLAR, IRM
COMMON/ BODY2 / MBPFL, MBOOYS, MTHET, XB(S1), RB(S1), ZD(S1),
1 THETA(11), RPP, XBPANL(20)
COMMON / BODYSP/ BRDI(S1), DZDI(S1), BRDZC(S1)
COMMON/ MACEL / MACELP, XN(S1,2), RN(S1,2), ZN(S1,2), ARN(2)
COMMON/COTAN/ MSTR, MTRANS(20), ISTMI(20), VA(3,2,20), BTJA(9,2,20)
COMMON/DEFAU / BODY, BODYS, BTMET, CPALC, MCASE, MRUM, BADEG, ARB,
1 PLANE, POLAR, RP, XNACEL, SYN, THICK, WING, XNACH,
2 XNB, XNC, PINF, IPUN-N, SID(20), CRASE, ARN(10), XNOM,
3 TAN
COMMON/ANGLE/ ALPHAC(55,10), ALPHAT(55,10), ARUT(55,10)
COMMON/WINGFM/ XN(30,20), TMI(30,20), ZMI(30,20), MNP(20), MCHORD,
1 MSPAN, ISOLID, MPLANE, ICARDR, ITWIST, ITWICK, IPORR(10)
2 PIVOT(10), DIN(10), MNPANL, XLG(10), RTG(10), BROLL
COMMON/INPUT / IX(2, 50)
COMMON/POINTP/TT(4), TYP(4), TYP(4)
COMMON/POINTS/ NPTS, NVAR
1 DIMENSION RBPANL(11), ZBPANL(11), X(40), XCR(40), YC(40),
2 AREA(40)
C
NBOOY = 1
MBOOYS = IFIX( BODYS )
MTHET = IFIX( BTMET )
C
CALL BODYPF(XB, RD, ZD, THETA, MBPFL, MTHET, MBOOYS, XOS1, XOS2,
1 DSB1, DSB2, XLAST)
C
CALL BODYPN(XLAST)
C
RESET NC AND NB AFTER MODIFICATION IN BODYPN
C
PLANE = FLOAT( MPLANE )
NC = XNC
NB = XNB
MC1 = NC - 1
MB1 = NB - 1
MBC = MB1-MC1
MBC2 = MBC + 1
MCOLW = MWING * MB1
MW2 = 2 * MWING
TRKOMP = 1.0
MWING = FLOAT(MWING)
SOLID = FLOAT(ISOLID)
PUNCH = FLOAT(IPUNCH)
C
REWINO 11
WRITE (11,1020) BODYS, BTMET, PLANE, RP, TAN
WRITE (11,1020) ( X(N), N=1, MBOOYS )
WRITE (11,1020) ( RE(N), N=1, MBOOYS )
WRITE (11,1020) ( ZD(N), N=1, MBOOYS )
WRITE (11,1020) ( THETA(N), N=1, MTHET )
WRITE (11,1020) XOS1, XOS2, OS1, OS2
WRITE (11,1020) ( ZBPANL(N), N=1, MPLANE )
IF ( RP .GT. 0.0 ) GO TO 410
WRITE (11,1020) ( RBPANL(N), N=1, MTHET )
WRITE (11,1020) ( ZBPANL(N), N=1, MTHET )

```

```

410 CONTINUE
  PERCT = 0.1
  DO 450 I = 1, MW2, 2
    MW1 = 1/2 * I
    J = 1
    IF ( IFORM(MW1) .EQ. 2 ) J = MW1
    WRITE (11,1020) XW(1,J),TWI(1,J),XW(2,J),TWI(6,J),TWI(6,J),
      TWI(3,J)
    WRITE (11,1020) XWB, INC, INC, WCASE, TWI(3,J)
    WRITE (11,1020) PIVOT(MW1), DIM(MW1), BROLL
    CALL WINGS( MW1, XC, XCR, YC, AREA)
450 CONTINUE
    TWIST = FLOAT( ITWIST )
    THICK = FLOAT( ITHICK )
    CAMB = FLOAT( ICAMB )
    WRITE (11,1020) XNACH, STM, THICK, YAW, WING, BROLL
C
    WRITE (11,1020) CBASE, CPCALC, POLAR, THICK, CAMB, XNACEL, PINF,
      PUNCH, SOLID, WING, ( SID(R), K=1,20 ), ( IFORM(K), K=1,10 ), XROM
C
    IF ( XNACEL .EQ. 0 ) GO TO 460
    CALL XNACEL( XNACEL, MWING )
460 CONTINUE
    WRITE (11,1020) ARB, DADEG
    WRITE (11,1020) ( ARW(I), I=1,10 ), TWIST, MWING
    IF ( ITWIST .EQ. 1 ) WRITE (11,1020) ( ARNT(J,1), J=1,MW1 ), I=1,
      MWING )
    IF ( ICAMB .EQ. 1 ) WRITE (11,1020) ( ALPHAC(I,J), I=1,MBC ),
      J=1,MWING )
    IF( ITHICK .NE. 1 ) GO TO 480
    DO 470 L = 1, MWING
    DO 470 J = 1, MW1
    K1 = 1 + (J-1)*MC
    KK = K1 + MC1
    WRITE (11,1020) ( ALPHAT(L,1), R=K1,KK )
470 CONTINUE
C
    XWTP = XW(4, 1)
C
480 CONTINUE
    REMIND 11
C
1020 FORMAT (7F10.4)
C
    RETURN
    END
C/
    LIST, MORE
C/
    SUBROUTINE WINGIN
C
    COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI, LO, MBOOT, MWING, XNACH,
      1 STM, XACE, MPOLAR, IEW
    1 COMMON/DEFAU / BODT, BODTS, BTMET, CPCALC, CASE, XROM, DADEG, ARB,
      1 PLANE, POLAR, RP, XNACEL, STM, THICK, WING, XNACH,
      2 XWB, INC, PINF, IPUNCH, SID(20), CBASE, ARW(10), XROM
    COMMON/WANGLE/ ALPHAC(55,10), ALPHAT(55,10), ARNT(55,10),
    COMMON/WINGFM/ XW(30,20), TWI(30,20), XW(30,20), MWPI(20), XCHORD,
      1 XSPAN, ISOLID, MPLANE, ICAMB, ITWIST, ITHICK, IFORM(10)
      2 , PIVOT(10), DIM(10), XNPARL, ILG(10), ITC(10), BROLL
C
    DIMENSION XC(40), XCR(40), YC(40), AREA(40)
C
    NBOODY = 0

```

```

C
THKOMP = 1.0
NPLANE = IFIX( PLANE )
PUNCH = FLOAT( IPUNCH )
SOLID = FLOAT( ISOLID )
WING = FLOAT( MWING )

C
NC = XNC
NB = XNB
NC1 = NC - 1
NB1 = NB - 1
NBC = NB1 - NC1
PACT = NBC + 1
COLW = MWING * NB1

C
WRITE ON TAPE 11 FOR AERO CODE
C
C
REMI1D 11
C
NW2 = 2 * MWING
DO 50 I=1, NW2, 2
J = 1
NW1 = I / 2 + 1
IF ( IFORM(NW1) .EQ. 2 ) J = NW1
WRITE ( 11,1020) NW1(1,J), TWI(1,J), TWI(4,J), TWI(6,J),
1 NW1(3,J)
WRITE (11,1020) XNB, XNC, XCASE, ZWI(1,J)
WRITE (11,1020) PIVOT(NW1), DINC(NW1), BROLL
50 CONTINUE
C
READ IN WING PLAN FORM AND COMPUTE WING CAMBER, TWIST, AND
THICKNESS SLOPE.
C
DO 100 I=1, MWING
CALL WINGS(I, XC, XCR, YC, AREA)
100 CONTINUE
THICK = FLOAT( ITHICK )
TWIST = FLOAT( ITWIST )
CAMB = FLOAT( ICAMBR )
WRITE (11,1020) XHACH, SYN, THICK, YAW, WING, BROLL
C
WRITE (11,1020) XCASE, CPALC, POLAR, THICK, CAMB, XHACH, PINF,
1 PUNCH, SOLID, WING, ( SIOCK), K=1,20), ( IFORM(K), K=1,10), XRON
WRITE (11,1020) ARB, BADGE
WRITE (11,1020) ( ARW(I), I=1,10), TWIST, MWING
C
IF ( ITWIST .EQ. 1 ) WRITE (11,1020) ( ARW(J,I), J=1, NB1), I=1,
1 MWING
IF ( ICAMBR .EQ. 1 ) WRITE (11,1020) ( ALPHAC(I,J), I=1, NBC),
1 J=1, MWING
IF( ITHICK .NE. 1 ) GO TO 480
DO 470 L = 1, MWING
DO 470 J = 1, NB1
K1 = 1 + (J-1)*NC
KK = K1 + NC1
WRITE (11,1020) ( ALPHAT(I,L), I=K1, KK)
470 CONTINUE
480 CONTINUE
C
REMI1D 11
C
1020 FORMAT ( 7F10.4)
C
C
RETURN
END

```



```

ALPHAT(IJ),I) = CF(IJ)
515 CONTINUE
520 CONTINUE
C
ITHICK = 1
50 TO 610
600 CONTINUE
ITHICK = 0
610 CONTINUE
C
RETURN
END
C
SUBROUTINE INTERP( M, I, Y, CF, MP, XP, YP, ZP )
C
C      INTERPOLATES AT NP POINTS ( XP, YP ) FOR DERIVATIVE
C      D/DX
C
C      DIMENSION X(1), Y(1), XP(1), YP(1), ZP(1), CF(1)
C
C      DO 100 I=1,MP
C      SUM = CF(2)
C      DO 110 J=1,M
C      JS = J + 3
C      RI = (XP(I) - X(J))**2 + (YP(I) - Y(J))**2
C      IF ( RI .LT. 1.E-9 ) 60 TO 110
C      SUM = SUM + 2.0*CF(JS)*(1.0*ALOG(RI))*(XP(I)-X(J))
C      ZP(I) = SUM
C      110 CONTINUE
C      100 CONTINUE
C
C      RETURN
C      END
C
SUBROUTINE WPADEC( XI, YI, XC, MB, XC, XCR, YC, AREA)
C
C      COMPUTE CONTROL POINT ( XC, YC ) OF WING PANELS
C
C      DIMENSION XI(1), YI(1), XC(1), YC(1), XCR(1), YCR(1), XC(1), YC(1),
C      1  XY(1), XCR(40), AREA(40)
C
C      PERCT = 0.1
C      XCPT = 0.95
C      XN(1) = XI(1)
C      XCHORD = XI(2) - XI(1)
C      XN(2) = XN(1) + PERCT*XCHORD
C      DX = (1.0 - PERCT) * XCHORD / ( MC - 2 )
C      DO 100 L=3,MC
C      XN(L) = XN(L-1) + DX
C      100 CONTINUE
C      BY = YI(4) - YI(1)
C      DBY = BY / ( MB - 1 )
C      YN(1) = YI(1)
C      DO 165 M=2,MB
C      YN(M) = YN(M-1) + DBY
C      165 CONTINUE
C      SLE = ( XI(4)-XI(1) ) / BY
C      STE = ( XI(3)-XI(2) ) / BY
C      IF ( SLE-STE ) 210, 250, 210
C      210 CONTINUE
C      DX = ( XI(3) - XI(4) ) / ( XI(2) - XI(1) )
C      DO 215 M=1,MC
C      XY(M, MB) = XI(4) + DX + ( XN(M) - XI(1) )

```



```

C
      NBI = NSPAN - 1
      NC1 = NCHORD - 1
      DO 200 K=1,NBI
        DT = 1 + ( K-1 ) * NCHORD
        DY = YC( K1 ) - YN1( 1, 1 )
        XLE = XN1(1,1) + VX(1) * DY / VY(1)
        XTE = XN1(2,1) + VX(2) * DY / VY(2)
        ZLE = ZN1(1,1) + VZ(1) * DY / VY(1)
        ZTE = ZN1(2,1) + VZ(2) * DY / VY(2)
        ARWTK(1) = ( ZLE-ZTE ) / ( XTE-XLE )
      200 CONTINUE
C
      RETURN
      END
      SUBROUTINE WTRANS( NW, IW )
C
C      TRANSFER COORDINATES DEFINING WING FORM INTO THAT OF AERO CODE
C
      COMMON/NSGFM/ XN1(30,20), YN1(30,20), MNPI(20), NCHORD,
1      NSPAN, ISOL10, NPLANE, ICAMER, ITWIST, ITALICK, IFORM(10)
2      PIVOT(10), SIN(10), NBPANL, XLG(10), XTG(10)
      COMMON/CORTN/ MSYM, NTRANS(20), ISTN(20), VALS(2,20), BTFR(9,2,20)
C
      DR = 0.0174532925
      NUP = MNPI( NW )
      IF ( IFORM(IW) ) .EQ. 2 ) NUP = 4
      DO 400 J=1,NUP
        R = XN1( J, NW )
        THETA = YN1( J, NW ) * DR
        PHI = ZN1( J, NW ) * DR
        N60 = ISTN( MSYM ) * 1
        GO TO ( 300, 100, 200 ) , N60
      100 CONTINUE
        XN1( J, NW ) = R * COS( THETA )
        YN1( J, NW ) = R * SIN( THETA )
        ZN1( J, NW ) = R * SIN( THETA )
        GO TO 300
C
      200 CONTINUE
        XN1( J, NW ) = R * SIN( THETA ) * COS( PHI )
        YN1( J, NW ) = R * SIN( THETA ) * SIN( PHI )
        ZN1( J, NW ) = R * COS( THETA )
C
      300 CONTINUE
C
      400 CONTINUE
      RETURN
      END
      SUBROUTINE MACEL( XMACEL, NWING )
C
C      INTERPOLATES INPUT OF MACEL CONFIGURATION AND WRITES ON TAPE 11
C      FOR AERO CODE.
C
      COMMON/ BODY2 / MBPFL, MBODTS, MTHET, XB(51), XB(51), ZB(51),
1      THETA(11), RP, XBPANL(20)
      COMMON/ BODYSP/ DRDX(51), DRDX(51), DRDX(51)
      COMMON/ MACEL /MACELP, XN(51,2), RN(51,2), ZN(51,2), ARN(2)
      COMMON/CORTN/ MSYM, NTRANS(20), ISTN(20), VALS(2,20), BTFR(9,2,20)
C
      ANCEL = ABS( XMACEL )

```



```

DO 300 I = 1, 3
  I1 = M1(I)
  I2 = M2(I)
  UNIVC(2,I) = UNIVC(1,I) + UNIVC(3,I2) - UNIVC(12) + UNIVC(3,I1)
  SUM = SUM + UNIVC(2,I) ** 2
300 CONTINUE
SUM = SORT(SUM)
DO 350 I = 1, 3
  UNIVC(2,I) = UNIVC(2,I) / SUM
350 CONTINUE
C
C      UNIT VECTOR 1
C
SUM = 0.0
DO 400 I = 1, 3
  I1 = M1(I)
  I2 = M2(I)
  UNIVC(1,I) = UNIVC(2,I2) + UNIVC(3,I1) - UNIVC(2,I1)
  SUM = SUM + UNIVC(1,I) ** 2
400 CONTINUE
SUM = SORT(SUM)
DO 450 I = 1, 3
  UNIVC(1,I) = UNIVC(1,I) / SUM
450 CONTINUE
C
C      RETURN
C      END
C
SUBROUTINE CUBERT
  DIMENSION X(3)
  COMMON/LOCN/ V(3), L, M, G(4), VMAX, VMIN
  COMMON/POINTS/ NPTS, NVAR
  L=0
  SUM = SORT( 6(1)**2 + 6(2)**2 + 6(3)**2 + 6(4)**2 )
  G1 = 6(1) / SUM
  G2 = 6(2) / SUM
  G3 = 6(3) / SUM
  G4 = 6(4) / SUM
C
C      EQ. 1  61*X**3 + 62*X**2 + 63*X = 0.
C      IF ( ABS( G4 ) .LT. 1.E-6 ) GO TO 4
C
C      EQ. 2  61*X**3 + 62*X**2 + 63*X + G4 = 0.
C      IF ( ABS( G1 ) .GT. 1.E-6 ) GO TO 9
C
C      EQ. 3  62*X**2 + 63*X + G4 = 0.
C      IF ( ABS( G2 ) .GT. 1.E-6 ) GO TO 7
C
C      EQ. 4  63*X + G4 = 0.
C      IF ( ABS( G3 ) .GT. 1.E-6 ) GO TO 5
C      RETURN
C
C      CONTINUE
V(1) = 0.0
L=1
RETURN
C
C      CONTINUE
V(1) = - G(4) / 6(3)
GO TO 35
C
C      CONTINUE
R1 = 6(3)**2 - 4.0*6(2)*6(4)
IF ( R1 .GT. 0.0 ) GO TO 107
WRITE (6,901)
901 FORMAT (1H0,'NEGATIVE RADICAL RESULTED FROM SOLVING THE QUADRATIC

```



```

V0 = A2 + A3 - A1*V1
35 CONTINUE
IF ( VO .GT. VMIN .AND. VO .LT. VMAX ) L = 1
IF ( L .EQ. 0 ) WRITE (6,905)
905 FORMAT (1H0, 'THE ONLY UNEQUAL ROOT IS OUTSIDE OF VMIN, VMAX/1X,
1 VMIN =',E14.4, 'V =',E14.4, 'VMAX =',E14.4)
V(1) = VO
RETURN
40 CONTINUE
DO 45 I=1,3
VO = X(1) - A1/3.0
X(1) = VO
V(1) = VO
IF ( VO .LT. VMIN .OR. VO .GT. VMAX ) GO TO 50
L=1
RETURN
50 CONTINUE
45 CONTINUE
WRITE (6,904) VMIN, V(1), V(2), V(3), VMAX
904 FORMAT (1H0, 'ALL THREE UNEQUAL ROOTS OUTSIDE OF LIMIT VMIN, VMAX/
1 1X, VMIN =',E14.4, 'X1 =',E14.4, 'X2 =',E14.4, 'X3 =',
2 E14.4, 'VMAX =',E14.4)
C
RETURN
END
C
SUBROUTINE POINT( M )
COMMON/CURFIT/ B(2,4, 50), T( 50), SCALEF(2)
COMMON/ENDPT/ XFIRST, XLAST, XSAVE( 50)
COMMON/LOCN/ V(3), L, M, A(4), VMAX, VMIN
COMMON/POINTS/ NPTS, NVAR
COMMON/POINTP/ X(4), XP( 4), XPP( 4)
DIMENSION B(4)
M1 = NPTS - 1
IF (N.NE.0) GO TO 4
DO 1 I=1,M1
M=I
IF ( V(1) .LE. T(I+1) ) GO TO 2
1 CONTINUE
WRITE(6,100)
100 FORMAT(1H0, '3M ERROR - PARAMETER EXCEEDS LIMITS)
RETURN
2 CONTINUE
DO 300 K=1,3
P = V(K)
DX(K) = 1.E10
IF ( P .LT. VMIN .OR. P .GT. VMAX ) GO TO 300
DX(K) = ABS(((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M))*P + B(1,4,M))
1 SCALEF(1) = XIND
300 CONTINUE
IF ( DX(1) .GT. DX(2) ) GO TO 302
IDENT = 1
IF ( DX(1) .GT. DX(3) ) IDENT = 3
GO TO 305
302 CONTINUE
IDENT = 2
IF ( DX(2) .GT. DX(3) ) IDENT = 3
305 CONTINUE
P = V(IDENT)
DO 3 I=1, NVAR
X(I) = ((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M))*P + B(1,4,M)) * SCALEF(1)
XP(I) = ((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M)) * SCALEF(1)
3 XPP(I) = ((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M)) * SCALEF(1)

```

```

C 310 CONTINUE
C RETURN
C
4 X(N)=X(N)/SCALEF(N)
DO 5 I=1,M
M=1
K = 1 + 1
IF ( XSAVE(K) -GE. X(N) ) 60 TO 6
5 CONTINUE
XIND = X(N) * SCALEF(N)
XFT = FIRST * SCALEF(1)
XLT = XLAST * SCALEF(1)
WRITE (6,101) XIND, XFT, XLT
101 FORMAT ('20.0 THE GIVEN VALUE OF INDEPENDENT VARIABLE *E12.4,
1
5X, *15 OUT OF RANGE TO BE INTERPOLATED *E12.4,5X,E12.6)
RETURN
6 DO 7 I=1,4
7 A(1)=B(N,I,M)
A(4)=A(4)-X(N)
IF ( XFIRST -EQ. X(N) ) 60 TO 112
IF ( XLAST -EQ. X(1) ) 60 TO 111
VMAX = T(M-1) + 1.0001
VMIN = T(M) + 0.9999
CALL CUBRT
XIND = X(1) * SCALEF(1)
60 TO 103
112 V(1) = 0.0
L=1
60 TO 103
111 V(1) = 1.0
L=1
103 CONTINUE
IF (L.NE.0) 60 TO 2
OP = P
WRITE (6,10) OP
10 FORMAT ('1XNOP =,E15.8)
DO 8 J=1,2
K = M + J - 1
B1 = B(N,1,K)
B2 = B(N,2,K)
B3 = B(N,3,K)
B4 = B(N,4,K)
P = T(K)
SUM = B1*P**3 + B2*P**2 + B3*P + B4
DIF = SUM - X(N)
8 WRITE (6,9) K,P,B1,B2,B3,B4,X(N),SUM,DIF
9 FORMAT ('1X13,E15.8)
WRITE (6,102)
102 FORMAT('1X0,32H FAILURE IN CUBE ROOT EXTRACTION)
RETURN
END
C
SUBROUTINE SPLFIT
REAL L,M,MU
COMMON/CURFIT/ B(2,4,50), T( 50), SCALEF(2)
COMMON/ENDFIT/ XFIRST, XLAST, XSAVE( 50)
COMMON/INPUT / X( 2, 50)
COMMON/POINTS/ NPTS, NVAR
COMMON/POINTS/ M(2, 50), S( 50), L( 50), MU( 50), B( 50), P( 50), B( 50)
1
), U( 50)
IF ( NPTS -GT. 50) NPTS= 50
DO 200 I=1, NVAR
200 SCALEF(I)=0.
DO 201 I=1,NPTS

```

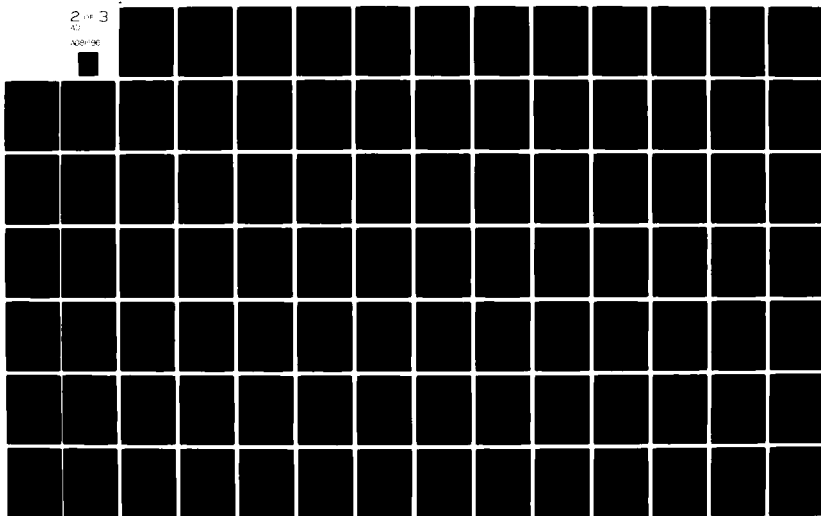
AD-A081 596

MCDONNELL DOUGLAS ASTRONAUTICS CO HUNTINGTON BEACH CA F/G 9/2
AERODYNAMIC COMPUTER CODE FOR COMPUTING PRESSURE LOADS ON A MIS--ETC(U)
DEC 79 K K WANG N00014-75-C-0462
MDC-68223 NL

UNCLASSIFIED

2 of 3
AD

NOV-90



```

DO 201 J=1,NVAR
201 SCALEF(J)=MAX1(SCALEF(J),ABS(X(I,J,I)))
DO 101 J=1,NVAR
IF ( SCALEF(J) .LT. 1.E-10 ) SCALEF(J) = 1.0
101 CONTINUE
DO 202 I=1,NPTS
DO 202 J=1,NVAR
202 X(I,J)=X(I,J)/SCALEF(J)
DO 203 I=1,NPTS
XSAVE(I) = X(I,I)
203 CONTINUE
XFIRST = X(1,1)
XLAST = X(1,NPTS)
T(I)=0.
S(I)=0.
SUM=0.
L(I)=2.
MU(I)=0.
L(NPTS)=0.
MU(NPTS)=2.
D(I)=0.
D(NPTS)=0.
P(I)=2.
Q(I)=1.
U(I)=0.
M1=NPTS-1
DO 2 I=2,NPTS
SUM1=0.
DO 1 J=1,NVAR
1 SUM1=SUM1+(X(I,J)-X(I,J-1))**2
2 T(I)=T(I-1)+S(I)
DO 3 I=2,NPTS
S(I)=S(I)/T(NPTS)
3 T(I)=T(I)/T(NPTS)
DO 4 I=2,M1
L(I)=S(I-1)/(S(I)+S(I+1))
MU(I)=1.-L(I)
P(I)=MU(I)-Q(I-1)+2.
4 Q(I)=-L(I)/P(I)
P(NPTS)=2.-Q(1)-Q(M1)
DO 7 I=1,NVAR
DO 5 J=2,M1
D(J)=6.-((X(I,J-1)-X(I,J))/S(J-1))-((X(I,J)-X(I,J-1))/S(J))/S(J)+
1S(J+1))
5 U(J)=((D(J)-MU(J)-U(J-1))/P(J)
U(NPTS)=-MU(NPTS)+U(M1)/P(NPTS)
M1,NPTS)=U(NPTS)
DO 6 J=1,M1
K=NPTS-J
M(I,K)=Q(K)*M(I,K+1)+U(K)
B(I,1,K)=M(I,K-1)-M(I,K)/(6.-S(K+1))
B(I,2,K)=5*(M(I,K)-T(K-1))-M(I,K-1)*T(K)/S(K+1)
B(I,3,K)=(-5*(M(I,K-1)*T(K))+2*M(I,K)-T(K-1)*2)*X(I,K-1)-B(I,K)+
1M(I,K)-M(I,K+1)+S(K+1)+2/6./S(K+1)
6 B(I,4,K)=M(I,K)*T(K-1)+3*M(I,K-1)*T(K)+3*(K-1)*X(I,K)-T(K-1)
1K-1)-X(I,K-1)*T(K)/S(K+1)+S(K+1)/6.*(M(I,K-1)*T(K)-M(I,K)*T(K+1)
1)
7 CONTINUE
RETURN
END

```

SUBROUTINE SURFIT(X, Y, N, B, TH)

C C C

```

C      COMPUTES COEFFICIENTS FOR SURFACE SPLINE.
C      N TOTAL NUMBER OF INPUT POINTS, CAN BE RANDOMLY LOCATED.
C      JEQ FOR EQUATION NUMBERING
C      MUK FOR COEFFICIENT NUMBERING
C
C      CF ARE THE COEFFICIENT MATRIX OF EQUATION
C       $W(1,Y) = B(1) + B(2) \cdot X + B(3) \cdot Y + \text{SUM OF } B(I) \cdot R(I) \cdot 2 \cdot$ 
C       $LN(R(1) \cdot 2)$ 
C
C      COMMON/REPEAT/ IREPT
C      DIMENSION IPS(115), CF(115,115), TH(1), SCALES(115)
C      DIMENSION X( 1), Y( 1), B( 1)
C
C      M4 = 115
C      M1 = M + 1
C      M2 = M + 2
C      M3 = M + 3
C
C      BYPASS MATRIX INVERSION FOR THE CASE WHERE THE DATA POINTS (X,
C      Y) ARE IDENTICAL FOR INTERPOLATION
C
C      IF ( IREPT .GT. 1 ) GO TO 300
C      DO 100 JEQ = 1, M
C      CF( JEQ, 1 ) = 1.0
C      CF( JEQ, 2 ) = X( JEQ )
C      CF( JEQ, 3 ) = Y( JEQ )
C      DO 90 MUK = 1, M
C      MUK1 = MUK + 3
C      IF ( MUK .EQ. JEQ ) GO TO 85
C      RJM = ( X( MUK ) - X( JEQ ) ) ** 2 + ( Y( MUK ) - Y( JEQ ) ) ** 2
C      CF( JEQ, MUK1 ) = RJM * ALOG( RJM )
C      GO TO 90
C      85 CF( JEQ, MUK1 ) = 0.0
C      90 CONTINUE
C      100 CONTINUE
C
C      DO 150 K=1,3
C      CF( M1, K ) = 0.0
C      CF( M2, K ) = 0.0
C      CF( M3, K ) = 0.0
C      150 CONTINUE
C      DO 200 MUK = 1, M
C      MUK1 = MUK + 3
C      CF( M1, MUK1 ) = 1.0
C      CF( M2, MUK1 ) = X( MUK )
C      CF( M3, MUK1 ) = Y( MUK )
C      200 CONTINUE
C
C      250 CONTINUE
C      CALL DECOMP( M3, CF, IPS, SCALES )
C
C      300 CONTINUE
C      B(M1) = B(M2) = B(M3) = 0.0
C      CALL SOLVE( M3, CF, B, TH, IPS )
C
C      RETURN
C      END
C      SUBROUTINE DECOMP( MM, A, IPS, SCALES )

```

```

C      DECOMPOSING OF MATRIX A
C
C      DIMENSION A(115, 115), SCALES( 1), IPS( 1)
C
C      N = NN
C
C      INITIALIZE IPS, A AND SCALES
C
C      DO 5 I = 1, N
C      IPS(I) = 1
C      ROWSUM = 0.0
C      DO 2 J = 1, N
C      IF ( ROWSUM - ABS( A( I, J) ) ) 1, 2, 2
C      1 ROWSUM = ABS( A( I, J) )
C      2 CONTINUE
C      IF ( ROWSUM ) 3, 4, 3
C      3 SCALES(I) = 1.0 / ROWSUM
C      GO TO 5
C      4 CALL SING(1)
C      SCALES(I) = 0.0
C      5 CONTINUE
C
C      GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
C
C      NN1 = N - 1
C      DO 17 K = 1, NN1
C      BIG = 0.0
C      DO 11 I = K, N
C      IP = IPS(I)
C      SIZE = ABS( A( IP, K) ) * SCALES(IP)
C      IF ( SIZE - BIG ) 11, 11, 10
C      10 BIG = SIZE
C      IDXPIV = I
C      11 CONTINUE
C      IF ( BIG ) 12, 12, 13
C      12 CALL SING(2)
C      GO TO 17
C      13 IF ( IDXPIV - K ) 14, 15, 14
C      14 J = IPS(K)
C      IPS(K) = IPS( IDXPIV )
C      IPS(IDXPIV) = J
C      15 KP = IPS(K)
C      PIVOT = A( KP, K)
C      KP1 = K + 1
C      DO 16 I = KP1, N
C      IP = IPS(I)
C      EM = - A( IP, K) / PIVOT
C      A( IP, K) = - EM
C      DO 16 J = KP1, N
C      A( IP, J) = A( IP, J) + EM * A( KP, J )
C
C      INNER LOOP
C
C      16 CONTINUE
C      17 CONTINUE
C      KP = IPS(N)
C      IF ( A( KP, N ) ) 19, 18, 19
C      18 CALL SING(2)
C      19 CONTINUE
C
C      RETURN
C      END
C      SUBROUTINE SOLVE( NN, A, B, X, IPS)

```

```

C      SOLVE FOR MATRIX A * X = B
C
C      DIMENSION A(115, 115), B( 1), X( 1), IPS(1)
C
C      N = NN
C      NP1 = N + 1
C
C      IP = IPS(1)
C      X(1) = B(IP)
C      DO 2 I=2,N
C      IP = IPS(I)
C      IM1 = I - 1
C      SUM = 0.0
C      DO 1 J = 1, IM1
C      SUM = SUM + A(IP, J) * X(J)
C      X(I) = B(IP) - SUM
C
C      IP = IPS(N)
C      X(N) = X(N) / A( IP, N)
C      DO 4 IBACK = 2, N
C
C      I GOES FROM (N-1) . . . . . 1
C
C      I = NP1 - IBACK
C      IP = IPS(I)
C      IP1 = I + 1
C      SUM = 0.0
C      DO 3 J = IP1, N
C      SUM = SUM + A( IP, J) * X(J)
C      X(I) = ( X(I) - SUM ) / A( IP, I)
C
C      RETURN
C      END
C      SUBROUTINE SING(INNY)
C
C      PRINTS STATEMENT
C
C      MOUT = STANDARD OUTPUT UNIT
C
C      MOUT = 6
C      GO TO ( 1, 2, 3), INNY
C      1 WRITE ( MOUT, 11)
C      GO TO 10
C      2 WRITE ( MOUT, 12)
C      GO TO 10
C      3 WRITE ( MOUT, 13)
C
C      11 FORMAT (1M0,'MATRIX WITH ZERO ROW IN DECOMPOSE')
C      12 FORMAT (1M0,'SINGULAR MATRIX IN DECOMPOSE, ZERO DIVIDE IN SOLVE')
C      13 FORMAT (1M0,'NO CONVERGENCE IN IMPRV. MATRIX NEARLY SINGULAR')
C
C      10 CONTINUE
C
C      RETURN
C      END
C      OVERLAY(WANG, 2, 1)
C      PROGRAM HIGHSP
C
C      COMMON/MAIN /LA, LB, LC, LD, LE, LF, LL, LO, MRODT, MZING, IMACHN,
C      1 SYMM, KACE, MPOLAR, INV
C      COMMON/ 00512 / MDPFL, MRODT3, MYNET, X0(51), Z0(51),
C      1 THETA(11), RPP, XDPARL(20)

```

```

COMMON / BODYTYP / DRX(51), DZD(51), DRDX(51), HURDUT,INTIP
COMMON/CONTR/ MSTR,MTRANS(20),ISTN(20),VA(3,2,20),BTFM(9,2,20)
COMMON/DEFA/ BODY, BODYT, BTHET, CPALC,MCASE, MRUN, MAGE, ARB,
1 PLANE, POLAR, RP, KNADEL, SYM, THICK, WING, IMACH,
2 XNB, INC, PINF, IPUNCH, SLD(20), CKASE, ARK(10), XHON,
3 YAN
COMMON/MANGLE/ ALPMAC(55,10), ALPHAT(55,10) ARMT(55,10)
COMMON/MINGEN/ HUL(30,20), TWI(30,20), MWPI(20),MCHORD,
1 NSPAN, ISOLID, MPLANE, ICAMB, ITWIST,ITWICK,IFORNI(10)
2 PIVOT(10), BIN(10), MUPANL,XL6(10),IT6(10),BROLL
COMMON/INPUT / TX(2, 50)
COMMON/POINTS/ TY(4), TYP(4)
COMMON/POINTS/ NPTS, NVAR
DIMENSION XC(40), XCR(40), YC(40), AREA(40)
RD = 57.29577957
NEODY = 1
NBOOTS = IFIX( BBOOTS )
NTHET = IFIX( BTHET )
MC = XMC
NB = XNB
NC1 = NC - 1
NB1 = NB - 1
NDC = NDC1
NW2 = 2*NWING
CALL BODYTPE(XB, RB, LB, THETA, MBPFL, MTHET,NBOOTS, XOS1, XOS2,
1 DSB1, DSB2, XLAST)
SET DRDX(1) = SLOPE OF THE TIP OF BODY
DO 90 K=1, NBOOTS
DSDX(MBOOTS+2-K) = DSDX(MBOOTS+1-K)
CONTINUE
DSDX(1) = DSB1
PCTC = 0.1
THIST = FLOAT( THIST )
THICK = FLOAT( THICK )
CARP = FLOAT( ICAMB )
WING = FLOAT(WWING)
SOLID = FLOAT(SOLID)
PUNCH = FLOAT(IPUNCH)
YAN = YAN * RB
BROLL = BROLL * RB
REWIND 11
1 WRITE (11,800) XNACH, PINF, POLAR, PUNCH, THICK, CARD, TWIST,
2 SOLID, WING, INC, XNB, BBOOTS, BTHET,(SLD(K), K=1,
20), (IFORNI(K), K=1,10), XHON, ARB, YAN, BROLL
WRITE (11,800) (XB(K), K=1,NBOOTS)
WRITE (11,800) (RB(K), K=1,NBOOTS)
WRITE (11,800) (THETA(K), K=1,NTHET)
DO 400 I = 1, NW2, 2
NW1 = I/2 + 1
J = I
IF ( IFORNI(NW1) .EQ. 2 ) J = NW1
CALL WINGC( NW1, XC, XCR, YC, AREA)
WRITE (11,800) (XC(K), K=1,NBC)
WRITE (11,800) (XCR(K), K=1,NBC)
WRITE (11,800) (YC(K), K=1,NBC)
WRITE (11,800) (AREA(K), K=1, NDC)
WRITE (11,800) (ARMT(K), NW1, K=1, NW1)

```


[illegible]

```

1190 BBETAN(1)=BBETAN(1)/BETA
BBETAN(3)=BBETAN(3)/BETA
1200 BBETAN(2)=BBETAN(1)
BBETAN(4)=BBETAN(3)
C
C CORNER POINTS K OF INFLUENCING PANEL I
C
DO 2420K=1,4
C
R(K)=0.0
C(K)=0.0
G(K)=0.0
BPM(K)=0.
CPM(K)=0.
APM(K)=0.
CPM(K)=0.
BPM=BBETAN(K)
IF(LV59)1240,1240,1210
1210 IF(SL)1220,1240,1230
1220 IF(K-2)2420,2420,1240
1230 IF(K-1)1240,1240,1240
1240 IF(ABS(BPM)-EPS)1260,1260,1250
1250 SM=SIGN(1.0,BPM)
GO TO 1270
1260 SM=1.0
BPM=0.0
1270 IF(ABS(ABS(BPM)-1.0)-EPS)1280,1280,1290
1280 BPM=SM
1290 BPM=SM*BPM
BPM2=BPM*BPM
CONSTC=1.-APM2*BPM2
C
C CONFIGURATION SYMMETRY CONDITION
C
C
DO 2410 ISIDE=1,ISYN
IF(ISIDE.EQ.2.AND.TEST.LE.EPS) GO TO 2410
SM=(-1.0)**ISIDE
DELTA1 = THETA(I) + THETA(J)*SM
IF(ABS(DELTA1)-LE.EPS) DELTA1=0.
SINT=SIN(DELTA1)
COSD=COS(DELTA1)
DELTA1=SM*(-YC(J))-Y(I,M,K)
DELTA2= ZC(J)-Z(I,M,K)
IF ( KACE .EQ. 1 ) GO TO 200
IF ( I .LE. NBOYT .AND. J .LE. NBOYT ) GO TO 200
DELTA2 = -SM*ZC(J) - Z(I,M,K)
IF ( BROLL .GT. 0.01) SM=(-1.0)**(ISIDE
+ 1)
C 200 CONTINUE
YPM=XCPY(J)-X(I,M,K)
VPM=(DELTA1-COST*DELTA2-SINT)*SM
ZPM=DELTA2-COST-DELTA1*SINT
IF (ABS(ZPM)-EPS)1300,1310
1300 ZPM=0.0
1310 Z2=SIGN(1.,ZPM)
ZPM2=ZPM*ZPM
IF (ABS(YPM)-EPS)1320,1320,1330
1320 YPM=0.0
1330 Y2=SIGN(1.,YPM)
YPM2=YPM*YPM
YPM2=YPM*YPM
1340 CONTINUE
C

```

C VELOCITY COMPONENTS INDUCED BY PANELS PARALLEL TO X- AXIS

```

ZPM=ZBR
TERMA=XIPM-BPM*YPM
TERMA=TERMA**2
TERMB=YPM*ZPM
TERMC=SQRT(TERMB)
IF(MACH-1.)1580,1810,1350
1350 TERMD=BPM*Z-1.
TERME=-TERMD
TERMG=TERMA*TERMD*YPM
IF(XIPM-TERMC)1360,1530,1370
1360 IF(BPM*Z-1.)1510,1570,1570
1370 IF(XIPM)1380,1410,1380
1380 FTR1=(BPM*TERMD-XIPM*YPM)/SQRT(TERMB*TERMG)
1390 IF(FTR1-1.)1450,1450,1390
1400 FTR1=ACOS(FTR1)*52
60 TO 1460
1410 IF(YPM)1420,1440,1430
1420 FTR1=0.
60 TO 1460
1430 IF(XIPM-BPM*YPM)1420,1440,1450
1440 FTR1=PI/2.
60 TO 1460
1450 FTR1=PI-52
1460 IF(BPM*Z-1.)1470,1480,1480
1470 FTR2=-SQRT(TERME)*ACOS((BPM*XIPM-YPM)/SQRT(TERMG))
60 TO 1490
1480 FTR2=SQRT(TERMD)*ACOSH((BPM*XIPM-YPM)/SQRT(TERMG))
1490 FTR4=SQRT(XIPM*Z-TERMC)/TERMD
FTR5=ACOSH(XIPM/TERMC)
B=CONSTB*(BPM*FTR5-YPM*FTR4-FTR2)
S=CONSTB*(BPM*FTR1-ZPM*FTR4)
P=FTR1/(4.*PI)
60 TO 2370
1510 IF(YPM)1570,1570,1520
1520 IF (XIPM-(BPM*YPM-SQRT(TERME)*ABS(ZPM)))1570,1540,1530
1530 IF (YPM-BPM*XIPM)1570,1540,1550
1540 CT=.5
60 TO 1560
1550 CT=1.0
1560 B=CONSTA*SQRT(TERME)*CT
S=CONSTA*BPM*52*CT
P=CT*52/4.
60 TO 2370
1570 B=0.0
S=0.0
P=0.0
60 TO 2370
1580 TERMD=1.-BPM*Z
TERME=SQRT(TERMD)
IF(ZPM)1600,1590,1680
1590 IF(TERMAA-57)1600,1610,1620
1600 FTR1=0.
60 TO 1630
1610 FTR1=PI/2.
60 TO 1630
1620 FTR1=PI
1630 IF(YPM)1640,1720,1720
1640 TERMG=TERMAA/(YPM-BPM*XIPM)
1650 IF(ABS(TERMG)-.001)1650,1650,1720
1650 TERMA=TERMAA/ABS(YPM)
IF(TERMD)1660,1670,1660
1660 FTR2=ALOG(TERMAA*TERMD/(2.*TERMD)*(1.-BPM*TERMD/TERMD))
60 TO 1730

```

```

1670 FTR2=ALOG(TERM)
60 TO 1730
1680 FTR1=(BPM*TERM-BPM*YPM)/SORT(TERM*(TERMA+TERM-ZPM2))
IF (FTR1-1.)1740,1740,1690
1690 IF (FTR1-1.)1710,1700,1700
1700 FTR1=0.
60 TO 1720
1710 FTR1=ACOS(FTR1)*S2
1720 FTR2=YPM*BPM*YPM*TERM-SORT(XIPM2*TERM)
IF (FTR2-LE. 0.0) 60 TO 1650
FTR2=ALOG(FTR2)
1730 FTR3=(XIPM*SORT(XIPM2*TERM))/TERM
IF (FTR3)1740,1750,1760
1740 FTR1=PI*S2
60 TO 1720
1750 FTR4=0.
60 TO 1770
1760 FTR4=BPM*ALOG(FTR3)
1770 IF (ZPM)1780,1790,1780
1780 FTR5=ATAN(YPM/ZPM)
60 TO 1800
1790 FTR5=PI/2.-S2
1800 P=(FTR1-FTR5)/(8.*PI)
S=CONSTB*(BPM*(FTR1-FTR5)-ZPM*FTR3)/2.
B=CONSTB*(YPM*FTR3-FTR4-TERM*FTR2)/2.
60 TO 2370
1810 IF (XPM)1570,1570,1820
1820 TERM=BPM**2
TERMA=(XPM*BPM*YPM)**2
IF (ZPM)1830,1840,1830
1830 FTR1=ACOS((-XPM*YPM*BPM*TERM)/SORT(TERM*(TERMA+TERM-ZPM2)))*S2
60 TO 1880
1840 IF (YPM)1850,1850,1860
1850 FTR1=0.
60 TO 1880
1860 IF (XPM*BPM*YPM)1850,1850,1870
1870 FTR1=PI
1880 P=FTR1/(4.*PI)
S=((ABS(XPM)*ZPM/TERM-BPM*FTR1)/(4.*PI)
IF (BPM)1890,1900,1890
1890 B=(ABS(XPM)*YPM/TERM-BPM*ALOG(BPM*TERM/SORT(TERMA+ZPM2*TERM)))
1/(4.*PI)
60 TO 2370
1900 B=ABS(XPM)*YPM/(TERM*PI*4.)
60 TO 2370
1910 CONTINUE
C
C VELOCITY COMPONENTS INDUCED BY WING SOURCES
ZPM=ZBR
TERMA=XIPM*BPM*YPM
TERMA=(XIPM*BPM*YPM)**2
TERM=YPM2*ZPM2
TERM=SORT(TERM)
IF (XPM-1.)2190,2370,1920
1920 TERM=BPM2-1.
TERM=TERM
IF (XIPM*TERM)1930,1930,1940
1930 IF (TERM)2100,1570,1570
1940 FTR1=ACOSH(XIPM/TERM)
IF (TERM)1950,1960,1970
1950 FTR2=ACOS((BPM*XIPM*YPM)/SORT(TERMA+TERM-ZPM2))/SORT(TERM)
60 TO 1980
1960 FTR2=SORT(XIPM2-TERM)/(XIPM*YPM)
60 TO 1960
1970 FTR2=ACOSH((BPM*XIPM*YPM)/SORT(TERMA+TERM-ZPM2))/SORT(TERM)

```

```

1980 IF (ZPM) 1990, 2020, 1990
1990 FTR3= (-XIPM+YPM+BPW+TERMD)/SQRT(TERMD*(TERMA+TERMD+ZPM2))
2000 IF (FTR3+1.0) 2060, 2060, 2000
2010 FTR3=ACOS(FTR3)+S2
2020 IF (YPM) 2030, 2030, 2040
2030 FTR3=0
2040 IF (XIPM+BPW+YPM) 2030, 2050, 2040
2050 FTR3=PI/2
2060 FTR3=PI+S2
2070 IF (LVSD) 2080, 2090, 2090
2080 P=TERMAA+PTR2+YPM+PTR1-ZPM+PTR3/(CHORD(NCT)+PI)
S=CONSTD*(TERMAA*(PTR1-BPW+PTR2)+BPW+ZPM+PTR3
1-SQRT(XIPM2-TERMD))/CHORD(NCT)
D=CONSTD*(ZPM*(BPW+PTR1-TERMD+PTR2)-TERMAA+PTR3)
1/CHORD(NCT)
60 TO 2370
2090 P=PTR2/(BETA+PI)
S=(PTR1-BPW+PTR2)/PI
D=PTR3/PI
60 TO 2370
2100 IF (YPM) 1570, 1570, 2110
2110 IF (XIPM-TERMD) 2120, 2130, 1940
2120 IF (XIPM-(BPW+YPM+SQRT(TERMD)+ABS(ZPM))) 1570, 2140, 2130
2130 IF (YPM-BPW+XIPM) 1570, 2140, 2130
2140 CT=-5
60 TO 2160
2150 CT=1
2160 IF (LVSD) 2170, 2180, 2180
2170 P=TERMAA/SQRT(TERMD)-ZPM+CT+S2/CHORD(NCT)
S=BPW+P+BETA
D=BETA*(TERMAA+CT+S2-ZPM+SQRT(TERMD))/CHORD(NCT)
60 TO 2370
2180 P=CT/(SQRT(TERMD)+BETA)
S=BPW+P+BETA
D=CT+S2
60 TO 2370
2190 TERMD=BPW2+1
TERNE=SQRT(TERMD)
PTR1=XIPM/TERNE
IF (PTR1) 2210, 2200, 2200
2200 PTR1=ASINH(PTR1)
60 TO 2220
2210 PTR1=-ASINH(-PTR1)
2220 PTR2=(YPM+BPW+XIPM)/SQRT(TERMA+TERMD+ZPM2)
IF (PTR2) 2240, 2230, 2230
2230 PTR2=ASINH(PTR2)/TERNE
60 TO 2250
2240 PTR2=-ASINH(-PTR2)/TERNE
2250 IF (YPM) 2260, 2290, 2260
2260 FTR3=(BPW+TERMD-XIPM+YPM)/SQRT(TERMD*(TERMA+TERMD+ZPM2))
IF (FTR3+1.0) 2330, 2330, 2270
2270 IF (PTR3-1.0) 2280, 2300, 2300
2280 FTR3=ACOS(FTR3)+S2
60 TO 2340
2290 IF (YPM) 2300, 2300, 2310
2300 FTR3=0
60 TO 2340
2310 IF (XIPM+BPW+YPM) 2300, 2320, 2330
2320 FTR3=PI/2
60 TO 2340
2330 FTR3=PI+S2

```

```

2340 FTR4=SQRT((VPR2+TERB)
      IF(LVS0)2350,2360,2360
2350 P=TERMAA*(FTR2+YPR*(FTR1-2PR*(FTR3)/(CHORD(INCT))2.0*PI)
      S=-CONST0*(TERMAA*(FTR1-BPR*(FTR2)+BPR*(FTR3-FTR4)/
      1(2.0*CHORD(INCT))
      D=-CONST0*(2PR*(BPR*(FTR1-TERB*(FTR2)-TERMAA*(FTR3)/CHORD(INCT)
      60 TO 2370
2360 P=FTR2/(2.0*BETA*PI)
      S=-(FTR1-BPR*(FTR2)/(2.0*PI)
      D=PTP3/PI
2370 CONTINUE
      SIM=1
      R(X)=R(K)*S*SINT
      IF(TMKW.NE.1.0) 60 TO 2390
      IF(DELTA.TEQ.0.0.AND.2PM.EQ.0.0) 0=0.0
      60 TO 2400
2390 IF(L.EQ.J.AND.ISIDE.EQ.1) P=0.
      IF(1SIDE.6T.1) 60 TO 2400
      IF(IM-LE.0)60 TO 2400
      VS2(IM)=0.
      IF(J.LT.1) 60 TO 2400
      NA=(IN-1)/NR+1
      NB=(JM-1)/NR+1
      IF(CNA.EQ.NB) S=0.
      IF(J.EQ.1) VS1(IM)=.25*BPM1
      IF(J.6T.1) VS2(IM)=.25*(BPM1-BPM3)
2400 C(K)=C(K)*D+COSDT*SM
      Q(K)=Q(K)*P*SM
      RPH(K)=RPH(K)*S*SM
      CPH(K)=CPH(K)*D*SM
      RPH(K)=RPH(K)*S*SM
      CPH(K)=CPH(K)*D*SM+SN
      CPH(K)=CPH(K)*D*SM+SN
2410 CONTINUE
2420 CONTINUE
      C
      U(J)=U(J)*Q(1)-Q(2)-Q(3)+Q(4)
      V(J)=V(J)*(R(1)-R(2)-R(3)+R(4))
      W(J)=W(J)*(C(1)-C(2)-C(3)+C(4))
      VPR(J)=VPR(J)*(RPH(1)-RPH(2)-RPH(3)+RPH(4))
      WPR(J)=WPR(J)*(CPH(1)-CPH(2)-CPH(3)+CPH(4))
      VPR(J)=VPR(J)*(RPH(1)-RPH(2)-RPH(3)+RPH(4))
      WPR(J)=WPR(J)*(CPH(1)-CPH(2)-CPH(3)+CPH(4))
2430 CONTINUE
      C
      CORRECTING WING DIEDRAL EFFECT, SEE EG(42) OF REPORT
      C
      IF(LVS0)2460,2460,2460
2460 00 2450 J=1,NPANEL
      U(J)=U(J)
      V(J)=VPR(J)*COST-WPR(J)*SINT
      W(J)=WPR(J)*COST+VPR(J)*SINT
2450 A(J)=V(J)*W(J)
      C
      60 TO 2510
2460 IF(IT)2490,2490,2470
2470 00 2480 J=1,NPANEL
      US(J)=U(J)
      VS(J)=VPR(J)*COST-WPR(J)*SINT
      WS(J)=WPR(J)*COST+VPR(J)*SINT
2480 AS(J)=V(J)*W(J)
      C
      IF(IT)2510,2510,1080

```



```

C      NUS = 1/PIX(WING)
C      BCOND=0
      KOND=BCOND
      IF (STR)1010,1000,1010
1000  1SYN=1
      GO TO 1020
1010  1SYN=2
1020  CONTINUE
C      CODE TO INDICATE IF THE CASE INCLUDES A THICK WING
C      THKW = THKOP
C      TEST FOR TYPE OF CONFIGURATION
C      IF (MBOY)1040,1030,1040
C      WING ONLY CASE
C1030  KACE=1
      JSAVE(1)=0
      NSAVE(1)=MWING
      MPANEL=MWING
      MBOYS=0
      MRG=1
      GO TO 1070
C1040  CONTINUE
C      READ NUMBER OF BODY SOURCE SEGMENTS AND THE INDEX OF THE X -
C      COORDINATE AT LEADING EDGE OF THE WING
C      READ (MTAPE) (XYZ(1),1=1,3),2A,3,Y
      ALPHA=XYZ(3)/XYZ(1)
      READ (MTAPE) MBOYS,MZLE
C      READ BODY X-COORDINATES, BODY RADIUS, AND INCREMENT OF BODY CAMBER
C      READ (MTAPE) (XB(1),R(1),ZDELTA(1),1=1,MBOYS)
      REMIND MTAPE
C      IF (MBOY)1050,1280,1040
C      BODY ONLY CASE
C1050  KACE=2
      MPANEL=0
      MWING=0
      MWINGS = 0
      WRITE (MTAPE) MBOY,MWING,MHACH,SYN,KACE,THROPP,YAW,ROLL
C      GO TO 1160
C      WING AND BODY CASE
C1060  KACE=3
      JSAVE(1)=0
      NSAVE(1)=MBOY
      JSAVE(2)=MBOY
      NSAVE(2)=MWING
      MPANEL=MWING+MBOY
      MRG=2
C1070  CONTINUE
C      DO 1120 L=1,MRG
C      J=JSAVE(L)
C      N=NSAVE(L)

```

```

DO 1100 I=1, N
J=J+1
C
C READS IN PANEL COORDINATES
READ (NTAPEC) NPM, MPART(J), X(J,1,K), Y(J,1,K), Z(J,1,K), K=1,4
IF (MPART(J)-1) 1080, 1100, 1080
1080 MPT=MPART(J)
DO 1090 M=2, MPT
READ (NTAPEC) X(J,M,K), Y(J,M,K), Z(J,M,K), K=1,4
1090 CONTINUE
1100 CONTINUE
C
J=J+MVE(L)
DO 1110 I=1, N
J=J+1
C
C READS IN COORDINATES OF CENTROIDS, CONTROL POINTS, AREAS, THETAS,
ALPHAS, AND CHORD LENGTHS
READ (NTAPEC) NPM, TBAR(J), TBAR(J), TCBAR(J), TCC(J), ZC(J), AREA(J),
THETA(J), ALPHAS(J), CHORD(J)
1110 CONTINUE
C
C READS IN THE NUMBER OF PANELS IN A COLUMN AND THE LOCATION OF THE
CONTROL POINT
READ (NTAPEC) MROW(L), XCTP
1120 CONTINUE
IF (MROW - NE. 0) READ (NTAPEC) MTHETA
IF (MROW - NE. 0) READ (NTAPEC) (THETA(M), M = 1, MTHETA)
REWRITE NTAPC
C
DO 1140 J=1, NMING
JJ=J+MROW
C
C STORES THICKNESS SLOPES AND CAMBER SLOPES
ALPHAT(J)=0.0
ALPHAC(J)=0.0
NS=MROW+1
WRITE 1ST FILE ON TAPE
WRITE (NTAPEC) MROW, NMING, XACH, YTM, KACE, TMCORP, YAW, BROLL
C
C
C
C WRITES COORDINATES OF CENTROIDS, CONTROL POINTS, AREAS, THETAS,
ALPHAS, CHORD LENGTHS, NUMBER OF REGIONS, NUMBER OF ROWS PER
COLUMN IN A REGION, AND CONTROL POINT LOCATION
WRITE (NTAPEC) (1, TBAR(1), TBAR(1), TCBAR(1), TCC(1), ZC(1), AREA(1),
THETA(1), ALPHAS(1), CHORD(1), I=1, MPAHEL), MRC, (MROW(1), I=1, MRS), NC
21P
CALL FOR EOF (NTAPEC)
C
C WRITE THICKNESS SLOPES. 2ND FILE ON TAPE
WRITE (NTAPEC) (ALPHAT(1), I=1, NMING)
CALL FOR EOF (NTAPEC)
C
C WRITE CAMBER SLOPES. 3RD FILE ON TAPE
WRITE (NTAPEC) (ALPHAC(1), I=1, NMING)
CALL FOR EOF (NTAPEC)
C
C IF (MROW) 1160, 1230, 1160
1160 CONTINUE
C
C WRITE SLOPE OF BODY AXIS WITH RESPECT TO THE DEFINING AXIS
FIRST PART OF 4TH FILE ON TAPE
WRITE (NTAPEC) ALPHA, OSB1, OSB2, XOS1, XOS2
C

```

```

C WRITE NUMBER OF BODY SOURCE SEGMENTS, INDEX OF X- COORDINATE AT
C LEADING EDGE OF WING, HEIGHT OF WING PLANE ABOVE BODY AXIS, X
C COORDINATE OF BODY. SECOND PART OF 4TH FILE ON TAPEC.
C WRITE (NTAPEC) NBOOTS,NMLE,ZA,(XB(1),I=1,NBOOTS)
C IF (NWING)1180,1170,1180
1170 CONTINUE
C
C FOR BODY ONLY CASE, READ NUMBER OF THETAS REPRESENTING BODY
C READ (NTAPEC) NOUNNY,NOUNNY,NOUNNY,NOUNNY,NOUNNY,NOUNNY
C
C READ BODY THETAS
C READ (NTAPEC) (THETB(J),J=1,NTNETB)
C REWIND NTAPED
1180 DO 1190 J = 1,NTNETB
1190 THETB(J) = 57.29578*THETB(J)
C IF (NWING .EQ. 0) GO TO 1210
C FOR WING-BODY CASE COMPUTE THETAS FROM PANEL REPRESENTATION OF BOB
C NROUB = NROUB(1)
C NTHETA = NROUB/NROUB
C DO 1200 J = 1, NTHETA
C JJ = (J-1)*NROUB+1
1200 THETAB(J) = ABS(THETAB(J))+57.29578
C
C WRITE NUMBER OF THETAS, AND THETAS. 3RD PART OF 4TH FILE ON C
C WRITE (NTAPEC) NTHETA,(THETAB(I),I=1,NTHETA)
1210 WRITE (NTAPEC) NTHETA,(THETB(I),I=1,NTNETB)
C
C WRITE RADII AND CAMBER INCREMENT OF BODY. LAST PART OF 4TH FILE
C WRITE (NTAPEC) (R(1),I=1,NBOOTS)
C WRITE (NTAPEC) (ZDELTA(1),I=1,NBOOTS)
C CALL FOR EOF (NTAPEC)
C IF (NBOOTS)1220,1220,1240
1220 CONTINUE
C CALL FOR EOF (NTAPEC)
C CALL FOR EOF (NTAPEC)
1230 CONTINUE
C CALL FOR EOF (NTAPEC)
C
C STORE GEOMETRY DATA ON SCRATCH/SAVE TAPE
C FOR USE IN FLOW VISUALIZATION LINK
1240 CONTINUE
C IF (NBOOTS .NE. -1) NWINGS=NWING+NWING/NROUB(NRG)
C WRITE 5TH FILE ON TAPEC
C WRITE (NTAPEC) KACE,NPANEL,NBODY,NWING,NBOOTS,NWINGS
C 1,NROUB,NRACH,SYN
C CALL FOR EOF (NTAPEC)
C
C IF (NPANEL)1270,1270,1250
1250 CONTINUE
C WRITE (NTAPEC) NPANEL
C DO 1260 J=1,NPANEL
C WRITE (NTAPEC) NPART(J)
C NPART=NPART(J)
C WRITE (NTAPEC) ((X(J,M,K),Y(J,M,K),Z(J,M,K),K=1,4),M=1,NPT)
1260 CONTINUE
C WRITE (NTAPEC) (EBAR(J),YBAR(J),ZBAR(J),J=1,NPANEL)
C WRITE (NTAPEC) (XC(J),YC(J),ZC(J),J=1,NPANEL)
C WRITE (NTAPEC) (ALPHAS(J),THETA(J),CHORD(J),J=1,NPANEL)
1270 CONTINUE
C CALL FOR EOF (NTAPEC)
1280 CONTINUE
C
C
C

```

```

C -----
C RETURN
C 1290 FORMAT(7F10.0)
C END
C
C OVERLAY(WOOD,4,0)
C PROGRAM REDUCE
C
C -----
C REDUCE AERODYNAMIC INFLUENCE COEFFICIENTS MATRIX AND FORM THE
C (D) AND (E) MATRICES
C -----
C
C COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPEF,NTAPEG,NTAPEH,NTAPEI,
C 1 NTAPEO,NBODY,NWING,XRACH,SYM,KACE,MPOLAR,IRV
C
C DIMENSION AB(100),ABB(100,100),ABW(100,25),ABV(110,100),AWW(110,25
C 1),E(100,25),B(110),NSIZE(5)
C
C EQUIVALENCE (ABB(1,1),ABW(1,1))
C
C COMPUTE SIZE OF PARTIONS
C MAXP=25
C CALL SIZE (NWING,MAXP,NPART,NSIZE)
C DO 1000 J=1,NBODY
C 1000 READ (NTAPEF) (ABW(I,J),I=1,NWING)
C CALL FSI(1,NTAPEF,IRR)
C
C FORM (D) MATRIX
C DO 1020 K=1,NBODY
C READ (NTAPEF) (AB(I),I=1,NBODY)
C DO 1010 I=1,NWING
C D(I)=0.0
C DO 1010 J=1,NBODY
C D(I)=D(I)+ABW(I,J)*AB(J)
C 1010 CONTINUE
C 1020 WRITE (NTAPEF) (D(I),I=1,NWING)
C 1020 CONTINUE
C CALL FOR EOF (NTAPEF)
C REWIND NTAPEF
C CALL FSI(1,NTAPEF,IRR)
C DO 1030 J=1,NBODY
C 1030 READ (NTAPEF) (ABB(I,J),I=1,NBODY)
C
C COPY INVERSE BODY MATRIX
C DO 1040 J=1,NBODY
C 1040 WRITE (NTAPEF) (ABB(I,J),I=1,NBODY)
C
C MATRIX REDUCTION
C DO 1130 L=1,NPART
C NS=NSIZE(L)
C IF (L.EQ. 1) GO TO 1060
C DO 1050 J=1,NBODY
C 1050 READ (NTAPEF) (ABB(I,J),I=1,NBODY)
C 1060 REWIND NTAPEF
C DO 1070 K=1,NS
C 1070 READ (NTAPEF) (ABW(I,K),I=1,NBODY), (AWW(I,K),I=1,NWING)
C
C FORM (E) MATRIX
C DO 1080 K=1,NS
C DO 1080 M=1,NBODY
C E(M,K)=0.
C DO 1080 N=1,NBODY
C E(M,K)=E(M,K)+ABB(M,N)*ABW(N,K)

```



```

C
C
      CALL FSP(1,NTAPEA,IRR)
      CALL FSP(1,NTAPEB,IRR)
      DO 1000 J=1,NBODY
        READ (NTAPEB) (U(I),V(I),W(I),I=1,NBODY), (UU(I),VV(I),WW(I),I=1,
          1,NWING)
        WRITE (NTAPEB) (U(I),V(I),W(I),I=1,NBODY)
        WRITE (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)
      1000 CONTINUE
      CALL FOR EOF (NTAPEB)
      REWIND NTAPEA
      CALL FSP(1,NTAPEA,IRR)
C
      DO 1010 J=1,NBODY
        READ (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)
        WRITE (NTAPEB) (UU(I),VV(I),WW(I),I=1,NWING)
      1010 CONTINUE
C
      CALL FOR EOF (NTAPEB)
      REWIND NTAPEA
      CALL FSP(1,NTAPEA,IRR)
C
      DO 1020 J=1,NWING
        READ (NTAPEB) (U(I),V(I),W(I),I=1,NBODY), (UU(I),VV(I),WW(I),I=1,
          1,NWING)
        WRITE (NTAPEB) (U(I),V(I),W(I),I=1,NBODY)
        WRITE (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)
      1020 CONTINUE
C
      REWIND NTAPEA
      REWIND NTAPEB
      CALL FOR EOF (NTAPEB)
      CALL FSP(1,NTAPEA,IRR)
C
      DO 1030 J=1,NWING
        READ (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)
        WRITE (NTAPEB) (UU(I),VV(I),WW(I),I=1,NWING)
      1030 CONTINUE
      CALL FOR EOF (NTAPEB)
      REWIND NTAPEA
      REWIND NTAPEB
C
C
C -----
C -----
C -----
C
      OVERLAY(000,6,0)
      PROGRAM 00060
      COMMON / MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPEF,NTAPEG,NTAPEH,NTAPEI,
        1 NTAPEJ,NBODY,NWING,NMACH,STR,KACE,MPOLAR,IRN
      COMMON /TRANSFER/ IOVR10,IOVR60,IOVR70
      COMMON /COMMON/ ISAVET
      IF (IOVR60.EQ.2) GO TO 100
      CALL SAVTAP(ISAVET)
      GO TO 999
    100 CALL USETAP
    999 CONTINUE
      END
C
      SUBROUTINE SAVTAP(ISAVET)
C
C
C ..... WRITES AERODYNAMIC MATRICES ON A LOGICAL TAPE FOR FUTURE USE .....
C .....

```

```

C
COMMON / MAIN / NTAPEA,NTAPED,NTAPEE,NTAPEF,NTAPEG,NTAPEH,NTAPEI,NTAPEJ,
1  NTAPED,NBODY,NWING,NKACH,SYN,KACE,TKNU,YAU,BROLL
COMMON / NTHICK/TKNU
DIMENSION A(210),B(210),C(210)
DIMENSION DUM(10),NROW(2)
C
IF (ISAVE) 1010,1010,1000
1000 READ (NTAPEE)NBODY,NWING,NKACH,SYN,KACE,TKNU,YAU,BROLL
C
C *****
GO TO (1020,1010,1020),KACE
C
1010 CALL FSF(6,NTAPEE,IRR)
GO TO 1090
1020 NPAEL=NBODY+NWING
READ (NTAPEE) ( M, ( DUM(J), J=1,10), I=1,NPAEL), NR6,
1 (NRON(I),I=1,NR6),DUNNY
NCOL=NWING/NROW(NR6)
CALL FSF( 6,NTAPEE,IRR)
GO TO (1030,1090,1060),KACE
1030 IF (TKNU.EQ. 0.) GO TO 1040
CALL TTAPE(0,NWING,NWING+NCOL,NTAPEA,NTAPEE,A,B,C)
1040 CALL FSF(1,NTAPEA,IRR)
CALL TTAPE(0,NWING,NWING+1,NTAPEA,NTAPEE,A,B,C)
CALL FSF(1,NTAPEA,IRR)
CALL TTAPE(0,NWING+1,NWING+1,NTAPEA,NTAPEE,A,B,C)
CALL FSF(1,NTAPEA,IRR)
CALL TTAPE(0,NWING+2,NWING+2,NTAPEA,NTAPEE,A,B,C)
REWIND NTAPED
IF (TKNU.EQ. 0.) GO TO 1050
CALL TTAPE(1,NWING,NWING+NCOL,NTAPEE,NTAPEE,A,B,C)
1050 CALL FSF(1,NTAPEE,IRR)
CALL TTAPE(1,NWING,NWING,NTAPEE,NTAPEE,A,B,C)
REWIND NTAPED
CALL TTAPE(0,NWING,NWING,NTAPEE,NTAPEE,A,B,C)
CALL FSF(1,NTAPEE,IRR)
CALL TTAPE(0,NWING,NWING,NTAPEE,NTAPEE,A,B,C)
REWIND NTAPED
GO TO 1090
C
1060 IF (TKNU.EQ. 0.) GO TO 1070
CALL TTAPE(0,NPAEL,NWING+NCOL,NTAPEA,NTAPEE,A,B,C)
1070 CALL FSF(1,NTAPEA,IRR)
CALL TTAPE(0,NWING,NWING,NTAPEA,NTAPEE,A,B,C)
CALL FSF(1,NTAPEA,IRR)
CALL TTAPE(0,NWING+1,NWING+1,NTAPEA,NTAPEE,A,B,C)
CALL FSF(1,NTAPEA,IRR)
CALL TTAPE(0,NWING+2,NWING+2,NTAPEA,NTAPEE,A,B,C)
REWIND NTAPED
IF (TKNU.EQ. 0.) GO TO 1080
CALL TTAPE(1,NPAEL,NWING+NCOL,NTAPEE,NTAPEE,A,B,C)
1080 REWIND NTAPED
CALL TTAPE(1,NBODY,NBODY,NTAPEE,NTAPEE,A,B,C)
CALL FSF(1,NTAPEE,IRR)
CALL TTAPE(1,NWING,NBODY,NTAPEE,NTAPEE,A,B,C)
CALL FSF(1,NTAPEE,IRR)
CALL TTAPE(1,NBODY,NBODY,NTAPEE,NTAPEE,A,B,C)
CALL FSF(1,NTAPEE,IRR)
CALL TTAPE(1,NWING,NWING,NTAPEE,NTAPEE,A,B,C)
CALL FSF(1,NTAPEE,IRR)
CALL TTAPE(1,NWING,NWING,NTAPEE,NTAPEE,A,B,C)
REWIND NTAPED
CALL TTAPE(0,NWING,NWING,NTAPEE,NTAPEE,A,B,C)
CALL FSF(1,NTAPEE,IRR)
CALL TTAPE(0,NWING,NWING,NTAPEE,NTAPEE,A,B,C)
REWIND NTAPED

```

```

CALL TTAPE(0,MWING,NBODY,NTAPEF,NTAPEC,A,B,C)
CALL PSF(1,NTAPEF,IRR)
CALL TTAPE(0,NBODY,NTAPEF,NTAPEC,A,B,C)
CALL TTAPE(0,MWING,NBODY,NTAPEF,NTAPEC,A,B,C)
REWIND NTAPEF

```

```

C 1090 CONTINUE
CALL FOR EOF (NTAPEC)
REWIND NTAPEC

```

```

C -----
C RETURN
C -----
C END

```

```

C SUBROUTINE USETAP

```

```

C .....
C READS AERODYNAMIC MATRICES FROM SAVE TAPE AND REWRITES
C .....

```

```

COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEF,NTAPEG,NTAPEH,NTAPEI,NTAPEJ,
1 NTAPEK,NTAPEL,NTAPEM,NTAPEN,NTAPEO,NTAPEP,NTAPEQ,NTAPER,NTAPES,
COMMON /MTHICK/THKN
DIMENSION A(210),B(210),C(210)
DIMENSION DUM(10),NROW(2)

```

```

C READ(NTAPEC)NBODY,MWING,NRACH,SYM,KACE,THKN,YAU,BROLL
C
C GO TO (1000,1070,1040),KACE
1000 MPANEL=NBODY,MWING
READ (NTAPEC) ( M, ( DUM(J), J=1,10), I=1,MPANEL), MRG,
1(NROW(I), I=1,MRG), DUMMY
NCOL=MWING/NROW(MRG)
CALL PSF( 6,NTAPEC,IRR)

```

```

C GO TO (1010,1070,1040),KACE
1010 IF (THKN .EQ. 0.) GO TO 1020
CALL TTAPE(0,MWING,NBODY,NCOLW,NTAPEC,NTAPEA,A,B,C)
1020 CALL FOR EOF (NTAPEA)
CALL TTAPE(0,MWING,NBODY,NTAPEF,NTAPEA,A,B,C)
CALL FOR EOF (NTAPEF)
CALL TTAPE(0,MWING-1,NBODY,NTAPEG,NTAPEA,A,B,C)
CALL FOR EOF (NTAPEG)
CALL TTAPE(0,MWING-2,NBODY,NTAPEH,NTAPEA,A,B,C)
CALL FOR EOF (NTAPEH)
REWIND NTAPEA

```

```

C IF (THKN .EQ. 0.) GO TO 1030
1030 CALL TTAPE(1,MWING,NBODY,NCOLW,NTAPEC,NTAPEB,A,B,C)
CALL FOR EOF (NTAPEB)
CALL TTAPE(1,MWING,NBODY,NTAPEF,NTAPEB,A,B,C)
CALL FOR EOF (NTAPEF)
REWIND NTAPEB
CALL TTAPE(0,MWING,NBODY,NTAPEG,NTAPEB,A,B,C)
CALL FOR EOF (NTAPEG)
CALL TTAPE(0,MWING,NBODY,NTAPEH,NTAPEB,A,B,C)
CALL FOR EOF (NTAPEH)
REWIND NTAPEB

```

```

C GO TO 1070
1040 IF (THKN .EQ. 0.) GO TO 1050
CALL TTAPE(0,MPANEL,MWING,NCOLW,NTAPEC,NTAPEA,A,B,C)
1050 CALL FOR EOF (NTAPEA)
CALL TTAPE(0,MWING,NBODY,NTAPEF,NTAPEA,A,B,C)
CALL FOR EOF (NTAPEF)

```



```

C      PRESSURE INTERPOLATION FOR THE WING
C
C      NSYM = 2
C      CALL OVERLAY(4NWANG, 7, 10)
C      60 TO 600
C
C      50 CONTINUE
C
C      RESTART RUN IRW = 2
C
C      IF ( IRW -EQ. 2 ) 60 TO 100
C
C      16070 = 10VR70 + 1
C      60 TO ( 100, 200, 300, 400, 500 ) , 16070
C
C      100 CONTINUE
C      CALL FORCES
C      60 TO 600
C
C      200 CONTINUE
C      CALL OVERLAY(4NWANG, 7, 1)
C      60 TO 600
C
C      300 CONTINUE
C      CALL OVERLAY(4NWANG, 7, 2)
C      60 TO 600
C
C      400 CONTINUE
C      CALL OVERLAY(4NWANG, 7, 3)
C      60 TO 600
C
C      500 CONTINUE
C      CALL OVERLAY(4NWANG, 7, 4)
C      600 CONTINUE
C
C      END
C
C      SUBROUTINE FORCES
C
C      C ATOL ROUTINE FOR FORCES LINK
C
C
C      COMMON / MAIN / NTAPES,NTAPES,NTAPES,NTAPES,NTAPES,NTAPES,
C      1 NTAPES,NBODY,NWING,IMACH,SYN,KACE,NPOLAR,IRW,IPLOT
C      COMMON / BODYSP / BRX(51), BZ(51), BRXC(51), BRZC(51)
C      COMMON / AVAR / A(210),ACB(21),ABX(100),AWX(110),AREA(210),ARM(2),A
C      1 RUT(20),ALPHA(210),ALPHAB(210),ALPHAT(110),
C      2 ALPHAN(110),ALPHAX(110),AMS(110),ABT(100),AM1N(
C      3 210,21),ALPHAN,ALPHAB,ARA,ARADEG,ARB,ARN(10), ARAS,
C      4 AT, AAA
C      COMMON / BODS / MUMB09,XMOS(3),YMOS(3),ZMOS(3),MBSTAT(3)
C      COMMON / VAR / R(210),BBCL,BBCT,BBCT,BBCT,BBCT,BBCT,BBCT,BBCT
C      COMMON / CVAR / C(210),CHORD(210),CL(210),CPHN(1),1,1
C      1 ,CLS(210)
C      2 CBR,CASE,CPCALC,CBAR,CONSMT,CLBAR,CLX,CLM,CBM
C      COMMON / VAR / R(210),RZDIB(51), BADEG,DARAD
C      COMMON / IVAR / IPOLAR
C      COMMON / KVAR / KASE,KONTIG,NPOLAR
C      COMMON / HVAR / HENT(9),HROM(2)
C      1 MS,MPANEL,NACL,HROMB,HROMW,HCOLV,MTHETA,MTHETS,
C      2 XLE,MAG,NPOLAR,NCLX
C      COMMON / PVAR / POLAR

```



```

C
C      SEMIS      WING SEMI-SPAN USEDIN SPANWISE CB AND CL
C      CALCULATIONS
C      : SEMIS = 1. IF NOT SPECIFIED
C
C      KONFIG=KACE
C      NPANEL=NMING*NBODY
C      NS=NBODY+1
C
C      CDR=57.2957795
C      IF ( LRM -EQ. 2 ) GO TO 1111
C      IF ( KONFIG-2 ) 1000,1010,1000
C
C      READ CONFIGURATION GEOMETRY
C      1000 CONTINUE
C      READ (NTAPEC) NBODY, NMING, XNACH, SYN, KACE, THK, TAN, BROLL
C
C      READ (NTAPEC) IJ, XBAR(I), YBAR(I), ZBAR(I), XC(I), YC(I), ZC(I), AREA(I
C      1), THETA(I), DUMY, CHORD(I), I=1, NPANEL), NEG, (XROH(I), I=1, NEG), RATIOX
C
C      REMIND NTAPEC
C
C      1010 CONTINUE
C      NACEL=0
C
C      READ CARD SA
C      READ (NTAPEC, 1700) CASE, CPCALC, POLAR, THICK, CAMB, XNACEL, PINF,
C      1 PUNCH, SOLID, WING, ( SIDIR, K=1,20), (IFORM(K), K=1,10), IP
C      1 PUNCH = IFIX( PUNCH )
C      ISOLID = IFIX( SOLID )
C      ICAMB = IFIX( CAMB )
C      NNG = IFIX( WING )
C      ANCEL=ABS(XNACEL)
C      NUNBOD=0
C      IF(NBODY.NE.0) NUNBOD=1
C      IPOLAR=0
C      POLAR = POLAR + 1
C      IF (CASE) 1020,1080,1020
C      1020 CONTINUE
C
C      IF (CBAR -EQ. 0.) CBAR=1.
C      IF (SERIS -EQ. 0.) SERIS=1.
C
C      IF (VOUT) 1130,1150,1130
C      1130 CONTINUE
C      DO 1140 J=1,9
C      1140 WMT(J)=J
C      GO TO 1170
C      1150 CONTINUE
C      DO 1160 J=1,9
C      1160 WMT(J)=0
C      1170 CONTINUE
C      KASE=CASE
C
C      1180 IF (KONFIG-2) 1200,1190,1200
C      1190 RFAREA=1.0
C      GO TO 1220
C      1200 RFAREA=0.0
C      DO 1210 J=NS, NPANEL
C      1210 RFAREA=RFAREA+AREA(J)
C      1220 CONTINUE
C

```

```

C      WRITE CASE DATA
C      CALL INOUT(NTAPE, KASE, CP, CALC, POLAR, THICK, VOUT, XNACH, SEMIS,
C      1XP, 2P, CBAR, RAREA, SYN)
C
C      ANY CONFIGURATION
C      IF (KONFIG .EQ. 1) GO TO 1280
C
C      BODY ALONE OR WING-BODY CONFIGURATION
C      MROW=MROW(1)
C
C      IF (KONFIG-2) 1240, 1240, 1250
C
C      BODY ALONE CONFIGURATION
C      READ (NTAPE) MBODY, MWING, XNACH, SYN, KASE
C      GO TO 1260
C
C      WING-BODY CONFIGURATION
C      1250 CALL F3FC3, NTAPEC, IRR)
C      1260 CONTINUE
C      READ (NTAPE) ALPHA, XSB1, XSB2, XSB3, XSB2
C      M1 = XSB1
C      M2 = XSB2
C      READ (NTAPE) MBOOTS, MILE, Z, (XB(1), I=1, MBOOTS)
C      IF (KONFIG .EQ. 2) GO TO 1270
C      READ (NTAPE) MTHETA, (THETA(1), I=1, MTHETA)
C
C      INPUT BODY VARIABLES
C      1270 READ (NTAPE) MTHETA, (THETA(1), I=1, MTHETA)
C
C      READ CARD 7A
C      READ (NTAPE, 1700) ARB, DABEG
C      READ CARD 8A
C      READ (NTAPE) ( R(1), I=1, MBOOTS )
C      READ CARD 9A
C      READ (NTAPE) ( ZDELTA(1), I=1, MBOOTS )
C      REWIND NTAPEC
C
C      REMAX = R(1)
C      DO 1271 I=2, MBOOTS
C      REMAX = MAX( REMAX, R(I) )
C      1271 CONTINUE
C
C      IF ( KONFIG .EQ. 2 ) RAREA = 3.14159 * REMAX**2
C      IF ( KONFIG .EQ. 2 ) WRITE (NTAPE, 1760) RAREA
C
C      ARAT=ALPHA*ARB/CBR
C      WRITE (NTAPE, 1770) ZA
C
C      IF (KONFIG .EQ. 2) GO TO 1290
C
C      WING-BODY CONFIGURATION
C      CALL BCAN(MBOOTS, MROW, XB, ZDELTA, XC, ZDXTD, ACB)
C
C      WING ALONE OR WING-BODY CONFIGURATION
C      1280 CONTINUE
C      MROW=MROW(MRG)
C      MCOL=MWING/MROW
C
C      IF (KONFIG .EQ. 3) GO TO 1290
C
C      WING ALONE CONFIGURATION
C      ALPHA=0.
C      ARAT=0.

```

```

C 1290 CONTINUE
C 1290 CONTINUE
CALL FSF(7,NTAPEC,IRR)
WRITE (NTAPEC) THICK,ARA
CALL FOR EOF (NTAPEC)
REWIND NTAPEC
ALPHA=ALPHA+CDR
ARADES=ARA+CDR
YANDEG = YAN + CDR
ROLDEG = BROLL + CDR
WRITE (NTAPEC,1750) ALPHA
WRITE (NTAPEC,1760) ARADEG
WRITE (6,1763) YANDEG
WRITE (6,1766) ROLDEG

C IF (KONFIG.EQ.2) GO TO 1400

C WING ALONE OR WING-BODY CONFIGURATION
C INPUT WING VARIABLES
C GO TO (1300,1320,1370),KASE

C 1300 READ CARD 10A
C 1300 CONTINUE
ICAN = 2
CALL READNTAPE1,NTAPEO,NTAPEC,MW,MS,ALPHA,KACE,ICAN,MROU)
DO 1310 J=MS,MPADEL
1310 CL(J)=CL(J)
GO TO 1390

C 1320 CONTINUE
C 1320 CONTINUE
IF (KONFIG.EQ.1) READ (NTAPE1,1700) DUM,PADES
READ (NTAPE1,1700) (ARNT(I),I=1,10),TWIST,MSUNG
IF (TWIST.NE.0.) GO TO 1340
DO 1330 J=1,MCOLW
1330 ARNT(J)=0.0
GO TO 1350

C 1340 READ CARD 11AA
C 1340 READ (NTAPE1,1700) (ARNT(I),I=1,MCOLW)
1350 CONTINUE
CALL FSF (2,NTAPEC,IRR)

C ICANB = 2
C CALL READNTAPE1,NTAPEO,NTAPEC,MW,MS,ALPHA,KACE,ICANB,MROU)
C REWIND NTAPEC

C READ IN WING CAMBER (ALPHA) AS ALPHA
C IF (ICANB.EQ.1) READ (NTAPE1,1700) (ALPHA(I),I=1,MWING)
C
C HAR = MWING / MWG
C MHPM = MWING / MWG

C INCORPORATE THE EFFECT OF DIBEDRAL (THETA) AND ROLL (BROLL)
C MHPM NO. OF PANELS PER WING
C
C COSA = COS(ARA)
C SINA = SIN(ARA)
C TANA = TAN(ARA)
C TANY = TAN(YAN)
C A1 = 1.0/SQRT(1.0 + TANA**2 + TANY**2)

```

```

COSB = A1*SQR( 1.0 + TANB**2 )
SINB = SQR( 1.0 - COSB**2 )
DO 1360 J=1,NWING
JJ=(J-1)/NRWB*1
JN=J-NBODY
JC = ( J - 1 ) / NAR + 1
NUN = (J-1) / NUPW + 1
COST = COS( THETA(JN) + BROLL )
SINT = SIN( THETA(JN) + BROLL )
ANGATT = ASIN( SINT*SINB-COSA + COST*SINA )
ALPHA(J)=ALPHA(J)-ANGATT-(ARW(JC)*ARUT(JJ))/CDB
GO TO 1390
C
C READ CARD 13A
1370 READ (NTAPE1,1700) CONSMT,CLBAR,ICPBAR
IF (CONSMT .NE. 0.) GO TO 1380
WRITE (NTAPE0,1730) CLBAR
GO TO 1390
1380 WRITE (NTAPE0,1730) CLBAR
WRITE (NTAPE0,1740) ICPBAR
1390 CONTINUE
C
C WING ALONE OR WING-BODY CONFIGURATION
IF (TRICK)1430,1400,1430
C
1400 CONTINUE
CALL PSF(8,NTAPEC,IRR)
CALL FOR EOF (NTAPEC)
REWIND NTAPEC
IF (KONFIG.EQ.2) GO TO 1480
DO 1420 J=1,NPANEL
1420 ALPHA(J)=0.
DO 1425 J=1,NWING
1425 ANS(J)=0.
IF (KONFIG.EQ.1) GO TO 1460
GO TO 1440
C
C WING ALONE OR WING-BODY CONFIGURATION
1430 CONTINUE
C
C READ CARD 14A
CALL TVELCA,B,C,D,ALPHAT,UBNT,VBNT,VWNT,VUNNT,VUUT,
ALPHA,CHORD,TRICK,NROW)
REWIND NTAPEA
REWIND NTAPEB
C
DO 1455 J=1,NWING
JJ=J-NBODY
1455 ANS(J)=ALPHA(JJ)
IF (KONFIG .EQ. 1) GO TO 1460
C
C WING-BODY CONFIGURATION
1278 CONTINUE
PI = 3.14159265359
COSA = COS(ARA)
SINA = SIN(ARA)
COST = COS(YAW)
SINT = SIN(YAW)
NBD = NBODY
DO 1450 I = 1, NTHETA
DO 1450 K = 1, NROW
J = K + (I-1)*NROW
SINT = SIN( THETA(I) )

```

```

COST = COS( THETA(J) )
SIND = SIN( DRDX(K) )
COSD = COS( DRDX(K) )
ALPHA(J) = ACOS( -COSA*COST*SIND + COSA*SINY*SINT*COSD
+ SINA*COST*COSD ) - 0.5*PI
1 ABX(J)=-ALPHA(J)
ABT(J)=ABX(J)
IF ( TAN .LT. 0.01 .AND. BROLL .LT. 0.01 ) 60 TO 1450
J2 = NBODY + K - I-NRORB
SINT = SIN( THETA(J) - PI )
COST = COS( THETA(J) - PI )
SIND = SIN( DRDX(K) )
COSD = COS( DRDX(K) )
NBD = 2-NBODY
ALPHA(J-NBODY) = ACOS( -COSA*COST*SIND + COSA*SINY*SINT*COSD
+ SINA*COST*COSD ) - 0.5*PI
1450 CONTINUE
C
C ANY CONFIGURATION
1460 CONTINUE
DO 1470 J=1,NWING
  UNBT(J)=0.
  UNBT(J)=0.
  UNBT(J)=0.
1470 ALPHA(J)=0.
C
1480 CONTINUE
CALL PSF(9,NTAPEC,IRR)
WRITE(NTAPEC) SUMRY
IF ( KONFIG .EQ. 1 ) 60 TO 1620
C
C BODY ALONE OR WING-BODY CONFIGURATION
C
1490 INOSE = X0(1)
IF(IRMACH-1.)1500,1620,1510
1500 CALL OVERLAY(4NWANG, 7, 5)
60 TO 1560
C
1510 CALL OVERLAY(4NWANG, 7, 6)
C
1560 CONTINUE
1620 CONTINUE
IF (NWING.LE.0) 60 TO 1628
DO 1625 J=1,NWING
1625 ALPHA(J)=ALPHA(J)+AUSC(J)
1628 CONTINUE
CALL FOR EOP(NTAPEC)
REWIND NTAPEC
C
C ANY CONFIGURATION
C
CALL OVERLAY(4NWANG, 7, 7)
C
FOR GRAPHIC REPRESENTATION, WRITE ON TAPE CPDB, CPB, CPU,CPL
IREPT = 1
IF ( IPLOT .EQ. 1 ) CALL OVERLAY(4NWANG, 7, 13)
1111 CONTINUE
IF ( IRW .GT. 0 ) CALL OVERLAY(4NWANG, 7, 11)
C
IF ( KONFIG .LT. 3 ) 60 TO 1600
C
ADDING PRESSURE DUE TO THE PRESENCE OF WING TO BODY PANEL
C
CALL OVERLAY(4NWANG, 7, 9)

```

```

C      INEPT = 2
      IF ( IPLOT .EQ. 1 ) CALL OVERLAY(4NUANG, 7, 13)
C      1800 CONTINUE
C      NSYM = 1
C      GO TO ( 1820, 1810, 1810 ) , KONFIG
C      C
C      INTERPOLATION OF PRESSURE
C      1810 CONTINUE
C      BODY
C      CALL OVERLAY(4NUANG, 7, 8)
C      IF ( KONFIG .EQ. 3 ) NSYM = NSYM + 1
C      IF ( KONFIG .EQ. 2 ) GO TO 1840
C      1820 CONTINUE
C      WING
C      CALL OVERLAY (4NUANG, 7, 10)
C      1840 CONTINUE
C      NPOLAR = 1
C      ANY CONFIGURATION
C      READ ADDITIONAL POLARS
C      IF ( POLAR ) 1630, 1680, 1630
C      1630 CONTINUE
C      1640 CONTINUE
C      IPOLAR=1
C      PARAD=9ADEG/C9R
C      ARA=ARA+PARAD
C      ARADEG=ARA+C9R
C      WRITE (MTAPE0,1710) TITLE
C      WRITE (MTAPE0,1720) ADEG
C      WRITE (MTAPE0,1760) ARADEG
C      IF (KONFIG .EQ. 2) GO TO 1670
C      WING ALONE OR WING-BODY CONFIGURATION
C      1650 I=1, MWING
C      JN=1, NBODY
C      COST=COS(THETA(JN) * BROLL )
C      1650 ALPHA(I)=ALPHA(I)-PARAD+COST
C      IF (KONFIG .EQ. 1) GO TO 1670
C      WING-BODY CONFIGURATION
C      1660 I=1, NBODY
C      ALPHA(I)=ALPHA(I)-PARAD+COST(THETA(I))
C      ANY CONFIGURATION
C      1670 CONTINUE

```



```

C
1030 NC=NVING/NROW(1)
NR=NROW(1)
60 TO 1050
1040 NC=NVING/NROW(2)
NR=NROW(2)
1050 N=0
NVINC=NVING*NC
IF ( THICK .LT. 1.0 ) 60 TO 1080

C
C INPUT THICKNESS SLOPES
K = -NR + 1
60 1070 I=1,NC
K = K+NR
KK = K+NR - 1
READ (NTAPEC,1190) ATLE(I),(ALPHAT(J),J=K,KK)
1070 CONTINUE
60 TO 1110

C
C USE PREVIOUSLY CALCULATED PANEL SLOPES
C AND COMPUTE LEADING EDGE SLOPES
1080 CALL F5F(1,NTAPEC,1RR)
READ (NTAPEC) (ALPHAT(I),I=1,NVING)
REWIND NTAPEC
NR1=NR-1

C
60 1100 J=1,NC
J1=(J-1)*NR
JN=NBODY*J1
AT=ALPHAT(J1+1)
60 1090 I=1,NR1
IJ=J+I
IJN=NBODY*IJ
AT=AT-ALPHAT(IJ)*ALPHAT(IJ+1)*CHORD(IJN+1)/
1CHORD(JN+1)
1090 CONTINUE
ATLE(J)=AT
1100 CONTINUE

C
1110 CONTINUE
CALL F5F(9,NTAPEC,1RR)
WRITE (NTAPEC) NVINC
60 1118 J=1,NC
J1=(J-1)*NR
IJ=J+I
IJN=NBODY*IJ
IF(I.EQ.1) 60 TO 1114
WS(IJ)=(ALPHAT(IJ-1)*ALPHAT(IJ))*0.5
60 TO 1116
1114 WS(IJ)=(ATLE(J)*ALPHAT(IJ))*0.5
1116 CONTINUE
1118 CONTINUE
60 1170J=1,NVINC
READ (NTAPEC) (B(I),I=1,NPANEL)
READ (NTAPEC) (A(I),B(I),C(I),I=1,NPANEL)
J1=J-N*(NR+1)
IF (J1)1130,1130,1120
1120 N=N+1
WT=ATLE(N)
60 TO 1140
1130 JN=J-N
WT=ALPHAT(JN)
1140 CONTINUE
1140 WRITE (NTAPEC) WT
C

```



```

1070 S(1)=SCONST
GO TO 1080
1080 CONTINUE
READ (NTAPEC) (S(1),I=1,NR)
1090 CONTINUE
C -----
C RETURN
C -----
1100 FORMAT(7F10.0)
END

C
C SUBROUTINE BTTHICK(NBODY,MBODY,MRWB,IB,R,IC,ALPHA,THETA)
C
C .....
C COMPUTES BODY THICKNESS SLOPES FOR GIVEN BODY RADII DISTRIBUTION
C .....
C
C DIMENSION IB(1),R(1),IC(1),ALPHA(1),THETA(1)
C
C MB=NBODY-1
C NCOL=MBODY/MRWB
C
DO 1000 I=1,MB
1800 ALPHA(I+1)=(R(I+1)-R(I))/(IB(I+1)-IB(I))
ALPHA(1)=ALPHA(2)
DO 1030 J=1,MRWB
DO 1010 I=1,MB
IF (IB(I)-IC(J))1010,1020,1020
1010 CONTINUE
--NBODY
1020 THETA(J)=ALPHA(I+1)*(IC(J)-IB(I-1))/(IB(I)-IB(I-1))*(ALPHA(I)-
1ALPHA(I-1))
1030 CONTINUE
C
DO 1040 J=1,MRWB
DO 1060 I=2,NCOL
IJ=(I-1)*MRWB+J
THETA(IJ)=THETA(J)
1040 CONTINUE
C
C RETURN
C -----
C END

C
C SUBROUTINE BCAN(NBODY,MRWB,IB,ZDELTA,IC,DZDB,ACB)
C
C .....
C COMPUTES BODY CANTER SLOPES, GIVEN BODY CANTER SHAPE
C .....
C
C DIMENSION IB(1),ZDELTA(1),IC(1),DZDB(1),ACB(1)
C
C MB=NBODY-1
DO 1000 I=1,MB
DO 1020 J=1,MRWB
ZDB(I+1)=ZDB(I)
1000 ZDB(I+1)=(ZDELTA(I+1)-ZDELTA(I))/(IB(I+1)-IB(I))
ZDB(I)=ZDB(I)
C
DO 1030 J=1,MRWB
DO 1010 I=1,MB
IF (IB(I)-IC(J))1010,1020,1020
1010 CONTINUE
1020 ACB(IJ)=ZDB(I)-ZDB(I-1)*(IC(J)-IB(I-1))/(IB(I)-IB(I-1))*(ZDB(I)-ZDB(I-1))
11-11)

```

```

1030 CONTINUE
C-----
C RETURN
C-----
C END
C
SUBROUTINE INOUT(TAPEO,KASE,CPCALC,POLAR,THICK,VOUT,IRACH,SERIS,
1XP,2P,CBAR,AREA,SYN)
C-----
C PRINTS OUT ALL INPUT AERODYNAMIC DATA
C-----
C
WRITE (NTAPEO,1200)
IF (SYN) 1010,1000,1010
1000 WRITE (NTAPEO,1210)
GO TO 1020
1010 WRITE (NTAPEO,1220)
1020 CONTINUE
C
GO TO (1030,1040,1050),KASE
1030 WRITE (NTAPEO,1230)
GO TO 1060
1040 WRITE (NTAPEO,1240)
GO TO 1040
1050 WRITE (NTAPEO,1250)
1060 CONTINUE
C
IF (CPCALC-1.) 1070,1080,1090
1070 WRITE (NTAPEO,1260)
GO TO 1190
1080 WRITE (NTAPEO,1270)
GO TO 1100
1090 WRITE (NTAPEO,1280)
1100 CONTINUE
C
IF (POLAR 1120,1110,1120
1110 WRITE (NTAPEO,1290)
GO TO 1130
1120 WRITE (NTAPEO,1300)
1130 CONTINUE
C
IF (THICK) 1150,1140,1150
1140 WRITE (NTAPEO,1310)
GO TO 1160
1150 WRITE (NTAPEO,1320)
1160 CONTINUE
C
IF (VOUT) 1180,1170,1180
1170 WRITE (NTAPEO,1330)
GO TO 1190
1180 WRITE (NTAPEO,1340)
1190 CONTINUE
C
WRITE (NTAPEO,1360) IRACH
WRITE (NTAPEO,1370) 1P,2P,CBAR
WRITE (NTAPEO,1380) AREA
WRITE (NTAPEO,1390) SERIS
C-----
C RETURN
C-----

```

```

C -----
1200 FORMAT(1H0,730M DESCRIPTION OF CASE REQUESTED)
1210 FORMAT(1H0,85M UNSYMMETRICAL CONFIGURATION - PANELS LOCATED ONLY 0
1H ONE SIDE OF X-Z PLANE(SYM = 0.))
1220 FORMAT(1H0,80M SYMMETRICAL CONFIGURATION - PANELS LOCATED ON BOTH
SIDES OF X-Z PLANE(SYM = 1.))
1230 FORMAT(1H0,9MCASE = 1.,10X,35M CALCULATE WING SHAPE, GIVEN WING CL)
1240 FORMAT(1H0,9MCASE = 2.,10X,25M CALCULATE CL, GIVEN SHAPE)
1250 FORMAT(1H0,9MCASE = 3.,10X,19M OPTIMIZE WING SHAPE)
1260 FORMAT(1H0,11MPCALC = 0.,8X,9M LINEAR CP)
1270 FORMAT(1H0,11MPCALC = 1.,8X,13M NON-LINEAR CP)
1280 FORMAT(1H0,11MPCALC = 2.,8X,8M EXACT CP)
1290 FORMAT(1H0,10MPOLAR = 0.,9X,20M POLARS NOT REQUESTED)
1300 FORMAT(1H0,10MPOLAR = 1.,9X,16M POLARS REQUESTED)
1310 FORMAT(1H0,10MTHICK = 0.,9X,40M WING THICKNESS PRESSURES NOT TO BE
ADDED)
1320 FORMAT(1H0,10MTHICK = 1.,9X,36M WING THICKNESS PRESSURES TO BE ADDE
ID)
1330 FORMAT(1H0,9MVOUT = 0.,10X,37M VELOCITY COMPONENTS NOT TO BE PRINTE
ID)
1340 FORMAT(1H0,9MVOUT = 1.,10X,33M VELOCITY COMPONENTS TO BE PRINTED)
1350 FORMAT(1H0,17M WING SEMI-SPAN = ,F10.4)
1360 FORMAT(1H0,21M WING REFERENCE AREA = ,F10.4)
1370 FORMAT(1H0,48M POINT ABOUT WHICH THE MOMENTS ARE TO BE COMPUTED/16M
1 X-COORDINATE = ,F10.4/16M 2-COORDINATE = ,F10.4//
225M REFERENCE CHORD LENGTH = ,F10.4)
1380 FORMAT(1H0,13M MACH NUMBER = ,F10.4)
END

C SUBROUTINE CENTRO
C
C COMPUTES CENTROID COORDINATES OF FINITE ELEMENT,
C TRIANGULAR OR QUADRILATERAL.
C COMMON/PEBATA/ MBODTS,XRIBD(51), RTHETO,THETB(15),CPB(51,15)
C ,XC(3,200),ME(350),NELR(700),ME,MRIBD,PINF,SPURCH,CPB
C ,SIR(20)
C COMMON/ELMCNT/ X(3, 350)
C
C DIMENSION CPB(150)
C DATA IMORD1 / 5MQUAD /
C
C READ FINITE ELEMENT GRID SPECIFICATION FROM DIM OUTPUTS.
C
DO 200 I= 1, ME
READ (5, 901) ICHECK, ID, IE, MD, ME1, ME2, ME3, ME4, 9M
NELR(I) = IE
ME4 = ME4
C
C SEARCH GRID POINT NUMBER
ICOUNT = 1
DO 110 K=1, MRIBD
IF ( ME(K) .EQ. ME1 ) GO TO 101
IF ( ME(K) .EQ. ME2 ) GO TO 102
IF ( ME(K) .EQ. ME3 ) GO TO 103
IF ( ME(K) .EQ. ME4 ) GO TO 104
GO TO 110
101 J1 = K
102 J2 = K
103 J3 = K
104 J4 = K
109 IF ( ICOUNT .EQ. 4 ) GO TO 150

```

IF (ICOUNT .EQ. 3 .AND. ICHECK .NE. IWORD1) 60 TO 111
 ICOUNT = ICOUNT + 1

110 CONTINUE

C TRIANGULAR ELEMENT

111 CONTINUE

90 120 J = 1, 3

XC(J, I) = (X(J, I1) + X(J, I2) + X(J, I3)) / 3.0

120 CONTINUE

60 TO 200

C 150 CONTINUE

C QUADRILATERAL ELEMENT

X1 = (X(1, I1) + X(1, I2) + X(1, I3)) / 3.0

X2 = (X(2, I1) + X(2, I2) + X(2, I3)) / 3.0

X3 = (X(3, I1) + X(3, I2) + X(3, I3)) / 3.0

X4 = (X(4, I1) + X(4, I2) + X(4, I3)) / 3.0

X5 = (X(5, I1) + X(5, I2) + X(5, I3)) / 3.0

X6 = (X(6, I1) + X(6, I2) + X(6, I3)) / 3.0

S12 = SORT((X(1, I1) - X(1, I2)) ** 2 + (X(2, I1) - X(2, I2)) ** 2

, (X(3, I1) - X(3, I2)) ** 2)

S23 = SORT((X(1, I2) - X(1, I3)) ** 2 + (X(2, I2) - X(2, I3)) ** 2

, (X(3, I2) - X(3, I3)) ** 2)

S31 = SORT((X(1, I3) - X(1, I1)) ** 2 + (X(2, I3) - X(2, I1)) ** 2

, (X(3, I3) - X(3, I1)) ** 2)

S34 = SORT((X(1, I3) - X(1, I4)) ** 2 + (X(2, I3) - X(2, I4)) ** 2

, (X(3, I3) - X(3, I4)) ** 2)

S41 = SORT((X(1, I4) - X(1, I1)) ** 2 + (X(2, I4) - X(2, I1)) ** 2

, (X(3, I4) - X(3, I1)) ** 2)

ST123 = 0.5 * (S12 + S23 + S31)

ST341 = 0.5 * (S31 + S34 + S41)

A123 = SORT(ST123 + (ST123 - S12) * (ST123 - S23) + (ST123 - S31)

A341 = SORT(ST341 + (ST341 - S31) * (ST341 - S34) + (ST341 - S41)

AT = A123 + A341

XC(1, I) = (X1 + A123 + X2 + A341) / AT

XC(2, I) = (X2 + A123 + X3 + A341) / AT

XC(3, I) = (X3 + A123 + X4 + A341) / AT

C 200 CONTINUE

901 FORMAT (A5, A3, S10, F8.0, F8.4)

RETURN

END

C SUBROUTINE CONVTC(X, N6, NGRIDP, NSYM, ISTM)

C CONVERT GRID POINT COORDINATES TO THE RECTANGULAR COORD.

C DIMENSION X(3, NGRIDP), M6(NGRIDP), ISTM(20)

DR = 0.0174532925

90 250 I = 1, NGRIDP

C READ OUTPUT OF BING FINITE ELEMENT GRID POINT COORDINATE

C READ (5, 901) CHECK, IOUN, M6(I), NS, X1, X2, X3, M0

C R = X1

C THETA = X2


```

C SINGULAR MATRIX
WRITE (6,1130)
C -----
C      GOTO 1130
C      RETURN
C -----
1130 FORMAT (29H ERROR THE MATRIX IS SINGULAR)
END
C SUBROUTINE SINVERT ( MROWS, M, IRI, IR2, SCALE, DET, MDETXP)
C -----
C -----
C -----
COMMON/MATRIX/ C1(115,115)
INTEGER M1(115), M2(115), M3(115)
C -----
C      IRI = 0
C      IR2 = 0
C      SCALE = 0.0
C      DET = 0.0
C      MDETXP = 0.0
C      CALL INVERT(GOTON,M1,M2,M3,MROWS)
C      IF (GOTON.EQ.2) GO TO 1000
C -----
C -----
C      RETURN
C -----
END
C SUBROUTINE SINP(R,X,T,M,IER)
DIMENSION X(M),T(M)
M=0.0
IF(M.GT.1) GO TO 1000
IER=2
RETURN
1000 IF(X(1).EQ.X(2)) GO TO 1110
M1=M-1
IF(M.EQ.2) GO TO 1120
IF(X(1).LT.X(2)) GO TO 1020
C TEST FOR X TO BE MONOTONICALLY DECREASING
DO 1010 I=2,M1
IF(X(I-1).GE.X(I)) GO TO 1110
1010 CONTINUE
GO TO 1040
C TEST FOR X TO BE MONOTONICALLY INCREASING
1020 DO 1030 I=2,M1
IF(X(I-1).LE.X(I)) GO TO 1110
1030 CONTINUE
1040 M2=M-2
IF(MOD(M,2).EQ.0) GO TO 1130
P=0.0
M1=1
1050 S1=X(M1+1)-X(M1)
S2=X(M1+2)-X(M1+1)
S3=X(M1+1)-X(M1+2)
S4=X(M1)-X(M1+1)

```

```

R=(2.-S1**2-S1-S2-S2**2)/S1**V(M1)*(2.-S4**2+S3-S4-S3**2)/S4**V(M)
M1=M1+1
DO 1060 I=M1,M1,2
S1=X(I)-X(I-1)
S2=X(I+1)-X(I)
1060 R=R*(S1-S2)**3/(S1-S2)*V(I)
IF(M-17.5) GO TO 1080
M1=M1+1
DO 1070 I=M1,M1,2
S1=X(I)-X(I-1)
S2=X(I+1)-X(I)
S3=X(I+2)-X(I+1)
S4=X(I+2)-X(I+1)
1070 R=R*((2.-S2**2-S1-S2-S1**2)/S2*(2.-S3**2+S4-S4-S3**2)/S3)*V(I)
1080 R=R/9.*P
1090 K = LEGVAR(R)
IF (K.NE.O) GO TO 1100
IER = 1
RETURN
1100 IER=3
1110 RETURN
1120 IER=4
RETURN
C TRAPEZOIDAL RULE FOR M=2
1120 R=(X(2)-X(1))*(Y(1)+Y(2))/2.0
GO TO 1090
C FIT POLYNOMIAL THRU FIRST 3 POINTS AND INTEGRATE FROM X(1) TO X(2).
1130 S1=X(2)-X(1)
S2=X(3)-X(1)
S3=Y(2)-Y(1)
S4=Y(3)-Y(1)
P=S1/6.*(2.-S3-6.*Y(1))*(S2**2+S3-S1**2-S4)/(S2*(S2-S1))
M1=2
GO TO 1050
END
OVERLAYCHANG, 7, 1)
PROGRAM INVT
C .....
C FOR MING ONLY CASE, STORES MATRIX (A) AND (A) INVERSE ON TAPE
C MAXIMUM SIZE MATRIX INVERSION =110
C .....
COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEG,NTAPEH,
1 NTAPED,NTAPED,NTAPED,NTAPED,NTAPED,NTAPED,NTAPED,NTAPED,NTAPED,NTAPED,
COMMON/MATRIX/ AMM(115,115)
C
NDEREN=115
C
READ (A) MATRIX INTO CORE, WRITE ON TAPE
CALL FSP(1,NTAPEA,IRR)
DO 1000 J=1,NMING
WRITE (NTAPEA) (AMM(I,J),I=1,NMING)
1000 CONTINUE
CALL FOR EOF (NTAPEE)
REWIND NTAPED
C
INVERT MATRIX (A)
CALL SINVT(NDEREN,NMING,IRR1,IRR2,SCALE,DET,NDERTP)
IF (IRR1)1010,1020,1010
1010 WRITE (NTAPEG,1040) IRR1,IRR2,SCALE
REWIND NTAPED

```



```

CALL FOR EOF (MTAPE)
REWIND MTAPE
C -----
C SECTION EELS
C 69 79 1223
C -----
C -----
C END OF FILE ON MTAPE
1040 WRITE (MTAPE,1040) MTAPE,J,NBND,MINS
C -----
C STOP
C -----
C 1223 CONTINUE
C -----
1050 FORMAT(1M1,3MERROR IN INVERSION OF BODY MATRIX,5H,
1610001 -13,5H,61002 -13,5H,7HSCALE -612.0)
1060 FORMAT(1M0,3M-----END OF FILE DETECTED ON UNIT ,13,3M J- ,13,
1,5H,NBND,13,5H,MINS,13)
END
C -----
C OVERLATCHING, 7, 3)
C PROGRAM INVER
C -----
C INVERT REDUCED MATRIX
C MAXIMUM SIZE MATRIX INVERSION -110
C -----
C CORROM/ MAIN / MTAPE,MTAPE,MTAPE,MTAPE,MTAPE,MTAPE,MTAPE,MTAPE,
1 MTAPE,NBND,MINS,INCH,3,M,SCALE,SCALE,100
C -----
C CORROM/MATRIX/ AFUN(115,115)
C -----
C NDEGEN=115
00 1000 J=1,MING
READ (MTAPE) (ARNU(1,J),I=1,M,NB)
1000 CONTINUE
CALL FOR EOF (MTAPE)
C -----
C MATRIX INVERSION
CALL SIVRTCHDEGEN,MING,1001,1002,SCALE,DET,MDETSP)
IF (1001)1010,1020,1010
1010 CONTINUE
WRITE (MTAPE,1040) 1001,1002,SCALE
REWIND MTAPE
C -----
C -----
C STOP
C -----
C -----
1020 CONTINUE
00 1030 J=1,MING
WRITE (MTAPE) (ARNU(1,J),I=1,M,NB)
1030 CONTINUE
CALL FOR EOF (MTAPE)
REWIND MTAPE
C -----
C -----
C -----
1040 FORMAT(1M1,3MERROR IN INVERSION OF REDUCED MATRIX,5H,
1610001 -13,5H,61002 -13,5H,7HSCALE -612.0)

```

```

END
OVERLAY(MANG, 7, 4)
PROGRAM MDATE
.....
C MAXIMUM SIZE MATRIX INVERSION =112
C MUST BE 2 PLUS MAXIMUM NUMBER OF WING PANELS(110)
C .....
C
COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPEF,NTAPEE,NTAPEF,NTAPEI,
1 NTAPED,NBODY,NWING,INACH,STR,KACE,MPOLAR,IRW
COMMON/MATRIX/ MW(115,115)
C
C DIMENSION XBAR(210),AREA(210)
C
C MW=NWING*1
C MW=NWING*2
C
C DRAG MINIMIZATION OPTION ELIMINATED BY SETTING DRIN = 0.0 HERE
C CAN BE RETORED BY PLACE THE STATEMENT - READ (LI,1020) DRIN -
C HERE AND REMOVE DRIN = 0.0.
C
C DRIN = 0.0
C
1000 IF(MW)1000,1000,1020
1000 CALL PSF(1,NTAPEA,IRR)
DO 1010 I=1,NNN
DO 1010 J=1,NNN
1010 MW(I,J)=0
DO 1010 J=1,NNN
1020 MW(I,J)=1
DO 1020 J=1,NNN
1020 MW(I,J)=1
C
C READ (NTAPEE) NBODY,NWING,INACH,STR,KACE
C MPANEL=NBODY*NWING
C READ (NTAPEE) (J, XBAR(I), DRT, DR2, DR3, DR4, DR5, DR6, DR7,
1, AREA(I), DUMNT6, DUMNT7, DUMNT8, I=1, MPANEL)
C REWIND NTAPES
DO 1030 J=1,NWING
READ (NTAPEE) (MW(I,J), I=1, NWING)
1030 CONTINUE
REWIND NTAPES
DO 1040 I=1, .N6
K=1
II=1-NBODY
DO 1040 J=1,K
JJ=J-NBODY
MW(I,J)=MW(I,J)+AREA(II)-MW(J,I)+AREA(JJ)
MW(J,I)=MW(I,J)
1040 CONTINUE
DO 1050 J=1,NWING
JJ=J-NBODY
MW(J,NN)=AREA(JJ)
MW(NN,J)=AREA(JJ)
K=-XBAR(JJ)+AREA(JJ)
MW(I,NN)=K
MW(NN,J)=K
1050 CONTINUE
MW(NN,NN)=0.0
MW(NN,NN)=0.0
MW(NN,NN)=0.0
MW(NN,NN)=0.0
CALL PSF(1,NTAPEA,IRR)
DO 1060 J=1,NNN
1060 WRITE (NTAPEE)(MW(I,J), I=1,NNN)

```

```

CALL FOR EOF (NTAPEA)
C
C INVERT DRAG MINIMIZATION MATRIX CONSTRAINED FOR CL
CALL SINVRT(MNEM,NM,IRR1,IRR2,SCALE,DET,MDETSP)
IF (IRR1)1070,1080,1070
1070 CONTINUE
WRITE (NTAPEA,1170) IRR1,IRR2,SCALE
REWIN NTAPEA
C .....
C STOP
C .....
C .....
1080 CONTINUE
DO 1090 J=1,NM
DO 1090 WRITE (NTAPEA)(UN(I,J),I=1,NM)
CALL FOR EOF (NTAPEA)
REWIN NTAPEA
CALL FSP(1,NTAPEA,IRR)
DO 1100 J=1,NM
1100 READ (NTAPEA)(UN(I,J),I=1,NM)
CALL FSP(2,NTAPEA,IRR)
C
C INVERT DRAG MINIMIZATION MATRIX CONSTRAINED FOR CL AND CM
CALL SINVRT(MNEM,NM,IRR1,IRR2,SCALE,DET,MDETSP)
IF (IRR1)1070,1110,1070
1110 CONTINUE
DO 1120 J=1,NM
1120 WRITE (NTAPEA)(UN(I,J),I=1,NM)
CALL FOR EOF (NTAPEA)
REWIN NTAPEA
C .....
C PROGRAM EXITS
GO TO 1200
C .....
C .....
1130 DO 1140 J=1,NM
1140 WRITE (NTAPEA)(UN(I,J),I=1,NM)
CALL FOR EOF (NTAPEA)
DO 1150 J=1,NM
1150 WRITE (NTAPEA)(UN(I,J),I=1,NM)
CALL FOR EOF (NTAPEA)
GO TO 1110
C
1200 CONTINUE
C
C FORMATION OF DRAG MINIMIZATION MATRIX
1160 FORMAT(7F10.0)
1170 FORMAT(1H1,6HERROR IN INVERSION OF DRAG MINIMIZATION MATRIX
1,6HIRR1 =,13,5H,6HIRR2 =,13,5H,7HSCALE =,812.6)
END
C
OVERLAY(WANG,7,5)
PROGRAM WANG75
C
C
COMMON / MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPEF,NTAPEG,NTAPEH,NTAPEI,
1 NTAPEJ,NTAPEK,NTAPEL,NTAPEM,NTAPEN,NTAPEO,NTAPEP,NTAPEQ,NTAPER,
COMMON / BODYSP / BODY(51),BZ(51),BRX(51),BRZ(51),BRX(51),BRZ(51)
COMMON / AYAR / A(210),ACB(21),ABX(100),ABY(110),AREA(210),ARB(2),A
1 ART(20),ALPHA(210),ALPHA(210),ALPHA(210),ALPHA(210),ALPHA(210),ALPHA(
2 ALPHA(110),ALPHA(110),ALPHA(110),ALPHA(110),ALPHA(110),ALPHA(
3 210,2),ALPHA(2),ALPHA(2),ALPHA(2),ALPHA(2),ALPHA(2),ALPHA(2),ALPHA(2),

```



```

1000 SUM=SUM+.5*(XT(1)-XT(1-1))*(VT(1)+VT(1-1))
27=SUM
RETURN
END

C
SUBROUTINE SUBROD(NBS,NT,NSTOR,IN,VM,IN,CPC,VEL,ARA,XP,YP,RFAREA,
1 XB,A,THETB,ZCAR,ZC,TA,UA,V,EM,CPU,MST,MSE,DST,DSE)
C
C COMPUTES SINGULARITY STRENGTHS, VELOCITY COMPONENTS AND PRESSURES
C .....
C .....
C .....
COMMON/MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPEF,NTAPEI,NTAPEJ,NTAPEK,
1 NTAPED,NTAPEH,NTAPEL,NTAPEM,NTAPEN,NTAPEO,NTAPEP,NTAPEQ,NTAPER,
COMMON / BOTSF,BOBSF(51),PZDI(51),ORZDI(51)
COMMON/MATRIX/ VB(115,115)
COMMON/MINGOT/ CPU,CPL,MWG,ISOLID,IFORM(10),TAW,BROLL
COMMON/MVAR/ B(210),BCL,BCB,BCCH,BCCH,BCCH,BCCH,BCCH,BCCH,BCCH,
COMMON/INPUT / XT(2,50)
COMMON/POINTS/ TT(4),TTP(4),TTP(4)
COMMON/POINTS/ NPTS,NVAR

C
DIMENSION XB(1),R(1),THETB(1),ZCAR(1),ZC(1),TC(1),
1 ZC(1),TA(1),U(1),V(1),W(1),EM(1),CPU(51,1),OC(51),
2 THETB(22),RR(210),YT(2),BB(51),VB(22),VB(22),
3 BB(22),UC(51),VC(51),RB(51),
4 X(51),ARC(51),OC(51),ZB(51)
DIMENSION A(52),B(52),C(52),X(52),Y(52),Z(52),X(22),Y(22),Z(22)
DIMENSION CP(22),C(52),COST(22),SINT(22),CPU(110),CPL(110)

EQUIVALENCE ( UC(1), RB(1)), ( VC(1), AR(52))

C
CALCULATION OF SINGULARITY STRENGTHS

NP=NV+NBOD
NB=NB5-1
KA=0
PI=3.1415926
NPAW = NV / NMG
NT1 = NT + 1
NT2 = 2*(NT - 1)
NT = NT
NT3 = NT
EM = NMACH
B2 = 1.0 - EM**2
SINA = SIN(ARA)
COSA = COS(ARA)
TANA = TAN(ARA)
SINY = SIN(YAW)
COSY = COS(YAW)
TANT = TAN(YAW)
A1 = 1.0/SORT( 1.0 + TANA**2 + TANY**2 )
COSB = A1*SORT( 1.0 + TANA**2 )
SINB = SORT( 1.0 - COSB**2 )
ARATAN = ACOS( A1 )
SINAT = SIN(ARATAN)
COSB = COS(BROLL)
SINB = SIN(BROLL)

IF ( TAW .LT. 0.01 .AND. BROLL .LT. 0.01) GO TO 100
NT = NT2
NT3 = NT2 + 1

```

```

IF (ARA .LT. 0.01 .AND. YAW .GT. 0.01) GO TO 90
FROLL = ATAN( TAN(YAW) / TAN(ARA) )
GO TO 150
90 CONTINUE
FROLL = 0.5 * PI
GO TO 150
100 CONTINUE
FROLL = 0.0
150 CONTINUE
DO 1000 M=1,NB
  Q(M)=0.
1000 G(M)=0.
  X1=XB(1)
  DO 1010 M=1,NBS
    1010 XB(M)=XB(M)-X1
    THETB(M)=THETB(M)/57.2957795
    IF (N .GT. NT) THETB(M) = THETB(M) - NT * 1 ) * PI
    CPU(M) = COS(THETB(M))
    CPL(M) = SIN(THETB(M))
    COST(M) = COS(THETB(M)*FROLL)
    SINT(M) = SIN(THETB(M)*FROLL)
1020 CONTINUE
    DO 1040 M=1,NB
      ARC(M) = ARAYAW - B2X(M)*COS(FROLL)
      RB(M)=(R(N+1)+R(M))/2.
      RB2(M)=RB(M)*2
      ZB(M)=(ZCAN(M+1)+ZCAN(M))/2.
      X(M)=(XB(M+1)+XB(M))/2.
      IF(ARC(M))1030,1040,1050
1030 KA=1
1040 CONTINUE
      DO 1060 M=1,NB
        EX(M)=XB(M)
        EL=XB(NBS)-XB(M)
        91=SORT((EX+EL)*2*B2*RB2(M))
        92=SORT((EX*2*B2*RB2(M))
        US=1/91-1/92
        VS=B2*RB(M)=(1/(91*(EX-EL*91))-1/(92*(EX*92)))
        VB(M,M)=VS-9RBX(M)*US
1060
1070
      DO 1080 M=1,NB
        EX(M)=XB(M)
        EL=XB(NBS)-XB(M)
        91=((EX-EL)*2*B2*RB2(M))*1.5
        92=((EX*2*B2*RB2(M))*1.5
        UR=B2*RB(M)*(1./92-1./91)
        VR=((EX-EL)*2*B2*RB2(M)/91-EX*(EX*2*B2*RB2(M))
          1/92)/RB2(M)
1080 VB(M,M)=VR-9RBX(M)*UR
1090 CALL SIMVRT(51,NB,11,12,SC,DET,NDETNP)
      IF(11)1100,1110,1120
1100 WRITE (NTAPE0,1590) 11,12,SC
1110 IF(KA)1150,1160,1170
1120 DO 1130 M=1,NB
      Q(M)=0.
      DO 1130 M=1,NB
        1130 Q(M)=Q(M)+VB(M,M)*9RBX(M)
        IF(KA)1170,1170,1160
1140 KA=-1
      GO TO 1070
1150 DO 1160 M=1,NB
      GC(M)=0.

```

```

DO 1160 M=1,NB
1160 QC(M)=QC(M)-VD(M,M)*ARC(M)
1170 CONTINUE
C
C VELOCITY COMPONENTS ON THE BODY
C
DO 1210 M=1,NB
U(M)=0.
UC(M)=0.
V(M)=0.
VC(M)=0.
W(M)=0.
DO 1190 N=1,NB
EX(N)=X(N)-XB(M)
EL=XB(MBS)-XB(M)
XL=EX-EL
D1=SQRT(XL**2+EL**2)*QB2(M)
D2=SQRT(EX**2+EL**2)*QB2(M)
D3=D1**3.
D4=D2**3.
US=1./D1-1./D2
VS=D2*QB(M)*(1./D1*(XL*D1)-1./D2*(EX*D2))
U(M)=U(M)+US*QB(M)
V(M)=V(M)+VS*QB(M)
IF(KA)1180,1190,1180
VS=D2*QB(M)*(1./D4-1./D3)
1180 VR=(XL*(XL**2+EL**2)*QB2(M))/D3-EX*(EX**2+EL**2)*QB2(M)/D4/QB2(M)
WR=(XL/D1-EX/D2)/QB2(M)
UC(M)=UC(M)+VR*QB(M)
VC(M)=VC(M)+VS*QB(M)
W(M)=W(M)+WR*QB(M)
1190 CONTINUE
DO 1200 L=1,MNT
UB(L)=U(M)+UC(M)*COST(L)
VB(L)=V(M)+VC(M)*COST(L)
1200 WB(L)=W(M)+SINT(L)
C
CALL FORMON(MBS,XB,NB,KNACH,COSA,SINA,COSB,SINB,UB,VB,WB,CPU,
1 CPL,CPB,M,MNT,CPC)
IF ( MNT .GT. MT ) CPB(M, MT) = CPB(M, 1)
C
1210 CONTINUE
C
IF ( BROLL .LT. 0.01 ) GO TO 600
C
C TRANSFORM CPB FROM FLOW COORDINATE TO BODY COORDINATE TO
C INCORPORATE THE EFFECT OF ROLL (BROLL)
C
NVAR = 2
NPTS = MT3
DO 500 I=1,NB
DO 510 L=1,MT3
TX(I,L) = TMEIB(L)
TX(2,L) = CPB(I, L)
510 CONTINUE
CALL SPLFIT
DO 520 L=1,MNT
TY(I) = TMEIB(L) - BROLL
IF ( TY(I) .LT. 0.0 ) TY(I) = 2.0*PI + TY(I)
IF ( TY(I) .GT. 2.0*PI ) TY(I) = TY(I) - 2.0*PI
CALL POINT(I)
CPB(I, L) = TY(I)
520 CONTINUE
500 CONTINUE

```

```

C 600 CONTINUE
C
C VELOCITY COMPONENTS ON THE WING
IF(NM)1300,1300,1220
1220 IF(INSTOR)1250,1240,1230
1230 MS=1
JN=2
SY(1)=1.
SY(2)=-1.
60 TO 1250
1240 MS=NBOD+1
JN=1
SY(1)=0.
1250 CONTINUE
L=0
C
C DO 1290 N=MS,MP
C
L=L+1
U(L)=0.
V(L)=0.
W(L)=0.
EN(L)=0.
DO 1290 J=1,JN
DTC=YC(N)-SY(J)*TM
DZC=ZC(N)-ZM
IF(DTC.EQ.0. AND DZC.EQ.0.) 60 TO 1260
THETA=ATAN2(DTC,DZC)
60 TO 1270
1260 THETA=0.
1270 COSTM=COS(THETA)
SINTM=SIN(THETA)
R2=(YC(N)-SY(J)*TM)**2+(ZC(N)-ZM)**2
R(L)=SQRT(R2)
BR2=BR2+R2
VR=0.
VT=0.
DO 1280 M=1,MB
EX=XC(N)-XM-XB(M)
EL=XB(MES)-XB(M)
XL=EX-EL
D1=SQRT(EL**2+BR2)
D2=SQRT(EX**2+BR2)
D3=D1**3
D4=D2**3
U(L)=U(L)+(1./D1-1./D2)*B(M)*B2*BR(L)*(1./D4-1./D3)*COSTM*BC(M)
VR=VR+B2*BR(L)*(1./D1*(XL*D1)-1./D2*(EX*D2))*B(M)*(XL*ENL**2+2
1-VR2)/D3-EX*(EX**2+2*BR2)/D4)/D2*OC(M)*COSTM
1280 VT=VT*(XL/D1-EX/D2)/D2*OC(M)*SINTM
MUS = ( M - MS ) / MBM + 1
THETS = THETA + TA(M)
V(L)=VR+SINTM*VT*COSTM*V(L)
W(L)=VR+COSTM*VT*SINTM*W(L)
EN(L)=EN(L)+VR*COS(THETS)-VT*SIN(THETS)
C
C 1290 CONTINUE
C
1300 CONTINUE
IF(EN)1390,1380,1370
1370 CPSTAG = 8000*((1.-0001)**3.5-1.)
60 TO 1390
1380 CPSTAG=1.
1390 CONTINUE
BCL=0.

```

```

BCD=0.
BCN=0.
IF(MS1)1410,1410,1400
1400 IF(MS2)1410,1410,1420
1410 MS1=1
MS2=MS
1420 MB1=MS2-1-MS1
MBT=MBL+1
M1=MS2-1
MM = 1
DO 1440 M = MS1,M1
MM = MM + 1
RM = RB(M)
DO 1430 M = 1,MMT
CPBB = CPBB(M,M)
CP(M) = CPBB
ARC(M) = CPBB + CPL(M)
YA(M) = CPBB + CPU(M)
YE(M) = YA(M) + CPU(M)
1430 CALL SIMP (CM,THETD,CP,MMT,IRR)
CALL SIMP (AM,THETD,YA,MMT,IRR)
CALL SIMP (BM,THETD,YE,MMT,IRR)
CALL SIMP (ARCM,THETD,ARC,MMT,IRR)
IF (IRR .NE. 1) GO TO 1500
A(MM) = AM + RM
B(MM) = BM + YM
C(MM) = CM + RM
APC(MM) = ARCM + RM
1440 CONTINUE
MM = MS1 - 1
DO 1450 M = 2,MBL
MM = MM + 1
C(M) = B2DX(MM) + A(M) + B2DX(MM) + C(M)
1450 XX(M) = X(MM)
XX(1) = XB(MS1)
XX(MBT) = XB(MS2)
C(1) = .5 + CPSTAG + .951
C(MBT) = .5 + CPSTAG + .952
IF (C(MBT) .NE. 0.) C(MBT) = 0.
A(1) = ZCAR(1)*C(1)
A(MBT) = ZCAR(MBT)*C(MBT)
CALL SIMP (CZ,XX,A,MMT,IRR)
CALL SIMP (CY,XX,ARC,MMT,IRR)
CM=-CM
IF (IRR .NE. 1) GO TO 1500
CALL SIMP (CCX,XX,C,MMT,IRR)
IF (IRR .NE. 1) GO TO 1500
MM = MS1 - 1
DO 1460 M = 2,MBL
MM = MM + 1
ARC(M) = ARC(M) + ( X(MM) + XM - XP )
1460 A(M) = A(M)+(X(MM)+XM-XP+RB(MM)+B2DX(MM)+2B(MM)+C(M)+RB(MM)
+ B2DX(MM)+B(M))
CALL SIMP (B2CMY,XX,ARC,MMT,IRR)
CALL SIMP (CM,XX,A,MMT,IRR)
IF (IRR .NE. 1) GO TO 1500
IF (YAW .LT. 0.01 .AND. BROLL .LT. 0.01 ) GO TO 1465
GO TO 1466
1465 CONTINUE
CCX = 2.0*CCX
CY = 0.0
CZ = 2.0*CZ
1466 CONTINUE
CCY = -CY

```

```

CCZ = -CZ
BDCL = -CCX*SINA*COSE + CCY*(SING*COSE+COSE*SING*SINA) + CCZ*
1 (COSE*COSE - SING*SING*SINA)
BDCB = CCX*COSE*COSE + CCY*(SING*SINA-COSE*SING*COSE) + CCZ*
1 (SING*SING*COSE + COSE*SINA)
BDCD = CCX*SING + CCY*COSE*COSE - CCZ*SING*COSE
BCH = CM + CZ*(2H-2P)
DO 1480 M=1,M5
1480 XB(M)=XB(M)*X1
C
NT = NNT
DO 1490 M=1,NT
1490 THETB(M)=THETB(M)*57.2957795
C
1570 CONTINUE
C PRINT OUT VELOCITY COMPONENTS
C
WRITE(NTAPEC) M5
WRITE(NTAPEC) (XB(J),J=1,M5)
WRITE(NTAPEC) (VC(J),J=1,M5)
C -----
C RETURN
C -----
C
1580 WRITE (NTAPEC,1690) IRR
60 TO 1570
C
C FORMAT STATEMENTS
C
1590 FORMAT (1M1,5X,10N ERROR CODE 1 = 15,10N ERROR CODE 2 = 15,
15N SCALE FACTOR = E16.8)
1690 FORMAT (1ND,-10N INTEGRATION ERROR, IRR //,110)
C
END
SUBROUTINE FORMOM(BODYTS,XB,VC,XHACH,COSE,SINA,COSE,SING,VB,VB,VTB
1 ,COSTB,SINTB,CPB,1,NTMETH,CPALC)
C
C CALCULATION OF PRESSURES, FORCES, AND MOMENTS
C
C DIMENSION XB(1), VC(1), VB(1), VTB(1), COSTB(1), SINTB(1),
1 CPB(1), 1)
C
BODYTS=XB(BODYTS)-XB(1)
XHACH2 = XHACH**2
CPSTAG = 1.42857 * ((1.-2.*XHACH2)**3.5 - 1.)/XHACH2
C
DO 1420 J=1, NTMETH
VPM = VTB(J) + COSTB(J) * VB(J) + SINTB(J)
VPM = VB(J) + COSTB(J) * VTB(J) + SINTB(J)
UWPM = VB(J)*COSE*COSE - VPM*SING*COSE + VPM*SINA
UWPM = 1. + UWPM
VWIM = VB(J)*SING + VPM*COSE
VWIM = -VB(J)*COSE*SINA + VPM*SING*SINA + VPM*COSE
TWIN2 = VB(J)**2 + VPM**2 + VPM**2 - UWPM**2
C
IF (CPALC - 1.) 1370,1380,1390
1370 CPB(1,J) = -2. * UWPM
60 TO 1400
1380 IIR = XHACH2 - 1.
CPB(1,J) = -2.*UWPM + IIR2*UWPM**2 - TWIN2
60 TO 1400
1390 O2 = UWIM ** 2 + VWIM ** 2 + VWIM ** 2

```

```

CPBB(I,J) = 1.42857*((1.+2.*XNACH2*(1.-Q2))**.5-.5-1.)/XNACH2
1600 CONTINUE
1610 CONTINUE
IF(R(MBODY5).EQ.0.) CPBB(MBODY5,J)=CPSTAG
1620 CONTINUE
C
RETURN
END
SUBROUTINE CUBERT
DIMENSION X(3)
COMMON/LOCN/ V(3), L, M, G(4), VMAX, VMIN
COMMON/POINTS/ NPTS, NYAR
L=0
SUM = SQRT( 6(1)**2+6(2)**2+6(3)**2+6(4)**2 )
G1 = 6(1) / SUM
G2 = 6(2) / SUM
G3 = 6(3) / SUM
G4 = 6(4) / SUM
C
EQ. 1 61*X**3 + 62*X**2 + 63*X + 64 = 0.
IF ( ABS( 66 ) .LT. 1.E-6) GO TO 4
C
EQ. 2 61*X**3 + 62*X**2 + 63*X + 64 = 0.
IF ( ABS( 61 ) .GT. 1.E-6) GO TO 9
C
EQ. 3 62*X**2 + 63*X + 64 = 0.
IF ( ABS( 62 ) .GT. 1.E-6) GO TO 7
C
EQ. 4 63*X + 64 = 0.
IF ( ABS( 63 ) .GT. 1.E-6) GO TO 5
RETURN
C
4 CONTINUE
V(1) = 0.0
L=1
RETURN
5 CONTINUE
V0 = - 6(4) / 6(3)
GO TO 35
7 CONTINUE
R1 = 6(3)**2 - 4.0*6(2)*6(4)
IF ( R1 .GT. 0.0 ) GO TO 107
WRITE (6,901)
1 EQUATION 62.X**2 + 63.X + 64 = 0.
RETURN
107 CONTINUE
R1=SQRT(R1)
V1 = ( -6(3) + R1 ) / 0.5 / 6(2)
V(1) = V1
IF ( V1 .GT. VMIN .AND. V1 .LT. VMAX ) GO TO 8
V2 = (-6(3) - R1) / 0.5 / 6(2)
V(2) = V2
IF ( V2 .GT. VMIN .AND. V2 .LT. VMAX ) GO TO 8
WRITE (6,902) VMIN, V1, V2, VMAX
902 FORMAT (1H0, 'BOTH ROOTS V1, V2 OF QUADRATIC EQ. 62.X**2 + 63.X + 64
1= 0 OUTSIDE OF LIMIT VMIN, VMAX / 1X, 'VMIN =',E14.4, 'V1 =',
2 E14.4, 'V2 =',E14.4, 'VMAX =',E14.4)
RETURN
8 CONTINUE
L = 1
RETURN
C
C THREE ROOTS

```



```

2      E16.4,IX,VMAX =,E16.4)
C
      RETURN
      END
C
      SUBROUTINE POINT( N )
      COMMON/CURFIT/ B(2,4, 50), T( 50), SCALEF(2)
      COMMON/ ENDST/ XFIRST, XLAST, XSAVE( 50)
      COMMON/LOCTR/ V(3), L, N, A(4), VMAX, VMIN
      COMMON/POINTS/ MPTS, MVAR
      COMMON/ POINTP/ X(4), XPC( 4), XPP( 4)
      DIMENSION DX(4)
      N1 = MPTS - 1
      IF (N,NE,0) GO TO 4
      DO 1 I=1,M1
      M=1
      IF ( V(1) ) -LE. T(1+1) ) GO TO 2
1      CONTINUE
      WRITE(6,100)
100  FORMAT(1H0,33H ERROR - PARAMETER EXCEEDS LIMITS)
      RETURN
2      CONTINUE
      DO 300 K=1,3
      P = V(K)
      DX(K) = 1.E10
      IF ( P -LT. VMIN ) -OR. P -GT. VMAX ) GO TO 300
      DX(K) = ABS(((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M))*P + B(1,4,M))
1      SCALEF(1) = XIND
300  CONTINUE
      IF ( DX(1) ) -GT. DX(2) ) GO TO 302
      IDENT = 1
      IF ( DX(1) ) -GT. DX(3) ) IDENT = 3
      GO TO 305
302  CONTINUE
      IDENT = 2
      IF ( DX(2) ) -GT. DX(3) ) IDENT = 3
305  CONTINUE
      P = V(IDENT)
      DO 3 I=1, MVAR
      X(I) = ((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M))*P + B(1,4,M)) * SCALEF(1)
      XPC(I) = ((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M))*P + B(1,4,M)) * SCALEF(1)
      XPP(I) = ((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M))*P + B(1,4,M)) * SCALEF(1)
3      XPP(I) = ((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M))*P + B(1,4,M)) * SCALEF(1)
C
310  CONTINUE
      RETURN
C
4      X(M)=X(M)/SCALEF(M)
      DO 5 I=1,M1
      M=1
      K = 1 + 1
      IF ( XSAVE(K) ) -GE. X(M) ) GO TO 6
5      CONTINUE
      XIND = X(M) * SCALEF(M)
      XPT = XFIRST * SCALEF(1)
      XLT = XLAST * SCALEF(1)
      WRITE (6,101) XIND, XPT, XLT
101  FORMAT (1H0, " THE GIVEN VALUE OF INDEPENDENT VARIABLE ",E12.4,
1      " IS OUT OF RANGE TO BE INTERPOLATED ",E12.4,3X,E12.4)
      RETURN
      DO 7 I=1,4
      A(I)=B(M,1,M)
      A(4)=A(4)-X(M)
      IF ( XFIRST -GE. X(M) ) GO TO 112

```

```

IF ( XLAST .EQ. X(1) ) GO TO 111
VMAX = T(M,1) + 1.0001
VMIN = T(M) + 0.9999
CALL CUBERT
XINH = X(1) * SCALEF(1)
GO TO 103
112 V(1) = 0.0
L=1
GO TO 103
111 V(1) = 1.0
L=1
103 CONTINUE
IF (L.NE.0) GO TO 2
OP = P
WRITE (6,10) OP
10 FORMAT (1X,OP =,E15.8)
DO 8 J=1,2
K = M + J - 1
B1 = B(M,1,K)
B2 = B(M,2,K)
B3 = B(M,3,K)
B4 = B(M,4,K)
P = T(K)
SUM = B1*P**3 + B2*P**2 + B3*P + B4
DIF = SUM - X(M)
8 WRITE (6,9) P,B1,B2,B3,B4,X(M),SUM,DIF
9 FORMAT (1X13,E15.8)
WRITE(6,102)
102 FORMAT(1H0,32H FAILURE IN CUBE ROOT EXTRACTION)
RETURN
END

```

```

6 SUBROUTINE SPLFIT
REAL L,M,MU
COMMON/CURFIT/ B(2,4,50), TC(50), SCALEF(2)
COMMON/ENDPT/ XFIRST, XLAST, XSAVE(50)
COMMON/INPUT / IC(2,50)
COMMON/POINTS/ NPTS, NVAR
DIMENSION M(2,50), SC(50), LC(50), MU(50), PC(50), QC(50)
, UC(50)
IF ( NPTS .GT. 50) NPTS= 50
DO 200 I=1, NVAR
SCALEF(I)=0.
DO 201 J=1,NPTS
DO 201 J=1,NVAR
SCALEF(J)=AMAX1(SCALEF(J),ABS(X(J,1)))
DO 101 J=1,NVAR
IF ( SCALEF(J) .LT. 1.E-10 ) SCALEF(J) = 1.0
101 CONTINUE
DO 202 I=1,NPTS
DO 202 J=1,NVAR
X(J,1)=X(J,1)/SCALEF(J)
DO 203 I=1,NPTS
XSAVE(I) = X(1,1)
203 CONTINUE
XFIRST = X(1,1)
XLAST = X(1,NPTS)
T(1)=0.
S(1)=0.
SUM=0.
L(1)=-2.
MU(1)=0.
L(NPTS)=0.
MU(NPTS)=-2.
D(1)=0.

```



```

COMMON/ POINTS/ NPTS, NVAR
C
DIMENSION XB(1),R(1),THETAB(1),ZDELTA(1),XC(1),YC(1),ZC(1)
1 U(1), V(1), W(1), AN(1), CPB(1), THETA(1)
DIMENSION ALPHA(1), T(1), TB(1), TC(1), TX(1), UB(22), VB(22)
1 YB(22), SY(2), FB(1), SB(1), HB(1), COSTB(22),
2 SINTB(22), CP(150), VA(22), VB(22), A(1), B(1), C(1),
3 C(1), XB(1), CPU(110), CPL(110)
C
EPS = 1.0E-6
PI=3.1415926
XRACH2=XRACH+XRACH
BT2=XRACH2-1.
BETA=SBRT(BT2)
NPN = NVING / NUG
NYAN = 0
NT = NTHETA
NT1 = NT + 1
NT2 = 2*( NT - 1 )
N1=NBODYS-1
X1 = XB(1)
NT3 = NT2 + 1
TANA = TAN(ARA)
COSAR = COS(ARA)
SINAR = SIN(ARA)
TANY = TAN(YAN)
COSY = COS(YAN)
SINY = SIN(YAN)
A1 = 1.0/SBRT( 1.0 + TANA**2 + TANY**2 )
COSB = A1*SBRT( 1.0 + TANA**2 )
SINB = SBRT( 1.0 - COSB**2 )
COSG = COS(BROLL)
SING = SIN(BROLL)
DO 1000 N = 1, NBODYS
1000 XB(N) = XB(N) - X1
C
C INCORPORATE EFFECT OF TAN TO ANGLE OF ATTACK
C
ARATAN = ACOS( A1 )
SINAT = SIN(ARATAN)
ARJ = ARA
IF ( YAN .LT. 0.01 .AND. BROLL .LT. 0.01 ) GO TO 100
NT = NT2
NT3 = NT + 1
IF ( ARA .LT. 0.01 .AND. TAN .GT. 0.01 ) GO TO 90
FROLL = ATAN( TANY / TANA )
GO TO 150
90 CONTINUE
FROLL = 0.5 * PI
GO TO 150
100 CONTINUE
FROLL = 0.0
150 CONTINUE
DO 1010 J = 1,NT3
THETAB(J) = THETAB(J) / 57.2957795
IF ( J .GT. NTHETA ) THETAB(J) = THETAB(J-NTHETA+1) + PI
C
C CPU, CPL BASED ON THETAB BEFORE INCORPORATION OF TAN EFFECT
C
CPU(J) = COS(THETAB(J))
CPL(J) = SIN(THETAB(J))
COSTB(J) = COS(THETAB(J)+FROLL)
1010 SINTB(J) = SIN(THETAB(J)+FROLL)

```

```

C      COMPUTE FIRST AND SECOND DERIVATIVES
C      DO 1015 N = 1, NBOOTS
1015  ALPHA(KN) = ARAYAN - 920X(N)*COS(FROLL)
      DO 115 I=1,M1
115    FB(I) = BRX(I)
      CONTINUE
      FD2 = FB(1) * FB(1)
      BTAN = BETA * FB(1)
      IF (BTAN .LT. 1.0) 60 TO 1020
      WRITE (6,1580)
C      .....
C      STOP
C      .....
C      1020 SBT2 = SORT ((1.-BT2*FD2)
      SL06 = AL06((1.-SBT2)/BTAN)*FD2
C      CALCULATE SOURCE STRENGTHS
C      DO 1150 I=1,M1
C      N=I+1
      VBSUM=0.
      VBSUR=0.
      VBSUM=0.
      VBSUM=0.
      VBSUM=0.
      PHIS = 0.
      PHID=0.
      RBRX = FB(N)*R(N)
      RM2=RM*RM
      RM2=RM*RM
      RBRD=RRBRX
      RS=RM2*ALPHAC(I)
      TRM1 = RB(N)-RB(1)
      BRN=BETA*RM
      BRM2 = BRN*BRN
      RAB1 = SORT((TRM1*TRM1-BRN2)
      IF (BRN.EQ.0.) 60 TO 1030
      UBS1 = -AL06 ((TRM1 + RAB1)/BRN)
      1030  ARAB1 = -.5*(TRM1+RAB1-BRM2*UBS1)
      RUB1 = RAB1
      RTUB1 = -.5*(TRM1+RAB1 + BRM2*UBS1)
      RM1 = RAB1-RBRD*UBS1
      RM1 = RAB1-RBRD*UBS1
      IF (I.NE.1) 60 TO 1050
C      IF I EQUALS 1, DO TANGENT CONE SOLUTION
C      Y11 = FD2/(SBT2 + SL06)
      TC11 = FD2*ALPHAC(1)/((FD2-.5)*SBT2 +.5*BT2*SL06)
      DO 1040 K=1,M1
1040    UB(K) = -SL06*Y11 /FD2
      VB(K) = SBT2*Y11/FB(1)
      VTB(K) = 0.
      IF (RM.EQ.0.) 60 TO 1040
      UB(1)=UB(K)*COSB(K)+SBT2/FB(1)*TC11
      VB(K)=VB(K)*-.5*COSB(K)+((SBT2*BT2*SL06)/FD2*TC11
      VTB(K)=-.5*SINB(K)+((SBT2-BT2*SL06)/FD2*TC11
      1040  CONTINUE
C      CALL FORMON(NBOOTS,IB,S,IMACH,COSAR,SINAR,COSB,SINB,UB,VB,VTB,
      C      CPD,CPL,CPSB,1,NT,CPCALC)

```

```

C  CPBB(1, NT3) = CPBB(1, 1)
1050 CONTINUE
IF (RM.EQ.0) GO TO 1060
R001 = RUB91*TC11/RM
R002 = RAD31*111/RM
R003 = RAB91*TC11/RM2
R004 = RV791*TC11/RM2
1060 GO 1130 J=1,1
TRNJ = XB(M)-XB(J) + BETA*R(J)
TRNJ2 = TRNJ*TRNJ
IF(BRM.EQ.0.) GO TO 1100
F1=SOR1(TRNJ2-BRM2)
F2 = AL06((TRNJ*F1)/BRM)
RAB52 = TRNJ*F1-BRM2*F2
UB52 = 2.*(F1-TRNJ*F2)
RM2=RAB52-R004M-UB52
RAB92 = -(TRNJ2-6.*BRM2)*F1/3.-BRM2*TRNJ*F2)
RV792 = -(TRNJ2 + 2.*BRM2)*F1/3.-BRM2*TRNJ*F2)
RM2=RAB92-R004M-RUB92
C
IF (J.NE.1) GO TO 1080
C
T(J) = (R004X-RM1*TC11)/RM2
TC(J) = -(R03-RM1*TC11)/RM2
1080 R004X = R004X-RM2*TC(J)
R5=RS-BRM2*TC(J)
IF(J.NE.1) GO TO 1090
UB5UM = UB51*711
UBDSUM = R001
V5UM = R002
VDSUM = R003
VTDSUM = R004
1090 CONTINUE
V5UM = UB5UM-UB52*TC(J)
VDSUM = V5UM-RAB52*TC(J)/RM
UBDSUM = UBDSUM + RUB52*TC(J)/RM
VDSUM = VDSUM+RAB92*TC(J)/RM2
VTDSUM = VTDSUM+RV792*TC(J)/RM2
GO TO 1130
1100 IF(L.EQ.J) GO TO 1120
IF (J.NE.1) GO TO 1110
PH19 = TRM1*711
PH19 = 1.5*TRM1*TRM1*TC11
1110 PH19 = PH19*TRNJ2*TC(J)
PH19 = PH19*TRNJ*TRNJ2*TC(J)
GO TO 1130
1120 T(J) = -PH19/TRNJ2
TC(J) = -PH19/(TRNJ*TRNJ2)
C
1130 CONTINUE
C
GO 1140 K=1,NT
UB(K) = UBDSUM-UBDSUM*CO5TB(K)
VB(K) = VDSUM-VDSUM*CO5TB(K)
1140 VTB(K) = VTDSUM-SINTB(K)
C
CALL FORMON(MB00T5,XB,A,XMACH,COSAR,SINAR,COSB,SINB,UB,VB,VTB,
1 CPU,CPL,CPBB,M,NT,CPCALC)
IF ( NT .GT. RTMETA ) CPBB(M,NT3) = CPBB(M, 1)
C
1150 CONTINUE
C
IF ( BROLL .LT. 2.01) GO TO 600

```

```

C      TRANSFORM CPD FROM FLOW COORDINATE TO BODY COORDINATE TO
C      INCORPORATE THE EFFECT OF ROLL (BROLL)
C
      NVAR = 2
      RPTS = RT3
      DO 500 I=1,NBODYS
      DO 510 L=1,RT3
      TX(L,L) = THETAB(L)
      TX(2,L) = CPB(L,L)
      510 CONTINUE
      CALL SPLIT
      DO 520 L=1,RT
      TX(L) = THETAB(L) - BROLL
      IF ( TX(L) .LT. 0.0 ) TX(L) = 2.0*PI + TX(L)
      IF ( TX(L) .GT. 2.0*PI ) TX(L) = TX(L) - 2.0*PI
      CALL POINT(1)
      CPB(L,L) = TX(2)
      520 CONTINUE
      500 CONTINUE
      600 CONTINUE
C      CALCULATION OF VELOCITY COMPONENTS IN THE FIELD (EFFECT OF WING
C      ON BODY)
C
      IF (NWING)1160,1360,1160
      1160 CONTINUE
      IF (HACEL.NE.0) 60 TO 1170
      NS=NBODY+1
      JN=1
      SY(1)=0.
      60 TO 1180
      1170 NS=1
      JN=2
      SY(1)=1.
      SY(2)=1.
      1180 CONTINUE
      NPANEL=NBODY+NWING
      11=0
      DO 1350 I=NS,NPANEL
      11=11+1
      U(11)=0.
      V(11)=0.
      W(11)=0.
      AN(11)=0.
      DO 1350 N=1,JN
      XCN=X(11)-XN
      YCN=Y(11)-YN
      ZCN=Z(11)-ZN
      IF (YCN.EQ.0.) 60 TO 1190
      THETA = ATAN2( YCN, ZCN )
      60 TO 1220
      1190 IF(ZCN)1210,1200,1200
      1200 THETA=0.
      60 TO 1220
      1210 THETA=PI
      1220 CONTINUE
      COSTHA = COS( THETA )
      SINTHA = SIN( THETA )
      R2=TCM*YCN+ZCN
      12(2,LE.99) R2 = 0.
      R1=SORI(R2)
      B1=BETA*R1

```

```

BR2=BT2*BR2
TRM=XCN-XB(NB00Y3)*BETA*B(NB00Y3)
IF(R1.EQ.0.) 60 TO 1240
IF(TRM-BR1) 1240, 1240, 1250
1230 61=TRM-TRM-BR2
TRM0=TRM/BR1
62=ALOG((TRM0-SORT(TRM0-TRM0-1.))
63=SQRT(61)
1240 CONTINUE
US=0.
VS=0.
UD=0.
VD=0.
VT=0.
VTF=0.
VF=0.
IF (TRM-LE.BR1) 60 TO 1250
0001 = 2.*61*63
0002 = R1*63
0003 = 62*BR2
0004 = BR2/61
0005 = 2.*R2
0006 = TC11*SINHA
0007 = TRM*63-BR2*62
0008 = 2.*63
0009 = 3.*61*63
0010 = 3.*63
0011 = 672*COSTHA*TC11
0012 = 672*SINHA*TC11
0013 = 2.*TRM
0014 = TC11*COSTHA
0017 = 2.*BR2
0018 = 672*COSTHA
0019 = 672*SINHA
1250 00 134G J=1, M1
BR=BETA*B(J)
TRM1=XCN-XB(J)*BR
TRM2=TRM1*TRM1
IF(R1.EQ.0.) 60 TO 1270
TRM3=TRM1/BR1
IF(TRM3-LT.1.) 60 TO 1340
C
C VELOCITY DUE TO QUADRATIC SINGULARITIES
F1=SQRT((TRM3-TRM3-1.))
F2=ALOG((TRM3-F1)
F1=BR1*F1
US=2.*(F1-TRM1*F2)*T(J)
VS=(TRM1*F1-BR2*F2)/R1
UD=VS*TC(J)
VS=VS*TC(J)
VD=-((TRM2-6.*BR2)*F1/3.-TRM1*BR2*F2)*TC(J)/R2
VT=-((TRM2-2.*BR2)*F1/3.-TRM1*BR2*F2)*TC(J)/R2
C
C VELOCITY DUE TO TANGENT CONE
IF (J-NE.1) 60 TO 1260
US=US-F2*T11
VS=VS-F1-T11/R1
VD=UD*F1*TC11/R1
V9=VD-.5*(TRM1*F1*BR2*F2)*TC11/R2
VT=VT-.5*(TRM1*F1-BR2*F2)*TC11/R2
1260 CONTINUE
C
C SUMMATION OF VELOCITY COMPONENTS
U(1) = U(1) + US + COSTHA*UD
VF = VF + VS + COSTHA*VD

```

```

VTF = VTF + VT * SIN(TNA)
1270 IF (TRM-ER-0.) 1330, 1330, 1280
1280 XI = XB(MB00TS) - (XB(J) - BETA * R(J)) - BETA * R(MB00TS)
TRMX = TRM * XI
IF (R1) 1330, 1310, 1290
1290 IF (J.NE.1) 60 TO 1300
U(1) = U(1) * T11 + (62 * XI / 63) - 6014 * (63 * XI +
1 (1 - XI * ER2 / 6001)) / R1
VF = VF - T11 * ((TRM * TRMX - ER2) / 6002 + 6014 * (6003
1 + TRMX * 63 - XI * 63 - TRM * (2 * XI * (1 - 6004) / 63))) / 6005
VTF = VTF + 6006 * (6007 * XI * TRM * (2 * XI / 63)) / 6005
1300 CONTINUE
U(1) = U(1) * 2 + T(J) * ((TRMX * 62 - 63 * XI * XI / 6008)
1 - T(C(J)) * ((TRMX * XI) * 63 - 6003 * XI * XI * (1 - XI * ER2 / 6009)) / R1
2 * COSTNA
VF = VF - T(J) * ((TRMX * XI) * 63 - 6003 * TRM * XI * XI / 63) / R1
1 * T(C(J)) * COSTNA - ((TRMX * TRMX - 4 * ER2) * 63 / 3 + TRMX * 6003 * XI * (TRM * XI)
2 * 63 / 3 + XI * XI * TRM * (1 - XI * (1 - 6004) / 6010)) / R2
VTF = VTF + T(C(J)) * SIN(TNA) * ((TRMX * 6017) * 63 / 3.0
1 - TRMX * 6003 * XI * (TRM * XI) * 63 / 3 + XI * XI * TRM * (1 - XI * 6010)) / R2
60 TO 1330
1310 IF (TRM-ER-0.) 60 TO 1330
64 = ALOG(TRMX / TRM)
IF (J.NE.1) 60 TO 1320
U(1) = U(1) * T11 + (XI / TRM - 64)
65 = -(64 * XI * XI / (2 * TRM2)) / 2.
VF = VF + 6011 * 65
VTF = VTF + 6012 * 65
1320 U(1) = U(1) * 2 + T(J) * ((XI * TRMX * 64 - XI * XI * 6013)
1 - TRMX * 64 - XI * (1 - XI * XI / (6 * TRM2))
VF = VF - 6018 * T(C(J)) * 66
VTF = VTF + 6019 * T(C(J)) * 66
1330 CONTINUE
1340 CONTINUE
C
VV(1) = VF * SIN(TNA) * VTF * COSTNA + VV(1)
VV(1) = VF * COSTNA - VTF * SIN(TNA) * VV(1)
THETAS = THETA + THETA(1)
AN1(1) = VF * COS(THETAS) - VTF * SIN(THETAS) * AN1(1)
C
1350 CONTINUE
1360 CONTINUE
C
BCL = 0.
BCCD = 0.
BCCN = 0.
CPSTAG = 1.42857 * ((1.0 - 0.2 * ERACN2) * 3.5 - 1.0) / ERACN2
IF (NS1 - ER-0) NS1 = 1
IF (NS2 - ER-0) NS2 = MB00TS
MBT = NS2 - NS1 + 1
THETA = PI / (MTHETA - 1)
MM = 0
DO 1440 M = NS1, NS2
MM = MM + 1
XBB(MM) = XB(M)
MM = R(M)
DO 1630 M = 1, MT3
CPB = CPBB(M, M)
CP(M) = CPB
TC(M) = CPB * CPL(M)
VA(M) = CPB * CPU(M)
1430 VB(M) = VA(M) * CPU(M)
CALL SIMP(SM, THETAB, CP, MT3, IRR)
CALL SIMP(DM, THETAB, TB, MT3, IRR)
CALL SIMP(AM, THETAB, TA, MT3, IRR)

```

```

CALL SIMP(TCH,THETAB,TC,NT3,IR)
IF(IRR.NE.1) GO TO 1570
A(MH) = AM*RM
D(MH) = DM*RM
G(MH) = GM*RM
SD(MH) = TCR*RM
1460 CONTINUE
MH = 0
DO 1450 N = NS1, NS2
  MH = MH + 1
  C(MH) = D2X(MH)*A(MH) + F9(MH)*G(MH)
  IF (N.EQ. 1) C(MH) = 5*CPSTAG*DS1
  IF (N.EQ. MBODY5 .AND. R(MBODY5).EQ. 0.) C(MH) = 5*CPSTAG*DS2
1450 CONTINUE
CALL SIMP(CX,XB,C,MOT,IRR)
CALL SIMP(CZ,XB,A,MOT,IRR)
CALL SIMP(CY,XB,59,MOT,IR)
CW=-CZ
IF(IRR.NE.1) GO TO 1570
MH = 0
DO 1460 N = NS1, NS2
  MH = MH + 1
  C(MH) = ZDELTA(MH)*C(MH)
  G(MH) = XB(MH) * IN - XP + R(MH)*F9(MH)
  TC(MH) = SD(MH) * ( XB(MH) + XM - XP )
  IF (N.EQ. 1) G(MH) = XB(MH) * IN - XP + DS1/(2.*PI)
  IF (N.EQ. MBODY5 .AND. R(MBODY5).EQ. 0.) G(MH) = XB(MH)*IN-XP*DS2/(2.*PI)
1460 A(MH) = A(MH)*G(MH) + C(MH) * R(MH)*D2X(MH)*B(MH)
CALL SIMP(CM,XB,A,MOT,IRR)
CALL SIMP(DBCRT,XB,TC,MOT,IR)
IF(IRR.NE.1) GO TO 1570
C
IF ( VAN .LT. 0.01 .AND. BROLL .LT. 0.01 ) GO TO 1465
GO TO 1466
1465 CONTINUE
CCX = 2.0*CCZ
CY = 0.0
CZ = 2.0*CZ
1466 CONTINUE
CCX = CX
CCY = -CY
CCZ = -CZ
BBCL = -CCX*SINAR*CO5B + CCY*(SIN6*CO5AR*CO5B+SINB*SINAR) + CCZ*
      (CO5B*CO5AR-SIN6*SINB*SINAR)
BBCH = CCX*CO5AR*CO5B + CCY*(SIN6*SINAR*CO5B+SINB*CO5AR) + CCZ*
      (SIN6*SINB*CO5AR+CO5B*SINAR)
BBCH = CCX*SINB + CCY*CO5B*CO5D -CCZ*SIN6*CO5D
BBCH = CH + CX*(ZN-XP)
C
MTMETA = MT
DO 1470 J=1,MTMETA
1470 THETAB(J) = THETAB(J) + 57.2957795
DO 1475 N = 1, MBODY5
1475 XB(N) = XB(N) * X1
C
1550 CONTINUE
WRITE (MTAPEC) MBODY5
WRITE (MTAPEC) (XB(J),R(J),J=1,MBODY5)
WRITE (MTAPEC) (TC(J),YC(J),J=1,MBODY5)
WRITE (MTAPEC) T11,TCT1
1560 CONTINUE
C
C -----
C RETURN
C -----

```

```

C
C 1370 WRITE(NTAPE9,1490) LRA
C 60 TO 1560
1580 FORMAT (53H***BODY POINT IS OUTSIDE MACH CONE - PROGRAM HALT IN
118M SUBROUTINE KARROR)
1690 FORMAT(2H0,18HINTEGRATION ERROR=12)
END
SUBROUTINE FORMON(MBODYS,IB,R,XMACH,COSA,SINA,COSB,SINB,UB,VB,VTO
,COSTB,SINTB,CPBB,1,MNETHA,CPCALC)
C
C CALCULATION OF PRESSURES, FORCES, AND MOMENTS
C
C DIMENSION RE(1), R(1), UB(1), VB(1), VTO(1), COSTB(1), SINTB(1),
C CPBB(51, 1)
C
C BODYL=IB(MBODYS)-IB(1)
C XMACH2 = XMACH**2
C CPSTAG = 1.42857 * ((1.+2.*XMACH2)**3.5 - 1.)/XMACH2
C
C 90 1420 J=1, MNETHA
C VPM = VTB(J) * COSTB(J) + VB(J) * SINTB(J)
C WPM = VB(J) * COSTB(J) - VTB(J) * SINTB(J)
C UPM = UB(J)*COSB-COSA - VPM*SINB-COSA + WPM*SINA
C
C UWINB = 1. + UPM
C UWINB = UB(J)*SINB + VPM*COSB
C UWINB = -UB(J)*COSB-SINA + VPM*SINB-SINA + WPM*COSA
C
C TWIND2 = UB(J)**2 + VPM**2 + WPM**2 - UPM**2 - UWINB**2
C
C IF (CPCALC - 1.) 1370,1380,1390
1370 CPBB(I,J) = -2. + UPM
C 60 TO 1400
1380 IM2 = XMACH2 - 1.
C CPBB(I,J) = -2.*UPM + IM2*UPM**2 - TWIND2
C 60 TO 1400
1390 Q2 = UWINB**2 + UWINB**2 + UWINB**2 + UWINB**2
C CPBB(I,J) = 1.42857*((1.+2.*XMACH2*(1.-Q2))**3.5-1.)/XMACH2
1400 CONTINUE
1410 CONTINUE
1420 CONTINUE
C
C RETURN
C
C SUBROUTINE CURBET
C DIMENSION X(3)
C COMMON/LOCN/ V(3), L, M, 6(4), VNAB, VMIN
C COMMON/POINTS/ NPTS, NVAR
C L=0
C SUM = 30RT( 6(1)**2+6(2)**2+6(3)**2+6(4)**2 )
C 61 = 6(1) / SUM
C 62 = 6(2) / SUM
C 63 = 6(3) / SUM
C 64 = 6(4) / SUM
C
C EO. 1 61*X**3 + 62*X**2 + 63*X = 0.
C IF ( ABS( 64) .LT. 1.E-6) 60 TO 4
C
C EO. 2 61*X**3 + 62*X**2 + 63*X + 64 = 0.
C IF ( ABS( 61) .GT. 1.E-6) 60 TO 9
C
C EO. 3 62*X**2 + 63*X + 64 = 0.

```

MCDONNELL DOUGLAS ✈

```

V1 = 1.0 / 3.0
Q = A2*V1 - (A1*V1)**2
R = .5*A1*A2*V1 - .5*A3 - (A1*V1)**3
RAD = Q**3 + R*R
IF ( RAD .GT. 0.0 ) GO TO 120
WRITE (6,903)
903 FORMAT (1H0, 'ERROR IN TESTING RADICAL, REDUNDANT TEST')
RETURN
120 CONTINUE
RAD = SQRT( RAD)
A2 = R + RAD
SIGN = 1.
IF ( A2 .LT. 0.0 ) SIGN = -1.
A2 = SIGN * ABS( A2 ) **V1
A3 = R - RAD
SIGN = 1.
IF ( A3 .LT. 0.0 ) SIGN = -1.
A3 = SIGN * ABS( A3 ) **V1
V0 = A2 * A3 - A1*V1
35 CONTINUE
IF ( V0 .GT. VMIN .AND. V0 .LT. VMAX ) L = 1
IF ( L .EQ. 0 ) WRITE (6,905)
905 FORMAT (1H0, 'THE ONLY UNEQUAL ROOT IS OUTSIDE OF VMIN, VMAX / 1X,
1 VMIN =', E16.4, 'X, V =', E16.4, 'X, VMAX =', E16.4)
V(1) = V0
RETURN
40 CONTINUE
DO 45 I=1,3
V0 = X(I) - A1/3.0
X(I) = V0
V(I) = V0
IF ( V0 .LT. VMIN .OR. V0 .GT. VMAX ) GO TO 50
L=1
RETURN
50 CONTINUE
45 CONTINUE
WRITE (6,904) VMIN, V(1), V(2), V(3), VMAX
904 FORMAT (1H0, 'ALL THREE UNEQUAL ROOTS OUTSIDE OF LIMIT VMIN, VMAX /
1 1X, VMIN =', E16.4, 'X, V1 =', E16.4, 'X, V2 =', E16.4, 'X, V3 =',
2 E16.4, 'X, VMAX =', E16.4)
C
RETURN
END
C
SUBROUTINE POINT( N )
COMMON/CURFIT/ B(2,4, 50), T( 50), SCALEF(2)
COMMON/ ENDPT/ IFIRST, ILAST, ISAVE( 50)
COMMON/LOCTM/ V(3), L, M, A(4), VMAX, VMIN
COMMON/POINTS/ NPTS, MYAR
COMMON/ POINTP/ X(4), XP( 4), XPP( 4)
DIMENSION DX(4)
N1 = NPTS - 1
IF (N.NE.0) GO TO 4
DO 1 I=1,N1
N=I
IF ( V(I) .LE. T(I+1) ) GO TO 2
1 CONTINUE
WRITE(6,100)
100 FORMAT(1H0, '33N ERROR - PARAMETER EXCEEDS LIMITS)
2 CONTINUE
DO 300 K=1,3
P = V(K)
DX(K) = 1.E10

```

```

IF ( P -LT. VMIN -OR. P -GT. VMAX ) GO TO 300
DX(K) = ABS(((B(1,1,M)*P - B(1,2,M))*P - B(1,3,M))*P - B(1,4,M))
SCALEF(1) = XIND
300 CONTINUE
IF ( DX(1) -GT. DX(2) ) GO TO 302
IDENTY = 1
IF ( DX(1) -GT. DX(3) ) IDENTY = 3
GO TO 303
302 CONTINUE
IDENTY = 2
IF ( DX(2) -GT. DX(3) ) IDENTY = 3
303 CONTINUE
P = V(IDENTY)
DO 3 I=1, NVAR
X(I) = ((B(1,1,M)*P - B(1,2,M))*P - B(1,3,M))*P - B(1,4,M) * SCALEF(1)
XP(I) = ((B(1,1,M)*P - B(1,2,M))*P - B(1,3,M))*P - B(1,4,M) * SCALEF(1)
3 XP(I) = ((B(1,1,M)*P - B(1,2,M))*P - B(1,3,M))*P - B(1,4,M) * SCALEF(1)
C
310 CONTINUE
RETURN
C
4 X(M) = X(M) / SCALEF(M)
DO 5 I=1, M
M=I
K = I + 1
IF ( XSAVE(K) -GE. X(M) ) GO TO 6
5 CONTINUE
XIND = X(M) * SCALEF(M)
XPT = XFIRST * SCALEF(1)
XLT = XLAST * SCALEF(1)
WRITE (6,10) XIND, XPT, XLT
101 FORMAT (1H0, ' THE GIVEN VALUE OF INDEPENDENT VARIABLE ',E12.6,
1 ' SA, IS OUT OF RANGE TO BE INTERPOLATED ',E12.6,X,E12.4)
RETURN
DO 7 I=1, 4
7 A(I) = B(M,I,M)
A(4) = A(4) - X(M)
IF ( XFIRST -EQ. X(M) ) GO TO 112
IF ( XLAST -EQ. X(1) ) GO TO 111
VMAX = T(M,1) + 1.0001
VMIN = T(M) + 0.9999
CALL CURBT
XIND = X(1) * SCALEF(1)
GO TO 103
112 V(1) = 0.0
L=1
GO TO 103
111 V(1) = 1.0
L=1
103 CONTINUE
IF (L.EQ.0) GO TO 2
OP = P
WRITE (6,10) OP
10 FORMAT (1X40P -,E15.8)
DO 8 J=1, 2
K = M + J - 1
O1 = B(M,1,K)
O2 = B(M,2,K)
O3 = B(M,3,K)
O4 = B(M,4,K)
P = T(K)
SUM = O1*OP + O2*OP + O3*OP + O4
OIF = SUM - X(M)
8 WRITE (6,9) K,P,O1,O2,O3,O4,X(M),SUM,OIF

```

```

9 FORMAT (X13,8E15.8)
WRITE(6,102)
102 FORMAT(1N0,32H FAILURE IN CUBE ROOT EXTRACTION)
RETURN
END

C
SUBROUTINE SPLFIT
REAL L,M,MU
COMMON/CONFIT/ B(2,4, 50), T( 50), SCALEF(2)
COMMON/ENDPT/ XFIRST, XLAST, XSAVE( 50)
COMMON/INPUT / X( 2, 50)
COMMON/POINTS/ NPTS, NVAR
DIMENSION M(2, 50), J( 50), L( 50), MU( 50), P( 50), PC( 50)
1 IF ( NPTS .GT. 50) NPTS= 50
DO 200 I=1, NVAR
DO 200 J=1, NPTS
SCALEF(I)=0.
DO 201 I=1, NPTS
DO 201 J=1, NVAR
201 SCALEF(J)=ARX1(SCALEF(J),ABS(X(J,I)))
DO 101 J=1, NVAR
IF ( SCALEF(J) .LT. 1.E-10 ) SCALEF(J) = 1.0
101 CONTINUE
DO 202 I=1, NPTS
DO 202 J=1, NVAR
202 X(J,I)=X(J,I)/SCALEF(J)
DO 203 I=1, NPTS
XSAVE(I) = X(1,I)
203 CONTINUE
XFIRST = X(1,1)
XLAST = X(1,NPTS)
T(1)=0.
S(1)=0.
SUM=0.
L(1)=-2.
MU(1)=0.
L(NPTS)=0.
MU(NPTS)=-2.
D(1)=0.
D(NPTS)=0.
P(1)=2.
P(NPTS)=0.
U(1)=1.
U(NPTS)=0.
N1=NPTS-1
DO 2 I=2, NPTS
SUM1=0.
DO 1 J=1, NVAR
1 SUM1=SUM1+(X(J,I)-X(J,I-1))**2
S(I)=SQR(SUM1)
2 T(I)=T(I-1)+S(I)
DO 3 I=2, NPTS
3 T(I)=T(I)/T(NPTS)
DO 4 I=2, N1
L(I)=S(I-1)/S(I)+S(I+1)
MU(I)=1.-L(I)
P(I)=MU(I)+Q(I-1)+2.
4 Q(I)=-L(I)/P(I)
P(NPTS)=2.+Q(1.-Q(N1))
DO 7 I=1, NVAR
DO 5 J=2, N1
5 U(J)=6.*(X(I,J-1)-X(I,J))/S(J+1)-X(I,J)-X(I,J-1))/S(J))/(S(J)+
1S(J+1))
3 U(J)=(Q(J)-MU(J)+U(J-1))/P(J)
U(NPTS)=-MU(NPTS)+U(N1)/P(NPTS)

```



```

1      CPL(110), CSHAPE(110)
1      DIMENSION U(150), UBB(150), UBR(100), UBW(150), UWB(110), UWN(110), UWW(
110)
1      DIMENSION V(150), VBB(150), VBR(100), VBW(150), VWB(110), VWN(110), VWW(
110)
1      DIMENSION W(150), WBB(150), WBR(100), WBW(150), WWB(110), WWN(110), WWW(
110), WSCLL(20), WSCLU(20), WSCBL(20), WSCBU(20), WSCW(20), WSC(20), WCL(140)
2      WSC(20), WSCBU(20), WSCBL(20), WSCW(20), WCL(140)

```

INTEGER TITLE

```

-----
1      KONFIG = 1      WING ALONE CONFIGURATION
2      KONFIG = 2      BODY ALONE CONFIGURATION
3      KONFIG = 3      WING-BODY CONFIGURATION
-----

```

```

PI = 3.1415926535898
BETA=SQRT(ABS(XMACH-XMACH-1.))
BETA2=BETA*BETA
BETAM=2.4/1.414214-XMACH**4/BETA**75
ARAXX=ARA
ARA=0.0
RATIO= .75
IF(KONFIG.EQ.1) NR=NRON(1)
IF(KONFIG.EQ.3) NR=NRON(2)

```

```

1000 CONTINUE
1010 CONTINUE
1020 CONTINUE
1030 CONTINUE
1040 CONTINUE
1050 CONTINUE
1060 CONTINUE
1070 CONTINUE
1080 CONTINUE
1090 CONTINUE
1100 CONTINUE
1110 CONTINUE
1120 CONTINUE
1130 CONTINUE
1140 CONTINUE
1150 CONTINUE
1160 CONTINUE
1170 CONTINUE
1180 CONTINUE
1190 CONTINUE
1200 CONTINUE
1210 CONTINUE
1220 CONTINUE
1230 CONTINUE
1240 CONTINUE
1250 CONTINUE
1260 CONTINUE
1270 CONTINUE
1280 CONTINUE
1290 CONTINUE
1300 CONTINUE
1310 CONTINUE
1320 CONTINUE
1330 CONTINUE
1340 CONTINUE
1350 CONTINUE
1360 CONTINUE
1370 CONTINUE
1380 CONTINUE
1390 CONTINUE
1400 CONTINUE
1410 CONTINUE
1420 CONTINUE
1430 CONTINUE
1440 CONTINUE
1450 CONTINUE
1460 CONTINUE
1470 CONTINUE
1480 CONTINUE
1490 CONTINUE
1500 CONTINUE

```

```

CALL DCPI (NWIN,NTAPE,A,ANZ,CL(NS))
REWIN NTAPE
CALL FPF (1,NTAPE,IRR)
CALL DCPI (NBODY,NWIN,NTAPE,A,B,C,CL,UNB,VUB,VUB,UNB)
REWIN NTAPE
WING ALONE OR WING-BODY CONFIGURATION
1320 CONTINUE
IF (KOMP16-2)1330,1340,1340
C
WING ALONE CONFIGURATION
1330 CONTINUE
CALL FPF (1,NTAPE,IRR)
CALL CVEL (NWIN,NWIN,NTAPE,A,B,C,CL,UNB,VUB,VUB,UNB)
REWIN NTAPE
GO TO 1350
C
WING-BODY CONFIGURATION
1340 CONTINUE
CALL FPF (1,NTAPE,IRR)
CALL CVEL (NWIN,NBODY,NTAPE,A,B,C,CL,UNB,VUB,VUB,UNB)
CALL FPF (2,NTAPE,IRR)
CALL CVEL (NWIN,NWIN,NTAPE,A,B,C,CL(NS),UNB,VUB,VUB,UNB)
REWIN NTAPE
WING ALONE OR WING-BODY CONFIGURATION
1345 CONTINUE
CALL FPF(10,NTAPE,IRR)
WRITE (NTAPE) NPAEL
CALL FOR EOP (NTAPE)
REWIN NTAPE
DO 1345 J=1,NWIN
JJ=NBODY
N2=(J-1)/NR+1
VSS(J)=VST(J)-CL(JJ)
PO 1345 I=1,NWIN
II=I-NBODY
N1=(I-1)/NR+1
IF(1-LY,J.AND.N1.EQ.NJ) VSS(J)=VSS(J)+VST(II)-CL(II)
1345 CONTINUE
NUPV = NWIN / NUG
DO 1360 J=1,NWIN
NJS = (J-1)/NUPV + 1
JJ=NBODY
TJ=TNETA(JJ)
COSTJ=COS(TJ)
SINTJ=SIN(TJ)
VX(J)=VSS(J)-COSTJ
VY(J)=VSS(J)-SINTJ
IF(TWICK.EQ.0.) GO TO 1360
WV(J)=WSS(J)-COSTJ
VW(J)=WSS(J)-SINTJ
UNW(J)=UNW(J)-25*CL(JJ)
1360 UNW(J)=UNW(J)-25*CL(JJ)
C
IF (THICK)1370,1410,1370
1370 CONTINUE
SINB = SIN(GBOLL)
COSB = COS(GBOLL)
DO 1380 J=1,NWIN
U(J)=UNB(J)+UNW(J)+UNBT(J)+UNBT(J)+UNW(J)+UNW(J)
VJ =VUB(J)+VUB(J)+VUBT(J)+VUBT(J)+VUB(J)+VUB(J)
WJ =WUB(J)+WUB(J)+WUBT(J)+WUBT(J)+WUB(J)+WUB(J)
V(J) = VJ+COSB - WJ*SINB
W(J) = WJ+SINB + VJ*COSB
1380 CONTINUE

```



```

1465 CONTINUE
WCPM = (WCMPL - WCMPL) / CBAR
WCRM = (WCMRL - WCMRL) / CBAR
WCTM = (WCMVL - WCMVL) / CBAR
IF (KONFIG.EQ. 1) GO TO 1540

C
C WING-BODY CONFIGURATION
CALL CVEL (NBOY,NBOY,NTAPED,A,B,C,CL,UBB,VBB,WBB)
CALL F3F (2,NTAPED,IRR)
CALL CVEL (NBOY,NBOY,NTAPED,A,B,C,CL(NS),UBV,VBV,WBV)
REMBD NTAPED
DO 1470 J=1,NBOY
1470 UBB(J)=UBB(J)-.25*CL(J)
C
C COMPUTE COEFFICIENT OF PRESSURE ON BODY WITH OR WITHOUT THE
C EFFECT OF WING THICKNESS
C
NBB = NBOY
NRB = NROB
NTA = NTHETA
IF (BROLL.LT. 0.01 .AND. TAN .LT. 0.01) GO TO 1488
NTA = 2*NTHETA
NRB = 2 * NBOY
NRB = 2 * NROB

C
DO 1485 J = 1, NTHETA
THETAB(J,NTHETA) = THETAB(J) + 180.0
1485 CONTINUE
1488 CONTINUE
C
3300 CONTINUE
C
DO 1490 J = 1, NROB
DO 1490 I = 1, NTHETA
IJ = J + (I-1)*NROB
U(IJ) = UBB(IJ) + VBB(IJ) + WBB(IJ)
V(IJ) = VBB(IJ) + VBB(IJ) + WBB(IJ)
W(IJ) = WBB(IJ) + WBB(IJ) + WBB(IJ)
1490 CONTINUE
C
CALL CP(NBB,NRACN,CPCALC,V,V,V,CPB,ARA,TAN)
C
C COMPUTE COEFFICIENTS OF LIFT, DRAG, MOMENT ON BODY PANELS
C
IF (BROLL.LT. 0.01 .AND. TAN .LT. 0.01) GO TO 1520
DO 3000 I=1, NTHETA
IR = NTHETA - I + 1
IR1 = (IR-1)*NROB
IR2 = (I + NTHETA - 1)*NROB
DO 3000 J=1, NROB
IJ = IR1 + J
J1 = IR2 + J
CPB(IJ) = CPB(IJ)
3000 CONTINUE
C
1520 CONTINUE
C
CALL BLDR(ARA,TAN,BROLL,BRXC,AREA,CPD,XP,IBAR,NROB,
1 NTHETA,NTA,THETAB,RFAREA,CBAR,BUCL,BUCB,BUCB,BUCB,BUCB)
C
C BODY ALONE OR WING-BODY CONFIGURATION
1530 CONTINUE
BBCL=BBCL/RFAREA
BBDB=BBDB/RFAREA

```

```

      DBCT = DBCT / RFAREA
      DBCM = DBCM / (RFAREA * CBAR)
      DBCYM = DBCYM / (RFAREA * CBAR)
      IF (KONFIG - EQ. 2) GO TO 1540
      WING-BODY CONFIGURATION
      BODY COEFFICIENTS
      DBCM = DBCM / CBAR
      DBCL = DBCL * DBCL
      DBCB = DBCB * DBCB
      BCT = BCB
      BCL = BCL
      BCM = DBCM * DBCL
      BCYM = DBCYM * BCL
      MISSILE COEFFICIENTS
      DBCL = DBCL * BCL * XNNCLT
      DBCB = DBCB * BCB * XNNCBT
      BCT = BCT * BCB
      DBCM = DBCM * BCB * XNNCBT
      DBCYM = BCB * BCL
      DBCYM = DBCYM * BCL
      ANY CONFIGURATION
      WRITE OUTPUT
      1540 CONTINUE
      ARA=ARAX
      IF (KONFIG-2) 1630, 1560, 1550
      WING-BODY CONFIGURATION
      1550 CONTINUE
      WRITE (NTAPE0, 1770)
      GO TO 1570
      BODY ALONE CONFIGURATION
      1560 CONTINUE
      WRITE (NTAPE0, 1820)
      BODY ALONE OR WING-BODY CONFIGURATION
      1570 CONTINUE
      WRITE (NTAPE0, 2010) DBCB, DBCL, DBCT, DBCM, DBCBM, DBCYM
      WRITE (NTAPE0, 1830)
      WRITE (NTAPE0, 1860) (THETA(I), I=1, NTHETA)
      WRITE (NTAPE0, 1870)
      IF (NACH.LT.1.) GO TO 1590
      NBS = NBOBYS
      DO 1580 I=1, NBOBYS
      XIGB(I) = XB(I)
      1580 WRITE (NTAPE0, 1880) XB(I), (CPDB(I, J), J=1, NTHETA)
      GO TO 1610
      1590 NBS = NBOBYS - 1
      DO 1600 I=1, NBS
      XS = S * (XB(I) - XB(I+1))
      XIGB(I) = XS
      1600 WRITE (NTAPE0, 1890) XS, (CPDB(I, J), J=1, NTHETA)
      1610 IF (KONFIG - EQ. 2) GO TO 1750
      WING-BODY CONFIGURATION
      WRITE (NTAPE0, 1790)
      WRITE (NTAPE0, 2010) DBCB, DBCL, DBCBM, DBCM, DBCYM
      WRITE (NTAPE0, 1840)
      CALL OUTB (NTAPE0, NBS, NTA, NROUB, THETA0, CPB)

```

```

C      WRITE (NTAPEO,1830)
C      CALL OUTB (NTAPEO,MBO,MTA,MROVB,TMETAB,ALPHAB)
C
C      WING ALONE OR WING-BODY CONFIGURATION
C      1630 CONTINUE
C
C      IF (NACEL .EQ. 0) GO TO 1680
C
C      WING ALONE OR WING-BODY CONFIGURATION WITH NACELLE(S)
C      DO 1670 J=1,NACEL
C      NIN=INTN(J)
C      NIN=NIN+1
C      WRITE (NTAPEO,1890) J
C      WRITE (NTAPEO,2010) INNCB(J),INNCB(J),INNCB(J)
C      WRITE (NTAPEO,1900)
C      WRITE (NTAPEO,1860) (TMETAB(I,J),I=1,NTM)
C      WRITE (NTAPEO,1870)
C      IF (INNCB(I,1)) GO TO 1650
C      DO 1640 I=1,NIN
C      1640 WRITE (NTAPEO,1880) IN(I,J),(CPHN(I,K,J),K=1,NTM)
C      GO TO 1670
C      1650 NIS=NIN-1
C      DO 1660 I=1,NIS
C      XX=.5*(IN(I,J)+IN(I+1,J))
C      1660 WRITE(NTAPEO,1880) XX,(CPHN(I,K,J),K=1,NTM)
C      1670 CONTINUE
C
C      1680 CONTINUE
C      DO 1690 J=NS,MPANEL
C      1690 CL(J)=CL(J)
C
C      IF (KONFIG=2)1700,1710,1710
C
C      WING ALONE CONFIGURATION
C      1700 CONTINUE
C      WRITE (NTAPEO,1810)
C      GO TO 1720
C
C      WING-BODY CONFIGURATION
C      1710 CONTINUE
C      WRITE (NTAPEO,1780)
C
C      WING ALONE OR WING-BODY CONFIGURATION
C      1720 CONTINUE
C      WRITE (NTAPEO,2010) MCB, MCL, MCB, MCB, MCB, MCB, MCB, MCB
C      WRITE (NTAPEO,1980)
C      WRITE (NTAPEO,2000) (I,I=1,NCOLW)
C      WRITE (NTAPEO,1760) (MSCB(I),I=1,NCOLW)
C      WRITE (NTAPEO,1990)
C      WRITE (NTAPEO,2000) (I,I=1,NCOLW)
C      WRITE (NTAPEO,1760) (MSCB(I),I=1,NCOLW)
C      WRITE (6,1991)
C      WRITE (6,2000) (I,I=1,NCOLW)
C      WRITE (6,1760) (MSCB(I),I=1,NCOLW)
C      WRITE (NTAPEO,1910)
C      CALL OUTM (NTAPEO,MWING,NCOLW,MROW,CPU)
C      WRITE (NTAPEO,1920)
C      CALL OUTM (NTAPEO,MWING,NCOLW,MROW,CPL)
C      WRITE (NTAPEO,1930)
C      CALL OUTM (NTAPEO,MWING,NCOLW,MROW,CL(MB))
C      IF (TRICK)1730,1740,1730
C      1730 WRITE (NTAPEO,1950)
C      CALL OUTM (NTAPEO,MWING,NCOLW,MROW,ALPHAB)
C      WRITE (NTAPEO,1960)
C      CALL OUTM (NTAPEO,MWING,NCOLW,MROW,ALPHAL)

```



```

C
C
C      DIMENSION Z(1)
C      WRITE (NTAPEO,1010) (I,I=1,NCOLW)
C      WRITE (NTAPEO,1020)
C      DO 1000 J=1,NROW
C      1000 WRITE (NTAPEO,1030) (J,Z(I),I=J,NVING,NROW)
C
C      RETURN
C
C      1010 FORMAT(1H0,2X,14HSPANSWISE STATION,19,9110/(18X,18,9110))
C      1020 FORMAT(1H ,2X,17HCHORDWISE STATION)
C      1030 FORMAT(1H ,110,F19.5,9F10.5/(F29.5,9F10.5))
C      END
C
C      SUBROUTINE OUTB (NTAPEO,NBODY,NTNETHA,NROW,NTNETHA,Z)
C
C      .....
C      BODY OUTPUT FORMAT
C      .....
C
C      DIMENSION Z(1),TNETHA(1)
C
C      WRITE (NTAPEO,1010) (TNETHA(I),I=1,NTNETHA)
C      WRITE (NTAPEO,1020)
C      DO 1000 J=1,NROW
C      1000 WRITE (NTAPEO,1030) (J,Z(I),I=J,NBODY,NROW)
C
C      RETURN
C
C      1010 FORMAT(1H0,1X,11HNTNETHA(DEC.),12F10.2/(13X,12F10.4))
C      1020 FORMAT(1H0,4X,7HROW NO.)
C      1030 FORMAT(1H ,18,3X,12F10.4/(13X,12F10.4) )
C      END
C
C      SUBROUTINE OLDN(ARA,TAN,BROLL,DRDZC,AREA,CPS,IP,XBAR,NROW,
C      1  TNETHA,NTA,THETA,RFAREA,CBAR,BUCL,BUCD,BUCB,BUCM,BUCYN)
C
C      COMPUTES LIFT, DRAG, BANK, PITCH AND BANKING MOMENTS FROM
C      THE PRESSURE COEFFICIENT CPS
C
C      DIMENSION DRDZC(1), AREA(1), CPS(1), XBAR(1), TNETHA(1),
C      1  COST(20), SINT(20)
C
C      TANA = TAN(ARA)
C      COSA = COS(ARA)
C      SINA = SIN(ARA)
C      TANT = TAN(TAN)
C      A = 1.0 / SORT( 1.0 + TANA**2 + TANT**2 )
C      COSB = A * SORT( 1.0 + TANA**2 )
C      SINSB = SORT( 1.0 - COSB**2 )
C      SING = SIN(BROLL)
C      COSG = COS(BROLL)
C      IJ = 0
C      NBB = NTA*NROW
C
C      DO 300 I=1, NTA
C      COST(I) = COS( TNETHA(I) )
C      SINT(I) = SIN( TNETHA(I) )
C
C      DO 250 J=1, NROW
C      IJ = IJ + 1
C      JI = IJ
C      IF ( 1.67. NTNETHA ) JI = (NTA-I)*NROW + J

```

```

1 = CDB(JJ) * AREA(JJ)
FXI = F*DBDX(J)
SINO = SORT( 1.0 - DBDX(J)**2 )
FYI = F*SINO*SINT(J)
FZI = -F*SINO*COST(J)
FX = FX + FXI
FY = FY + FYI
FZ = FZ + FZI
XMP = XMP + FZI*(ZBAR(JI) - XP )
XMY = XMY + FYI*(YBAR(JI) - XP )
250 CONTINUE
C
FY = -FY
FZ = -FZ
300 CONTINUE
IF ( NTA .EQ. NTHETA ) FY = 0.0
BUCL = (-FX*SINA*CSDB + FY*(SING*CSDB+CSG*SINO*SINA) + FZ*
1 CSG*CSA -SING*SINO*SINA) /RFAREA
EUCB = (FX*CSA*CSDB + FY*(SING*SINA-CSG*SINO*CSA) + FZ*
1 (SING*SINO*CSA + CSG*SINA) ) /RFAREA
UBCB = (FZ*SINO + FY*CSG*CSDB - FZ*SING*CSDB) /RFAREA
UBCH = XMP / RFAREA / CBAR
UBCYM = -XMY / RFAREA / CBAR
RETURN
END
C
SUBROUTINE WLNH(NH,NROW,NP,ZP,RFAREA,AREA,ZBAR,YBAR,ALPHAN,
1 THETAN,CPM,SCL,SCB,CL,CB,CB,CMP,CHE,CNY,ARA)
C
C.....COMPUTES COEFFICIENT OF LIFT, DRAG, AND MOMENT ON WING
C SPANWISE DISTRIBUTION OF LIFT AND DRAG COEFFICIENTS
C
COMMON/INIGOT/ CPU, CPL,NWG,ISOLIP,IFORM(10),YAW,BROLL
1 DIMENSION AREA(1),ZBAR(1),YBAR(1),ALPHAN(1),THETAN(1),CPM(1)
1 ,SCL(1),SCB(1),SCB(1),YBAR(1)
1 DIMENSION ANS(1), CPU(110), CPL(110)
C
PL=J-14159
NCOL=NH/NROW
NWC = NCOL / NWG
NDRAG=0.0
WLIFT=0.0
WBANK = MPH = WRM = WYM = 0.0
C
J=0
COST = COS(TAU)
SINT = SIN(TAU)
CSA = COS(ARA)
SINA = SIN(ARA)
TANA = TAN(ARA)
TANT = TAN(TAU)
A = 1.0/SORT( 1.0 + TANA**2 + TANT**2 )
CSB = A* SORT( 1.0 + TANA**2 )
SIB = SORT( 1.0 - CSB**2)
CSG = COS(BROLL)
SING = SIN(BROLL)
C
DO 1010 K=1,NCOL
NWS = (K-1)/NWC + 1
DRAG=0.0
XLIFT=0.0
BANK = PMOM = NROW = WROW = 0.0

```

```

C
SUN=0.0
CON=1.0
DO 1000 I=1,NROW
  J=J+1
  IF(1.0-1) TEST=ABS(YBAR(J))*ABS(ABS(TAN(J))-5.0)
  IF(TEST.LT.0.001) CON=5
  SUN=AREA(J)+SUN
  SINAP= SIN(ALPHAR(J))
  COSAP= COS(ALPHAR(J))
  SINT = SIN(THETAR(J))
  COST = COS(THETAR(J))
  F=CPM(J)*AREA(J)
  FX = F*COST*SINAP
  FY = F*SINT
  FZ = F*COST*COSAP
C
C SIGN CONVENTIONS ARE FOR POSITIVE PRESSURE ON THE LOWER
C WING SURFACE, SIGN OF UPPER SURFACE FORCES ARE REVERSED IN
C THE PROGRAM FCALC
C
  XL = -FX*SINA+COSB + FY*(SING*COXA+COSG*SINB*SINA) + FZ*
    1 (COSG*COXA -SING*SINB*SINA)
  XD = FX*COXA+COSB + FY*(SING*SINA-COSG*SINB*COXA) + FZ*
    1 (SING*SINB*COXA + COSG*SINA)
  XB = FX*SINB + FY*COG*COGB - FZ*SING*COGB
  BANK = BANK + XB
C
  XLIFT=XLIFT+XL
  DRAG=DRAG+XD
  PRON = PRON + XL*(ZBAR(J)-XP) + XD*(ZBAR(J)-ZP)
  RRON = RRON + F * COXA + SINT( YBAR(J)-2) + (ZBAR(J)-ZP)*2 )
  YRON = YRON + XB *(ZBAR(J)-XP) - XD*YBAR(J)
1000 CONTINUE
C
  SCL(K) = XLIFT / SUN
  SCB(K) = DRAG / SUN
  SCB(K) = BANK / SUN
  WDRAG=WDRAG+DRAG
  WLIFT=WLIFT+XLIFT
  WBANK = WBANK + BANK
  WPN = WPN + PRON
  WRN = WRN + RRON
  WYN = WYN + YRON
1010 CONTINUE
C
  CL=WLIFT/AREAA
  CD=WDRAG/AREAA
  CB = WBANK / AREAA
  CIP = WPN / AREAA
  CRR = WRN / AREAA
  CY = WYN / AREAA
C
C -----
C RETURN
C -----
C END
C
C SUBROUTINE CAMBER(COLM,NROW,CHORD,CHORDL)
C
C .....
C COMPUTES SURFACE SHAPE, GIVEN SURFACE SLOPES
C .....
C DIMENSION CHORD(1),CHORDL(1)

```



```

SUBROUTINE CONVERK( X, ME)
      CONVERT COORDINATE X, Y, Z INTO X AND THETA FOR SURFACE
      FIT PURPOSES.
      DIMENSION X( 3, ME)
      RD = 57.2957795
      DO 100 I = 1, ME
        R = SQR( X(2, I)**2 + X( 3, I)**2 )
        ANG = ACOS ( X( 3, I) / R ) * RD
        IF ( X( 2, I) .GT. 0.0 ) X( 2, I) = ANG
        IF ( X( 2, I) .LT. 0.0 ) X( 2, I) = 360.0 - ANG
      100 CONTINUE
      RETURN
      END
      OVERLAY(ANG, 7, 11)
      PROGRAM PSUM
      THIS SUBROUTINE INTERPOLATES THE PRESSURE COEFFICIENTS CPB (
      EFFECT OF WING ON THE BODY ) AT THE BODY STATIONS AND ADDS TO
      THAT OF THE BODY ALONE CPBD.
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI, LO, MBOBY, MWING, XNACH,
      SYN, XACE, MPOLAR, IRN
      1 COMMON/ NYAR / NYH(9), NYH(12), NS, MPANEL, MACEL, MROUB, MROUB,
      1 NCOLV, MTHETA, MTHETA, MLE, MLE, MPOLR, MCL
      1 COMMON /YVAR / TITLE(20), THETA(210), THETA(15), THETA(15), THETA(
      1 11), TGL(11), TGL(11), TGL(11), THICK, TWIST
      1 COMMON /XVAR / XBAR(210), X(210), X(12), XNCH(2), XNCH(2), X(11), X(
      1 51, 2), XNCH(2), XNCH(2), XNCH(2), XNCH(2), X(11), X(
      1 11), XNCH(2), XNCH(2), XNCH(2), XNCH(2), X(11), X(
      1 XCB(3, 700), X(350), X(19(51), MTHETA, THETA(15), CPB(51, 15)
      1 COMMON/REPEAT/ INEPT
      DIMENSION XADD(115), YADD(115), CF(115), CPB(150), FC(115, 2)
      INEPT = 1
      CDR = 57.2957795
      MTHETA = MBOBY / MROUB
      CHECK THE TOTAL NUMBER OF POINT OF CPB AND DIVIDE INTO 2
      SEGMENTS IF IT EXCEEDS 110
      LOOP = 0
      NSTART = 1
      NSEG = 1
      NCL1 = 1
      NCL2 = MTHETA
      NCOL1 = NCL2 - NCL1 + 1
      NCL21 = 1
      NCL22 = MTHETA
      NCOL2 = NCL22 - NCL21 + 1
      NBO = MBOBY
      IF ( MBOBY .LT. 110 ) GO TO 70
      NSEG = MBOBY/110 + 1
      NCOL1 = MTHETA/NSEG + 1
      NCOL2 = MTHETA/NSEG

```

C SCALE XC AND THETAB FOR BETTER SURFACE FIT

C 50 CONTINUE

MCL1 = 1
MCL2 = MCOL1 + MCL1 - 1
MCL21 = 1
MCL22 = MCOL2 + MCL21 - 1
MCD = MCOL1 * MROVB
GO TO 70

C 60 CONTINUE

MCL1 = MCL2 - 1
MCL2 = MTHETA
MCOL1 = MCL2 - MCL1 + 1
MCL21 = MCL22 + 1
MCL22 = MTHETA
MCL2 = MCL22 - MCL21 + 1
MCD = MCOL1 * MROVB
MSTART = MROVB * (MCL1 - 1) + 1

C 70 CONTINUE

CALL SCALE(MROVB, XC, FNC(1,1))
DTB = THETB(MCL22) - THETB(MCL21)

JJ = 1
DO 80 J=MCL1, MCL2
FNC(J,2) = (THETAB(J) - THETB(MCL21)) / DTB
JJ = JJ + 1

C 80 CONTINUE

DO 100 I=1, MCOL1
DO 100 J=1, MROVB
IJ = J * (I-1) + MROVB
XADD(IJ) = FNC(J,1)
YADD(IJ) = FNC(J,2)

C 100 CONTINUE

C CALL SURFIT(XADD, YADD, MCD, CPB(MSTART), CP)

C SCALE XMI0B AND THETB FOR CORRECT INTERPOLATION

IC = 1
DXC = XC(MROVB) - XC(1)
DO 105 I=1, MROVB
IF (XMI0B(I) .LT. XC(1) .OR. XMI0B(I) .GT. XC(MROVB)) GO TO 105
IF (IC .EQ. 1) MDS = I - 1
FNC(I,1) = (XMI0B(I) - XC(1)) / DXC
IC = IC + 1

C 105 CONTINUE

MXT = IC - 1
DTB = THETAB(MTHETA) - THETAB(1)
DO 110 I=MCL21, MCL22
FNC(I,2) = (THETB(I) - THETB(MCL21)) / DTB

C 110 CONTINUE

DO 120 J=MCL21, MCL22
DO 120 I=1, MXT
SUMB = CF(1) + CF(2) + FNC(1,1) + CF(3) + FNC(J,2)
DO 130 K=1, MDS
K3 = K+3
RI = (FNC(I,1) - XADD(K)) ** 2 + (FNC(J,2) - YADD(K)) ** 2
IF (RI .LT. 1.E-5) GO TO 130
SUMB = SUMB + CF(K3) * RI + ALOG(RI)

```

130 CONTINUE
  I1 = 1 + NBS
  CPB(11,J) = SUNB + CPB(11,J)
120 CONTINUE
  LOOP = LOOP + 1
  IF ( NSEG .GT. 1 .AND. LOOP .EQ. 1 ) GO TO 60
  WRITE (6,1830)
  WRITE (6,1860) ( THETA(I), I=1,NTHETS)
  WRITE (6,1870)
  NBS = NBOOTS
  IF ( XNACH .LT. 1.0 ) NBS = NBOOTS - 1
  DO 200 I=1,NBS
    WRITE (6,1880) XNIB(I), ( CPB(11,J), J=1,NTHETS)
200 CONTINUE
C
1830 FORMAT (1H0/2X,"BODY PRESSURE COEFFICIENTS (CP), INCLUDING THE EF-
      ,EFFECT OF WING")
C
1860 FORMAT (1H0,4X,"THETA (DEG)",F14.4,9F10.4/((F29.4,9F10.4) )
C
1870 FORMAT (1H ,9X,"X")
C
1880 FORMAT (1H ,F14.4,F15.5,9F10.5/((F29.5,9F10.5) )
C
END
SUBROUTINE SCALE( MX, FM1, FM2)
C
C      GENERATES FM2 FROM FM1 BY SCALING TO A RANGE BETWEEN 0 AND 1
C
C
C      DIMENSION FM1(1), FM2(1)
C
C      FM2 = FM1(1)
C      FM1 = FM1(1)
C      DO 100 I=2, MX
C        FM1 = AMIN1( FM1, FM1(I) )
C        FM2 = AMAX1( FM2, FM1(I) )
C      100 CONTINUE
C
C      DFN = FM2 - FM1
C      DO 200 I=1, MX
C        FM2(I) = ( FM1(I) - FM1 )/DFN
C      200 CONTINUE
C
C      RETURN
C      END
C
C      OVERLAY(WANG, 7, 12)
C      PROGRAM WINGPR
C
C      READ IN FINITE ELEMENT, INTERPOLATE PRESSURE AND GENERATES
C      PRESSURE LOAD ON THE WING.
C
C      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LG, LH, LI, LO, NBOOTS, NUNG, XNACH,
C      STM, XACE, MPOLAR, IAW
C      COMMON/CORTR/ NSTA, NTRANS(20), ISTN(20), VA(3,2,20), STN(9,2,20)
C      COMMON/WVAR / WFA(6), NROW(2), NS, MPANEL, MACEL, MROW, MROW,
C      MCOL, MTHETA, MTHETS, MLE, MGG, MPOLS, MCLX
C      COMMON/PDATA/ MBOOTS, MIBO(51), MTHETS, MTHETS(13), CPB(31,13)
C      , XCOL(3,700), MCL(350), MCL(700), ME, MCL(50), PINT, IPUNCH
C      , CPB
C      2

```

```

COMMON/ELMENT/ XC(3, 350)
COMMON/WINGDT/ CPU, CPL, NUG, ISOLID, IFORM(10), YAW, BROLL
COMMON/XVAR/ XBAR(210), XC(210), XN(2), XNXX(2), ENTH(2), XB(51), XN(
1 51,2), XNMC(2), XNCL(2), XNCH(2), XYZ(3), XCL(11), XCB
2 (11), XNCEL, XP, XCPBAR
COMMON/YVAR/ YBAR(210), YC(210), YN(2)
COMMON/ZVAR/ ZBAR(210), ZC(210), ZN(2), ZDELTA(51), ZON(51,2), ZP, ZA
DIMENSION CPU(110), CPL( 110)
DIMENSION CPB(150)

C
CALL WINGO( XC, YC)
C
END
C
SUBROUTINE WINGO( XC, YC)
C
C INTERPOLATE OUTPUT FOR MASTRAN USE, GRID POINTS AND ELEMENT
C LOCATIONS ARE GENERATED FROM PING.
C
COMMON/ MAIN / LA, LB, LC, LP, LE, LF, LI, LO, NOBODY, NUNG, INACH,
1 STM, KACE, NPOLAR, INW
COMMON/COEFFB/ CF(100, 4)
COMMON/CORTRN/ MSYM, MTRANS(20), ISYM(20), VA(3,2,20), BTM(9,2,20)
COMMON/MVAR / MFT(6), MROW(2), MS, MPANEL, KACEL, MROWB, MROWU,
1 MCOLV, MINEYA, MTHEIS, MILE, MRG, MPOLR, RELX
COMMON/PCOORD/ X(210), Y(210), ZPT
COMMON/WINGDT/ CPU, CPL, NUG, ISOLID, IFORM(10), YAW, BROLL
COMMON/PBDATA/ MBOYS, XMB(51), MTHEIS, YMB(15), CPB(51,15)
1 , XC(3,300), MEL(350), MELN(20), ME, MNEIDP, PINF, IPUNCH
2 CPB
COMMON/REPEAT/ IREPT
DIMENSION CPB(150)
DIMENSION CPU(110), CPL(110), XC(1), YC(1)
DIMENSION MELN(6)
DIMENSION COF(115)

C
PCOV = 0.5 + 1.4 * PINF * INACH * INACH
MADJ = 0
IF ( INACH .GT. 2.5 ) MADJ = MBODY
IREPT = 0

C
NW = 1
NPT = ( MPANEL - MBODY ) / NUG
10 CONTINUE

C
STORE PRESSURE AT CONTROL POINT AND ITS LOCATION FOR
INTERPOLATION.
C
NC1 = NPT
NC3 = NC1 + 3
C
IF ( ISOLID .EQ. 0 ) GO TO 205
C
SOLID WING CASE , LIFTING PRESSURE
C
DO 200 J=1,NC1
JA = J + ( NW-1)*NPT
COF (J) = ( CPU(JA) - CPU(JA) ) * PCOV
JB = JA + MBODY - MADJ
XC(J) = XC(JB)
YC(J) = YC(JB)
200 CONTINUE

C
WRITE (6,803) ( X(1), Y(1), COF(1), I=1,NC1)

```

```

C      CALL SURFITE X, Y, NC1, COP, CFC(1, NW, 1) )
C
C      GO TO 250
C
C 203 CONTINUE
DO 210 J = 1, NC1
  JA = J + (NW-1)*NPT
  COP(J) = CPU(JA) * PCOVY
  JB = JA + NBOBT - NADJ
  XC(J) = XC(JB)
  YC(J) = YC(JB)
210 CONTINUE
C
C      WRITE (6,801) ( XC(I), YC(I), COP(I), I=1, NC1 )
C
C      CALL SURFITE X, Y, NC1, COP, CFC(1, NW, 1) )
C
C
C 215 J = 1, NC1
  JA = J + (NW-1)*NPT
  COP(J) = CPL(JA) * PCOVY
215 CONTINUE
C
C      WRITE (6,802) ( XC(I), YC(I), COP(I), I=1, NC1 )
C
C      CALL SURFITE X, Y, NC1, COP, CFC(1, NW, 2) )
C
C
C 250 CONTINUE
C
C      COMPUTES PRESSURE AT DESIRED LOCATION ( CENTROID OF ELEMENT ),
C      FOR EACH COMPLETE WING.
C
C      CALL PWING( NW )
C
C      NW = NW + 1
C
C
C      IF ( NW .LE. NWS ) GO TO 10
C
C 801 FORMAT (1H0, ' PRESSURE AT CONTROL POINT - UPPER SURFACE = / 15X,
1    'X', Y, 'X', 'PRESSURE = / (11X, F8.4, 2X, F8.4, 2X, E12.4) )
C 802 FORMAT (1H0, ' PRESSURE AT CONTROL POINT - LOWER SURFACE = / 15X,
1    'X', Y, 'X', 'PRESSURE = / (11X, F8.4, 2X, F8.4, 2X, E12.4) )
C 803 FORMAT (1H1, ' TOTAL LIFTING PRESSURE ON THE WING AT THE
1    'CONTROL POINT = / 15X, 'X',
1    'X', Y, 'X', 'PRESSURE = / (11X, F8.4, 2X, F8.4, 2X, E12.4) )
C
C      RETURN
C      END
C
C      SUBROUTINE PWING( NW )
C
C      INTERPOLATES PRESSURE ON WING AT GIVEN LOCATION X, Y FOR
C      LOCATION XC, YC ON CENTROID OF ELEMENT
C
C      COMMON/COEFFB/ CFC(100, 4, 2)
C      COMMON/CONTRM/ MSTW, MTRANS(20), ISYM(20), VAL(3, 2, 20), STW(9, 2, 20)
C      COMMON/PSDATA/ NBOBT, EN100, 511, MTHET6, THET6(15), CPB(51, 15)
C      COMMON/PCOORD/ X(210), Y(210), MP
C      COMMON/ELACMT/ XI(3, 350)
C      COMMON/WING60/ CPX, CPL, NW6, ISOL10, IFORM(10), YAW, DBOLL

```

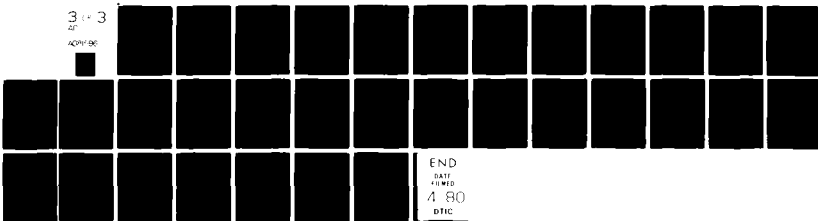
AD-A081 596

MCDONNELL DOUGLAS ASTRONAUTICS CO HUNTINGTON BEACH CA F/6 9/2
AERODYNAMIC COMPUTER CODE FOR COMPUTING PRESSURE LOADS ON A MIS--ETC(U)
DEC 79 K K WANG N00014-75-C-0462
MDC-68223 NL

UNCLASSIFIED

3 3

ADP/90



END

DATE

4 80

DTIC

DIMENSION CPU(110), CPL(110), CPB(150), P(210)
INTEGER CHECK, WORD1

DATA WORD1/ SMCQUAD/

WRITE TITLE OF WING PRESSURE OUTPUT

IF (MW .GT. 1) GO TO 50

WRITE (6,901)

WRITE (6,900) MWG

CONTINUE

WRITE (6,910) MW

MPUNCH = 0

MSID = IPIX(SID(MSYN))

MS = 2

IF (ISOLID .NE. 0) MS = 1

DO 150 K=1,MS

CHECK SIGN OF SID(MSYN) FOR PRESSURE OUTPUT SIGN AS DESIRED.

ASID = ABS(SID(MSYN))

PSIGN = SID(MSYN)/ ASID

MSID = IPIX(ASID)

READ IN FINITE ELEMENT SPECIFICATIONS FROM PING OUTPUT

READ IN NUMBER OF GRID (NG) AND ELEMENT SPECIFICATION (NE)
CARDS.

READ (5,902) NGRIDP, NE

CALL COMVT(XI, NG, NGRIDP, MSYN, ISYN)

CALL COORDT(XI, NGRIDP)

CALL CENTRO

DO 100 IE = 1,NE

CHANGE THE SIGN OF Y COORDINATE OF THE LEFT WING TO REFLECT
THE SYMMETRY OF WING ARRANGEMENT

TCOORD = ABS(XCB(2, IE))

SUM = CFC(1,MV,K) + CFC(2,MV,K)*XCB(1,IE) + CFC(3,MV,K)*TCOORD

DO 110 J = 1, MP

J3 = J + 3

RI = (XCB(1,IE) - X(J))**2 + (TCOORD - T(J))**2

IF (RI .LT. 1.E-9) GO TO 110

SUM = SUM + CFC(J3, MV, K) * RI * ALOG(RI)

110 CONTINUE

P(IE) = SUM * PSIGN

100 CONTINUE

MSYN = MSYN + 1

IF (ISOLID .EQ. 0 .AND. K .EQ. 1) WRITE (6,903)

IF (ISOLID .EQ. 0 .AND. K .EQ. 2) WRITE (6,904)

WRITE (6,909)

WRITE (6,906) (HELM(J), XCB(1,J), XCB(2,J), P(J), J=1,NE)

PUNCH PRESSURE AND GRID POINT IDENTIFICATION IN THE FORM OF
BULK DATA FORMAT OF NASTRAN

[illegible]

```

PROGRAM WRITTP
C
C
C      WRITE ON TAPE FOR GRAPHIC REPRESENTATION BY 3 D PLOTTING
COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI, LO, MBOOT, MWING,
1  XWACH, SYM, KACE, MPOLAR, IAW, IPLOT
COMMON /TVAR/ ,TITL(20),THETA(210),THETAB(15),THETAS(15),THETAM(
1  11,21),TCL(11),TCB(11),TCM(11),THICK,THIST
COMMON /XVAR/ ,XBAR(210),XC(210),XMI(2),XMM(2),XMMN(2),XZ(3),XCL(11),XCD
1  51-2),XMM(2),XMMCL(2),XMMN(2),XZ(3),XCL(11),XCD
2  (11),XMMCL(2),XMMN(2),XZ(3),XCL(11),XCD
COMMON /YVAR/ ,YBAR(210),YC(210),YMI(2),YMM(2),YMMN(2),YZ(3),YCL(11),YCD
1  51-2),YMM(2),YMMCL(2),YMMN(2),YZ(3),YCL(11),YCD
COMMON/PODATA/ MBOOTS,MI(100),MTN(20),THETA(15),CPDB(51,15)
1  ,TCB(3,200),MC(350),HELM(200),ME,MRI(10),PINF,IPURCH,CPB
2  ,SIN(50)
COMMON/WINGOT/ CPU, CPL,MVG,ISOLID,IFORM(10),YAW,ROLL
COMMON/REPEAT/ IREPT
COMMON/ MYAR / MFT(9), MROH(2), MS, MPANEL, MACEL, MROH, MROH, MROH,
1  MCOLM, MTHETA, MTHETS, MILE, MEG, MPOLR, MCLI
C
C      DIMENSION CPB(150), CPU(110), CPL(110)
C
C      MTO = MTHETA
C      MTA = MTHETA
C      IF ( YAW .GT. 0.01 ) MTO = 2*(MTHETA-1)
C      IF ( YAW .GT. 0.01 .OR. ROLL .GT. 0.01 ) MTA = 2*MTHETA
C      IF ( IREPT .EQ. 2 ) GO TO 200
C      REWIND 12
C
C      PRESSURE ON THE WING SURFACE CPU, CPL
C
C      MPWING = (MPANEL - MBOOT)/MVG
C      DO 100 K=1, MVG
C      JA = (K-1)*MPWING
C      JB = JA + MBOOT
C      WRITE (12) ( XC(I+JB), YC(I+JB), CPU(I+JA), I=1,MPWING)
C      WRITE (12) ( XCL(I+JB), YCL(I+JB), CPL(I+JA), I=1,MPWING)
C      100 CONTINUE
C
C      PRESSURE ON THE BODY, BODY ALONE
C
C      200 CONTINUE
C
C      WRITE (12) ( XMI(1), THETA(1), CPDB(1-J), J=1,MTO), I=1,
1  MBOOTS)
C
C      IF ( IREPT .EQ. 2 ) GO TO 300
C
C      PRESSURE ON THE BODY, DUE TO THE PRESENCE OF WING
C
C      DO 250 I=1, MTA
C      IJ = (I-1)*MROH
C      WRITE (12) ( XCL(I), THETAB(I), CPDB(I+J), J=1, MROH)
C      250 CONTINUE
C
C      300 CONTINUE
C
C
C      END
C      OVERLAY(WING, 7, 15)
C      PROGRAM MYSUP
C
C      HIGH SUPERSONIC FLOW ANALYSIS FOR MACH NO. 2.0 - 6.0

```

```

COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI, LO, MBOOT, MNING, XHACHN,
1 SYHN, KACE, NPOLAR, IN
COMMON/CONST / DR, RD, S, GP1, GR1, GR, AV1, AV2, AV3
COMMON/FREES / XHACH, PF, ARBTAN, BETA, MEND
COMMON/FLOW / FLOWF(10, 2), X(51), Y(51)
COMMON / BODYSP / DRDX(51), ZDRX(51), DRDXC(51), ZDRDXC(51), XROOT, XWTIP
COMMON / NVAR / NVAR(12), MPANEL, MACEL, MROUB, NC, %Q
COMMON / PBDATA / MBOOTS, XHIDB(51), MNETB, MNETB(15), CPDB(51, 15)
1 XEB(3, 700), M6(350), MELN(700), ME, MGR10P, P1NF, IPUNCH
2 CPB, S10(20)
COMMON / VAR / BUNTY(210), XC(210), BUNY(6), X0(51)
COMMON / YAR / BUNY2(210), YC(210)
COMMON / MINGOT / CPW, MW6, ISOL10, IFORM(10), YAW, BROLL, SLOPE(51)
1 DIMENSION CPW(110, 2), ALPHA(110, 2), ALPHAT(110), ARNT(110),
2 AREA(210), XCR(40), CPB(150)
DATA DR, RD / 0.0745329232, 57.29577957 /
DATA AV1, AV2, AV3 / 225.0, 8.0382434E-10, 1.0 /
6 = 1.4
GP1 = 6 * 1.0
GR1 = 6 - 1.0
GR = SORT( GP1/GR1 )
READ (11, 800) XHACH, P1NF, POLAR, PUNCH, TRICK, CAMB, TWIST,
1 SOLID, MING, INC, IND, MBOOTS, BTNET, (S10IC), K-1,
2 Z0, (IFORM(K), K=1, 10), INOH, ARB, YAW, BROLL
XHACHN = XHACH
B72 = 0.5-5-XHACH**2
MBOOT = IFIX(MBOOTS)
NPOLAR = IFIX(POLAR)
IPUNCH = IFIX(PUNCH)
ITRICK = IFIX(TRICK)
ICAMB = IFIX(CAMB)
ITWIST = IFIX(TWIST)
ISOLID = IFIX(SOLID)
MNING = IFIX(MING)
NC = IFIX(NC)
NB = IFIX(NB)
MBOOTS = IFIX(MBOOTS)
MNETB = IFIX(MNETB)
TANA = TAN( ARB*DE )
TARY = TAN( YAW*DE )
A1 = 1.0/SORT( 1.0 + TANA**2 + TARY**2 )
ARBTAN = ACOS(A1)*RD
READ (11, 800) (X(K), K=1, MBOOTS)
1 READ (11, 800) (Y(K), K=1, MBOOTS)
2 READ (11, 800) (MNETB(K), K=1, MNETB)
X(1) = Y(1) = 0.0
YF = Y(1)
DO 50 I=2, MBOOTS
XMI00(1) = 0.5*( X(1) + X(I-1) )
YF = Y(I)
Y(1) = 0.5*( YF + YN )
YF = YN
50 CONTINUE
XMI00(1) = 0.0
CALL SECEXP(2)
1 NV2 = 2-MNING

```

```

NC1 = NC - 1
NB1 = NB - 1
NBC = NB1 - NC1
HPANEL = HBODY + NBC
RFAREA = 0.0
CBAR = 1.0
DO 200 I = 1, NW2, 2
  READ (11,800) (XC(K), K=1,NBC)
  READ (11,800) (XCR(K), K=1,NBC)
  READ (11,800) (YC(K), K=1,NBC)
  READ (11,800) (YCR(K), K=1,NBC)
  READ (11,800) (AREA(K), K=1,NBC)
  READ (11,800) (ARUT(K), K=1,NB1)
  READ (11,800) (SLOPE(K), K=1,NBC)
  DO 100 L=1, NB1
    KS = 1 + (L-1)*NC
    KE = KS + NC1
    READ (11,800) (ALPHAT(K), K=KS, KE)
    ALPHAKS(1) = ALPHAT(KS)
    ALPHAKS(2) = -ALPHAT(KE)
    KS1 = KS + 1
    DO 110 K=KS1, KE
      ALPHAK(1) = SLOPE(K-1) + ALPHAT(K)
      ALPHAK(2) = SLOPE(K-1) - ALPHAT(K)
110 CONTINUE
100 CONTINUE
  C
  DO 120 K=1, NBC
    RFAREA = RFAREA + AREA(K)
120 CONTINUE
200 CONTINUE
  C
  NBC1 = NB1 - NC
  DO 350 L=1, 2
    NW6 = L
    DO 300 J=1, NB1
      DO 310 I=1, NC
        IJ = I + (J-1)*NC
        SLOPE(IJ) = -((-1.0)**L*(ALPHAK(IJ, L) - ARUTANDR))
310 CONTINUE
300 CONTINUE
    CALL SECEXP(I)
    DO 320 K=1, NBC1
      CPM(K, L) = CPM(K, L)/BY2
320 CONTINUE
330 CONTINUE
    NW6 = NWING
  C
  CALL FORCEBX, Y, APB, TAN, BROLL, DBCL, DBCO, DBCH, DBCHN, DBCHN,
    DBCTH)
  C
  DBCO = DBCO/RFAREA
  DBCL = DBCL/RFAREA
  DBCH = DBCH/RFAREA
  DBCHN = DBCHN/(RFAREA*CBAR)
  DBCTH = DBCTH/(RFAREA*CBAR)
  C
  WRITE (6,801)
  WRITE (6,802) DBCO, DBCL, DBCH, DBCHN, DBCTH
  WRITE (6,803)
  WRITE (6,804) (THETA(I), I=1, NTHETA)
  WRITE (6,805)
  NB1 = HBODY - 1
  DO 360 I=1, N991

```



```

3      0.10422, 0.063801/
C      BETAB = BETA * 2.9
      BETAR = BETA * 0.9
      ALPHAR = ALPHA * 0.9
      COS9 = COS( BETA )
      SINA = SIN( ALPHA )
      COSA = COS( ALPHA )
      SINB = SIN( BETA )
      CACB = COSA * COSB
      CACB2 = CACB ** 2
      SASB = SINA * SINB
      PHIN = 0.3
      DBETAR = 2.9 * BETAR
      ALPBET = ABS( ALPHA/BETA )
      T = SIN(DBETAR) * TAN( BETAR )
      IM12 = IM1 ** 2
      CALL MCONEC( IM1, BETA, CP, DSC )

C      DSCD SHOCK ANGLE FOR ZERO ANGLE OF ATTACK CASE
C
      DSCD = DSC * 0.9
      SINDS = SIN( DSC )
      CPCORE = ( CP - 1.0 ) / ( 0.3 * 6 * IM12 )

C      CALCULATES THE PRESSURE ON CONE (PCONE), FLOW MACH NO.
C      (XRP), ANGLE OF ATTACK (ALPP) IN MERIDIAN PLANE
C
      DO 200 J=1, NTHETA
      THEN = ( 180.0 - THETA(J) ) * 0.9
      COSP = COS( THEN )
      COS2P = COS( 2.0 * THEN )
      SINP = SIN( THEN )
      DO 100 I = 1, 6
      K = 3 * ( I - 1 ) + 1
      AAC1 = A(K) * A(K) * A(K) * COSP * A(K * 2) * COS2P
      100 CONTINUE
      CPE(J) = ( AAC1 * T + AAC2 * T / IM12 + AAC3 / IM12 ) * ALPBET +
      1      ( AAC4 * T + AAC5 * T / IM12 + AAC6 / IM12 ) * ALPBET ** 2
      CPT(J) = CPE(J) * CPCORE * CPSTAG
      PCONE(J) = 0.5 * 6 * IM12 * CPT(J) + 1.0
      IF ( PCONE(J) .LT. PHIN ) PCONE(J) = PHIN
      XRP(J) = XMI * SORT( COSA ** 2 + ( SINA * COSP ) ** 2 )
      ALPP(J) = ATAN( SINA * COSP / COSA ) * 0.9
      IF ( ABS(ALPPA) .LT. 1.0E-2 ) 60 TO 201
      200 CONTINUE
      201 CONTINUE

C      CALCULATES THE SURFACE MACH NO. (XMS), ( SHOCK ANGLE -
C      DEFLECTION ANGLE ) (THETA)
C
      DO 300 J=1, NTHETA
      DELI = BETAR
      IS = 1
      FPP = 1.0
      XMC = XMP(J)
      60 TO 400
      450 CONTINUE
      XMS = ( XMI * SINDS ) ** 2
      PS = ( 2.0 * 6 * XMS - 6 * IM1 ) / 0.9
      FPP = PS / 2
      XMC = XMP(J)
      DELI = ALPP(J) * 0.9 + BETAR
      DPSK = 0.0

```

```

IF ( DELI .LE. 0.0 ) 60 TO 500
IS = 2
460 CONTINUE
DELI = DELI + RD
XND = XNC - SIN(DELI)
CK = 2.0 * GP1 / (6. + 3.0) * XND
XMS = ( CK + EXP(-CK) ) ** 2
P2 = ( 2.0 * G * XMS - GM1 ) / GP1
50 TO ( 450, 470 ), IS
670 CONTINUE
DPSK = PCONE(J) - P2 * FDP
500 IF ( DPSK .LT. 0.0 ) DPSK = 0.0
PSNK = PCONE(J) - DPSK
IF ( PSNK .LT. 1.0 ) PSNK = 1.0
XN12 = XNIP(J) ** 2
XMS(J) = SORT(XN12 * (GP1 * PSNK + GM1) - 2.0 * GP1 * GM1) / (PSNK * (
    GM1 * PSNK + GP1) )
EN2 = XNIP(J) ** 2
SINBT2 = (GP1 * PSNK + GM1) / (2.0 * G * EN2)
IF ( SINBT2 .GT. 1.0 ) SINBT2 = 1.0
IF ( SINBT2 .LT. 0.0 ) SINBT2 = 1.0 / EN2
SHOCKA = ASIN( SORT(SINBT2) )
THETA(J) = SHOCKA * RD - ALFP(J)
IF ( ABS(ALPHA) .LT. 1.0E-2 ) 60 TO 301
300 CONTINUE
301 CONTINUE
WRITE(6,801)(THETA(J),CPT(J),PCONE(J),XNIP(J),XMS(J),THETA(J),
    ALFP(J),1-1,RTMETB)
801 FORMAT ('NO.201, FLOW FIELD IN THE MERIDIONAL PLANE:///
    1 5X, MERIDIAN CP SHOCK ANGLE ANGLE OF:
    2 3X, M1 M2 CP SHOCK ANGLE ANGLE OF:
    3 5X, ANGLE, SHOCK, ATTACK,/(3X,19.2,3E,6F15.6)')
RETURN
END
SUBROUTINE MCONEC(EM, DCD, PCP1, DSH)
C
C CALCULATES THE CONE PRESSURE AT ZERO ANGLE OF ATTACK
C SECOND ORDER THEORY USED FOR SMALL SIMILARITY PARAMETER
C APPROXIMATE SOLUTION OF HAMMITT AND MURPHY FOR LARGE VALUES
C MIDRANGE GIVEN BY SUITABLE TRANSITION FUNCTION
C
COMMON/CONST /DR, RD, 6, GP1, GM1, GR, AV1, AV2, AV3
LSDET = 0
RC = 0.017453292
DCR = DCD * DR
SINDC = SIN(DCR)
GM12 = 0.5 * GM1
EMSIM = EM * SINDC
EM30 = EM ** 2
KSOL = 2
C
C CALCULATE UPPER TRANSITION POINT, ENSINF
C
ENSINF = 1.4
IF ( EM .GE. 10.0 ) 60 TO 8
ENSINF = 0.325
IF ( EM .GT. 1.5 ) ENSINF = 1.4 - 1.075 * EXP((EM - 1.5) * (-0.8) )
8 IF ( ENSIN .LT. ENSINF ) 60 TO 20
KSOL = 1
C
C CALCULATE CONE SURFACE PROPERTIES USING METHOD OF HAMMITT AND

```



```

31 CONTINUE
CPHM = CP
BCR = ASIN( EMSIN / EM )
GO TO 21

32 CONTINUE
A = ( CPHM - CP - DCP2*DXF ) / DXF**2
CP = CP + DX * ( DCP2 + A*DX )
IF ( ISDET - EQ. 2 ) RETURN
PCP1 = 0.5 * 6 * EMS0 + CP + 1.0
EMNSF = EM * SIN(DSR)
EMNS = ( EMNSF - 1.0 ) * ( EMSIN / EMSINP )**2 + 1.0
EMNS0 = EMS0**2
EMNS0 = (6P1*EM*EMNS)**2 - 6.0*(EMNS0-1.0)*(6*EMNS0 + 1.0)
DSR = ASIN( EMNS / EM )
EMNS0 = EMNS0 / ((2.0*6*EMNS0 - 6P1)*(EM1*EMNS0 + 2.0))
PSPC = (2.0*6*EMNS0 - 6P1)/6P1
EMC = SORT( ( 1.0*6M12*EMNS0 ) * PSPC**2 / (6M1/6) - 1.0 / 6M12 )
TCT1 = ( 1.0*6M12*EMNS0 ) / ( 1.0*6M12*EMC**2 )
40 CONTINUE
RETURN
END
SUBROUTINE COMPC (DANG, CP, SKANG)
C
C COMPUTES THE CONDITION BEHIND THE SHOCK
COMMON/CONST /DR, DR, 6, 6P1, 6M1, 6R, AV1, AV2, AV3
COMMON/FLOW /FLOWA(10, 2), X(51), Y(51)
C
C CHECK THE DEFLECTION ANGLE DANG
XMI = FLOWA(1,1)
XMI2 = XMI**2
IF ( ABS(DANG) .LT. 1.0E-3 ) GO TO 240
IF ( DANG .GT. 55.0 ) GO TO 10
IF ( DANG .LE. 2.0 ) GO TO 260
C
C SET UP CUBIC AND SOLVE FOR SIN(THETA)**2 OR SHOCK ANGLE
SIN0 = SIN(DANG*DR)
COS0 = COS(DANG*DR)
B = -(XMI2 + 2.0)/XMI2 - 6*SIN0**2
C = (2.0*XMI2 + 1.0)/XMI2**2*(6P1**2/4.0 + 6M1/XMI2)*SIN0**2
D = -(COS0 / XMI2)**2
C
C CHECK FOR SHOCK DETACHMENT
CHECK = (-B**2/4.0 + C/3.0)**3 + ( (B/3.0)**3 - (B*C - 3.0*D)
1 / 4.0)**2
IF ( CHECK .GE. 0.0 ) GO TO 10
C
C ATTACHED SHOCK
T = B**2 - 3.0*C
Z = (9.0*B*C - 2.0*D**3 - 27.0*B)/(2.0*D**1.5)
W = ACOS(Z)
T = W/3.0
T = 2.0*SORT(T)
R1 = (T-COS(Z) - 0)/3.0
R2 = -(T-COS(Z) + 60.0*DR) / 0/3.0
R3 = -(T-COS(Z) - 60.0*DR) / 0/3.0
GO TO 20

```

```

C      NO CUBIC ROOT WAS FOUND, SHOCK DETACHED
C
C 10 CONTINUE
C   WRITE (6, 800)
C
C      CUBIC SOLUTION FOUND, ONLY THE MIDDLE ROOT IS REALISTIC
C      AND WILL BE USED
C
C 20 CONTINUE
C   IF ( R1 - R2 ) 30, 110, 40
C 30 K = 1
C   GO TO 50
C 40 K = 2
C 50 IF ( R2 - R3 ) 60, 120, 70
C 60 M = 1
C   GO TO 80
C 70 M = 2
C 80 IF ( K - 60. M ) 60 TO 120
C   IF ( R1 - R3 ) 90, 130, 100
C 90 GO TO ( 130, 110 ), K
C 100 GO TO ( 110, 130 ), K
C 110 SKANG2 = R1
C   GO TO 140
C 120 SKANG2 = R2
C   GO TO 140
C 130 SKANG2 = R3
C
C      CHECK IF SHOCK ANGLE IS NEGATIVE AND PRINT
C
C 140 CONTINUE
C   IF ( SKANG2 - 67. 0 ) 60 TO 170
C   WRITE (6, 805) SKANG2
C   SKANG2 = 0.0
C 170 IF ( SKANG2 - 16. 1 ) 60 TO 200
C   WRITE (6, 810) SKANG2
C   SKANG2 = 1.0
C
C 200 CONTINUE
C   XMSIN2 = XM12 + SKANG2
C 210 CONTINUE
C   IF ( XMSIN2 - 1. 0 ) XMSIN2 = 1. 01
C   FLOWP(4,2) = FLOWP(4,1) * GP1 * XMSIN2 / (GM1 * XMSIN2 + 2. 0)
C   FLOWP(3,2) = FLOWP(3,1) * (2. 0 * XMSIN2 - GM1) / GP1
C   FLOWP(5,2) = FLOWP(5,1) * (2. 0 * XMSIN2 - GM1) * (2. 0 * GM1 * XMSIN2)
C   / ( GP1 * 2 * XMSIN2 )
C   FLOWP(6,2) = FLOWP(6,1) + SORT( FLOWP(5,2) / FLOWP(5,1) )
C   IF ( FLOWP(5,2) - 6E. AV1 ) FLOWP(7,2) = 2. 27E-8 * FLOWP(5,2) * 1. 5
C   / ( FLOWP(5,2) + 198. 6 )
C   IF ( FLOWP(5,2) - 1. 0 ) FLOWP(7,2) = AV2 * FLOWP(5,2) * AV3
C   FLOWP(1,2) = ( GP1 * 2 * XMSIN2 - GM1 ) * ( 2. 0 * GM1 * XMSIN2 +
C   1. 0 ) / ( (2. 0 * XMSIN2 - GM1) * ( 2. 0 * GM1 * XMSIN2 ) )
C   FLOWP(1,2) = SORT( FLOWP(1,2) )
C   IF ( FLOWP(1,2) - 1. 0 ) FLOWP(1,2) = 1. 01
C   FLOWP(2,2) = FLOWP(1,2) * FLOWP(6,2)
C   FLOWP(6,2) = FLOWP(4,2) * FLOWP(2,2) / FLOWP(7,2)
C   SKANG = SORT(SKANG2)
C   IF ( ABS( SKANG ) - 67. 1 ) SKANG = 1. 0
C   SKANG = ASIN( SKANG ) * 90
C
C   CP = 4. 0 * (XMSIN2 - 1. 0) / ( GP1 * XM12 )
C
C 230 RETURN

```

ANGLE OF DEFLECTION EQUALS TO ZERO, CONTINUE THE FLOW FIELD

```

240 CONTINUE
DO 250 I=1, 8
  FLOWP(I,2) = FLOWP(I,1)
250 CONTINUE
CP = 0.0
RETURN

C WEAK OBLIQUE SHOCK RELATIONSHIP
C
260 CONTINUE
SKANG2 = (1.0 + 0.5*CP1/SORT(XM12 - 1.0)*DANG*DR) / XM12
GO TO 200

C 800 FORMAT (1H0,10X,NO CUBIC ROOT, DETACHED SHOCK*)
C 805 FORMAT (1H0,10X,NEGATIVE SHOCK ANGLE, SINE OF SHOCK ANGLE **,
C 1, F12.5)
C 810 FORMAT (1H0,10X,SINE OF SHOCK ANGLE .67, 1.0*,F12.3)
C
C
END
SUBROUTINE EXPAND( DANG, CP)
C
C PRANDTL - MEYER EXPANSION ( DANG .GT. 0 ) OR COMPRESSION
C ( DANG .LT. 0 ) FOR A GIVEN FREE STREAM CONDITIONS
C
COMMON/CONST /DR, RD, G, GP1, GM1, GR, AV1, AV2, AV3
COMMON/FLOW /FLOWP(10,2), X(S1), Y(S1)
C DIMENSION A(2), C(2)
C
C XM1 = FLOWP(1,1)
C
C IF ( FLOWP(1,1) .GE. 1.0 ) GO TO 10
C
C FLOW SUBSONIC, SET MACH NO AT UNITY FOR CONTINUATION
C
C XM12 = 1.0
C GO TO 20
C
10 CONTINUE
C XM12 = XM1**2
C
20 CONTINUE
C
C CALCULATE THE PRANDTL-MEYER ANGLE MU1 (RADIAN) FOR FREE
C STREAM
C
C MU1 = 60*ATAN( SORT(XM12 - 1.0)/GR ) - ATAN( SORT(XM12 - 1.0) )
C
C CALCULATE THE PRANDTL-MEYER ANGLE MU2 AFTER TURNING
C
C XMU1D = XMU1*DR
C XMU2D = XMU1D + DANG
C XMU2 = XMU2D*DR
C
C IF ( XMU2D .GT. 0.0 ) GO TO 30
C
C FLOW AFTER TURNING IS SUBSONIC
C
C XM2A = 1.0
C GO TO 190
C

```

```

30 CONTINUE
CHECKN = 2.2768532
IF (XNU2 .LT. CHECKN) GO TO 40
    FLOW EXPANDS TO INFINITE MACH NUMBER, SET TO 100.0
    XN2A = 100.0
    GO TO 190
    START OF ITERATION FOR THE MACH NUMBER AFTER TURNING
40 CONTINUE
    I = 1
    EPS = 1.0E-4
    BEGIN WITH THE APPROXIMATE MACH NUMBER XN2A
    XN2A = XN1*(1.0 + (XNU2-XNU1)*(1.0 + 0.5*GM1*XN12)/
    3*RT(XN12 - 1.0))
    IF (ABS( DARG ) .LT. 1.0E-1) GO TO 190
50 CONTINUE
    XN1 = (XNU2 + ATAN( SORT( XN2A**2 - 1.0 ) ) ) / GR
    XN2 = TAN ( XN1 )
    XN2 = SORT( 1.0 + ( XN2**2 - 1.0 ) )
    IF ( I .GT. 1 ) GO TO 60
    I = I + 1
    DN1 = XN2 - XN2A
    XN2B = XN2A
    XN2A = 1.1*XN2A
    GO TO 50
60 CONTINUE
    I = I + 1
    DN2 = XN2 - XN2A
    IF ( ABS( DN2/XN2A ) .LT. EPS ) GO TO 190
    IF ( I .GT. 20 ) GO TO 200
    XN2C = XN2A
    XN2A = XN2A - DN2*(XN2A-XN2B)/(DN2-DN1)
    XN2B = XN2C
    DN1 = DN2
    GO TO 50
70 CONTINUE
    ITERATION COMPLETED, CALCULATE FLOW FIELD AFTER TURNING
190 CONTINUE
    FLOWF(1,2) = XN2A
    ONEOM = 1.0/XN2A
    SKANG = ASIN(ONEOM)*60
    Z = (2.0 + GM1*XN12)/(2.0 + GM1*XN2A**2)
    FLOWF(3,2) = FLOWF(3,1) + Z*(6/GM1)
    FLOWF(4,2) = FLOWF(4,1) + Z*(11.0/GM1)
    FLOWF(5,2) = FLOWF(5,1) + Z
    FLOWF(6,2) = FLOWF(6,1) + SORT( Z )
    IF ( FLOWF(5,2) .GE. AV1 ) FLOWF(7,2) = 2.276-8*FLOWF(5,2)**1.5
    1 / ( FLOWF(5,2) + 196.6 )
    IF ( FLOWF(5,2) .LT. AV1 ) FLOWF(7,2) = AV2*FLOWF(5,2)**AV3
    FLOWF(2,2) = XN2A*FLOWF(4,2)
    FLOWF(8,2) = FLOWF(6,2)*FLOWF(2,2)/FLOWF(7,2)
    CP = (2.0*(6/GM1) - 1.0)/(0.5*GM12)
    GO TO 300
200 CONTINUE

```

```

WRITE (6,800) I, IN2A, IN2
C 300 CONTINUE
C 800 FORMAT (1H0, 'ITERATION FAIL TO CONVERGE AFTER ', IS, 2X, ' TIMES.',
, ' CALCULATION CONTINUES', 2F15.3)
C RETURN
C END
C SUBROUTINE SECEXP(LBW)
C C
C C SECOND ORDER SHOCK EXPANSION
C LBW = 1, 2-D FLOW
C 2, CONICAL FLOW
C C
C COMMON/CONST /R, R0, G, GP1, GM1, GR, AV1, AV2, AV3
C COMMON/FRES /ZMACH, PF, ALPHA, BETA, NEND
C COMMON/FLOW /FLOWB(10, 2), XS(51), Y(51)
C COMMON /MYAR /MYAR(14), MROU, MROU, MCOLW
C COMMON/PDATA/MDOTS, INIDB(51), MTHETB, THETB(15), CPDB(51, 15)
C COMMON/XVAR /XURT(426), XB(51)
C COMMON/WINGOT /CPH, RUE, ISOLID, IFORM(10), YAW, DRILL, SLOPE(51)
C DIMENSION CPW(110, 2)
C C
C C MACH NUMBER - FLOWB(1, 1)
C VELOCITY - FLOWB(2, 1)
C PRESSURE - FLOWB(3, 1)
C DENSITY - FLOWB(4, 1)
C TEMPERATURE - FLOWB(5, 1)
C SPEED OF SOUND - FLOWB(6, 1)
C VISCOSITY - FLOWB(7, 1)
C REYNOLDS NUMBER - FLOWB(8, 1)
C C
C FLOWB(1, 1) = ZMACH
C FLOWB(2, 1) = 1.0
C FLOWB(3, 1) = 1.0
C FLOWB(4, 1) = 1.0
C FLOWB(5, 1) = 520.0
C FLOWB(6, 1) = 49.8 * SORT(FLOWB(5, 1))
C FLOWB(7, 1) = FLOWB(1, 1) * FLOWB(6, 1)
C IF ( FLOWB(5, 1) .GE. AV1 ) FLOWB(7, 1) = 2.27E-8 * FLOWB(5, 1) ** 1.5
C / ( FLOWB(5, 1) * 198.4)
C IF ( FLOWB(5, 1) .LT. AV1 ) FLOWB(7, 1) = AV2 * FLOWB(5, 1) ** AV3
C FLOWB(8, 1) = FLOWB(4, 1) * FLOWB(7, 1) / FLOWB(1, 1)
C C
C C CALCULATE PRESSURE ON THE BODY
C C
C 60 TO ( 200, 100), IDW
C 100 CONTINUE
C CALL BODYP
C 60 TO 300
C C
C C CALCULATE PRESSURE ON THE WING
C 200 CONTINUE
C C
C MB1 = MCOLW - 1
C DO 250 I=1, MB1
C FLOWB(1, 1) = ZMACH
C FLOWB(3, 1) = 1.0
C CALL WINGP(I)
C 250 CONTINUE

```


285 CONTINUE

GO 310 J=1, NTHETB
IF (I .GT. 3) GO TO 405
XN1 = XNS(J)
P1 = CPDB(1, J)
PC = PCONEC(J, 2)
PSC(J) = PCONEC(J, 1)
GO TO 407

405 CONTINUE

XN1 = XNS(J)
XN10(1, J) = XN10(2, J)
P1 = PSC(J)
PC = PCONEC(J, 2)
OPDS1 = OPDS(J)
407 CONTINUE

XN1 = GREATAN(SORT(XN1**2 - 1.0)/6R) - ATAN(SORT(XN1**2 - 1.0))

XN2 = XN1 - DANG*DR
IF (XN2 .GT. XN1) GO TO 410

IF (DANG .LT. 1.0) GO TO 410

SHOCK CALCULATION

FLOWD(1, 1) = XN1
FLOWD(3, 1) = P1
CALL COMP(DANG, CP, SKANG)

SKANG = SKANG*DR
XN2 = FLOWD(1, 2)
P2 = FLOWD(3, 2)
TANG = TAN(DANG*DR)
COS = COS(SKANG)
SINS = SIN(SKANG)
SINSA = SIN(SKANG - DANG)*DR
SINH = SIN(1.0/XN2)
XN2 = ASIN(1.0/XN2)
TANG = TAN(XN2)

TH1 = (CP1*TANG*COS - SINS)*(XN1*SINS)**2 + SINS
TH2 = 1.0 + (1.0 - 2.0*SINS**2 + 2.0*TANG*SINS*COS)*(XN1*SINS)

TH3 = (1.0 + 0.5*GRI*XN1**2)*SINS*TH1/(CP1*TH2/4.0)

B1 = CP1*XN1**2/12.0*(XN1**2 - 1.0)

B2 = CP2*XN2**2/12.0*(XN2**2 - 1.0)

TH4 = 2.0*B2/RAD1*(SINS*SIN(CORANG*DR)/SINSA - SIN(CORANG*DR))

OPDS1 = OPDS1*(B2*SINS/(B1*SINSA) + (P2/P1-TN3)*COS*TANG/SINSA)

OPDS2 = TH4 / (1.0 + TANG/TAN(SKANG - DANG*DR))

GO TO 420

410 CONTINUE

PRANDTL-MEYER EXPANSION

DANG1 = -DANG
FLOWD(1, 1) = XN1
FLOWD(3, 1) = P1
CALL EXPAND(DANG1, CP)
XN2 = FLOWD(1, 2)
P2 = P1*(0.5*(CP+XN1**2 + 1.0)
IF (P2 .LT. PHIN) P2 = PHIN
IF (ABS(DANG) .LT. 0.1 .AND. ICHECK .EQ. 0) GO TO 430
ONEG12 = XN2/XN1*((1.0 + 0.5*GRI*XN1**2)/(1.0 + 0.5*GRI*XN2**2))
XN1 = 0.5*CP1*XN1**2/(XN1**2 - 1.0)

```

02 = 0.5*0.02*XM2**2/(XM2**2 - 1.0)
0P052= 02/(1-1)*(COS(0.12)*SIN(CORANG*0.0R) - SIN(CORANG*0.0R) ) +
1 02/01*ONE012*0P051
C 420 CONTINUE
IF ( PC .EQ. P2 ) GO TO 430
ETA = 0P052* 03/(PC - P2)
IF ( ETA .GT. 100.0 ) ETA = 100.0
IF ( ETA ) 430, 430, 440
C 430 CONTINUE
XM3 = XM2
P3 = P2
CP00(I-1,J) = P3
XM10(2, J) = XM3
0P053 = 0.0
GO TO 400
C 440 CONTINUE
P3 = PC - (PC - P2)*EXP(-ETA)
XM3 = SORT(2.0/6M1*((1.0 + 0.5*6M1*XM2**2)*(P2/P3)**6R1 - 1.0))
ETAM10 = 0.5*ETA
CP00(I-1,J) = PC - (PC-P2)*EXP(-ETAM10)
XM10(2, J) = SORT(2.0/6M1*((1.0 + 0.5*6M1*XM2**2)*(P2
1 /CP00(I-1,J))**6R1 - 1.0) )
0P053 = (PC - P3)/(PC - P2)*0P052
C 400 CONTINUE
C
XMS(J) = XM3
PS(J) = P3
0P05(J) = 0P053
C
IF ( ABS( ALPHA ) .LT. 1.0E-2 ) GO TO 311
310 CONTINUE
GO TO 350
311 CONTINUE
DO 315 J=2, NTHET0
CP00(I-1, J) = CP00(I-1, 1)
315 CONTINUE
C 320 CONTINUE
C
RS(I) = RS(I-1) + (XS(I)-XS(I-1))*TAN( THETA(4)*0.0R )
C 375 CONTINUE
IC = IC + 1
C
CORANG = CORANG
C
IF ( ABS(CORANG) .LT. 0.1 .AND. ICHECK .EQ. 1 ) GO TO 450
ICHECK = 1
RAT = ( XROOT - XS(I-1) )/( XS(I) - XS(I-1) )
XROOT = XM10(1,4) + ( XM10(2,4) - XM10(1,4) ) * RAT
XROOT = CP00(I-2,4) + ( CP00(I-1,4) - CP00(I-2,4) ) * RAT
XROOT = Y(I-1) + ( Y(I) - Y(I-1) ) * RAT
XMK1 = XS(I-1) + ( XS(I) - XS(I-1) ) * RAT
C 450 CONTINUE
C
DO 500 J=1, NROOTS
00 500 J=1, NTHET0
CP00(I, J) = (CP00(I, J) - 1.0)/0.012
500 CONTINUE
C
C

```

```

RETURN
END
SUBROUTINE FORCED( XB, RB, ARB, YAW, BROLL, DBCL, DBCD, DBCO,
1 DBCM, DBCHM, DBCTH)
C
C
C INTEGRATES PRESSURE (CPDB) ON THE BODY FOR FORCES
COMMON/CONST /AR, RB, S, GP1, SM1, SR, AV1, AV2, AV3
COMMON / BODYSP/ BDR(51), DROT(51), DRDC(51), DRDTC(51), DRDTP
COMMON / PDATA/ BODYS, IMIDB(51), MTHETB, MTHETB(15), CPDB(51,15)
COMMON / INPUT / TX(2, 50)
COMMON / POINTP/ T1(4), T2(4), T3(4), T4(4)
COMMON / POINTS/ MPTS, NVAR
1 DIMENSION CP(15), CPT(15), CP2(15), CP3(51), CP3(51),
CP2(51), COSB(15), SINB(15), XB(1), RB(1)
PI=3.1415926
NT = MTHETB
NT1 = NT * 1
NT2 = 2*( NT - 1 )
NT3 = MTHETB
NT4 = NT2 * 1
DX = XB(2) - XB(1)
MTHETB = (MTHETB(2) - MTHETB(1)) * DX
TANA = TAN(ARB * DX)
COSA = COS(ARB * DX)
SINA = SIN(ARB * DX)
TANY = TAN(YAW * DX)
A1 = 1.0 / SQR( 1.0 + TANA**2 + TANY**2 )
COSBT = A1 * SQR( 1.0 + TANA**2 )
SINBT = SQR( 1.0 - COSBT**2 )
COSA = COS(BROLL * DX)
SINA = SIN(BROLL * DX)
SINB = SIN(BROLL * DX)
MDD1 = MDDYS - 1
FROLL = 0.0
DO 60 I=1, NT4
MTHETB(I) = (2-1)*MTHETB/DR
COSB(I) = COS( MTHETB(I) * DR )
SINB(I) = SIN( MTHETB(I) * DR )
60 CONTINUE
C
C INCORPORATE EFFECT OF YAW TO ANGLE OF ATTACK
C
C
ARATAN = ACOS( A1 )
SINAT = SIN(ARATAN)
IF ( YAW .LT. 0.01 .AND. BROLL .LT. 0.01 ) GO TO 190
NT = NT2
NT3 = NT * 1
IF ( ARB .LT. 0.01 ) GO TO 90
FROLL = ATAN( TANY / TANA ) * RB
GO TO 150
90 CONTINUE
FROLL = 90.0
150 CONTINUE
C
C SPLINE FIT CPDB IN THE BODY COORDINATE SYSTEM
C
C
DO 100 I=1, NT
CP3(I) = MTHETB(I) * FROLL - BROLL
IF ( CP3(I) .GT. 180.0 ) CP3(I) = ABS( 360.0 - CP3(I) )
IF ( CP3(I) .LT. 0.0 ) CP3(I) = ABS( CP3(I) )
100 CONTINUE
NVAR = 2
MPTS = MTHETB
DO 100 J=1, MDD1

```



```

      XL = -FX*SINA*CO5B + FY*(SINA*CO5A+CO5B*SINA) + FZ*
      (CO5B*CO5A - SINA*SINA)
      XD = FX*CO5A*CO5B + FY*(SINA*SINA-CO5B*CO5A) + FZ*
      (SINA*SINA+CO5A*CO5B)
      XB = FX*SINA + FY*CO5B*CO5B - FZ*SINA*CO5B
      XLIFT = XLIFT + XL
      DRAG = DRAG + XB
      BANK = BANK + XB
      WCPM = WCPM - XL*ZCR(IJ)
      WCPM = WCPM + F*CO5A*TC(IJ)
      WCTM = WCTM + XB*ZCR(IJ) - XD*TC(IJ)
200 CONTINUE
      MCL = MCL + XLIFT
      MCD = MCD + DRAG
      MCB = MCD + BANK
100 CONTINUE
C
      RETURN
      END
      SUBROUTINE WRITEM(MCT, MD1, 2)
      WRITE WING SURFACE SLOPE AND PRESSURE COEFFICIENT
C
      DIMENSION Z(1)
      MD2 = MD1 - 1
      MC = MCT + 1
      MD22 = MD2*MC
      WRITE (6,801) ( I, I=1, MD1)
      WRITE (6,802)
      DO 100 J=1, MC1
      JS = J + 1, MC1
      JE = JS + MD22
      WRITE (6, 803) ( J, (Z(I), I=JS,JE,MC) )
100 CONTINUE
C
      801 FORMAT (1H, ' SPANWISE STATION', I8, 9110/(10H,18,9110))
      802 FORMAT (1H, ' CHORDWISE STATION', I8)
      803 FORMAT (1H, I10, F19.5, 9110.5/(12H,5,9110.5) )
C
      RETURN
      END
      SUBROUTINE CUBERT
      DIMENSION X(3)
      COMMON/LOCIN / V(3), L, N, 6(4), VVAR, VVAR,
      COMMON/POINTS/ MPTS, NVAR
      L=0
      SUM = SORT( 6(1)+2*6(2)+2*6(3)+2*6(4)+2 )
      61 = 6(1) / SUM
      62 = 6(2) / SUM
      63 = 6(3) / SUM
      64 = 6(4) / SUM
C
      E0.1 61*X+3 + 62*X+2 + 63*X + 64 = 0.
      IF ( ABS( 64 ) .LT. 1.E-6) GO TO 4
C
      E0.2 61*X+3 + 62*X+2 + 63*X + 64 = 0.
      IF ( ABS( 61 ) .GT. 1.E-6) GO TO 9
C
      E0.3 62*X+2 + 63*X + 64 = 0.
      IF ( ABS( 62 ) .GT. 1.E-6) GO TO 7
C
      E0.4 63*X + 64 = 0.
      IF ( ABS( 63 ) .GT. 1.E-6) GO TO 5

```

```

C 4 CONTINUE
  V(1) = 0.0
  L=1
  RETURN
C 5 CONTINUE
  V(1) = - G(4) / G(3)
  GO TO 35
C 7 CONTINUE
  R1 = G(3)*2 - 4.0*G(2)*G(4)
  IF ( R1 .GT. 0.0 ) GO TO 107
  WRITE (6,901)
  901 FORMAT (1H0,'NEGATIVE RADICAL RESULTED FROM SOLVING THE QUADRATIC
    EQUATION 62.X2 + 63.X + 64 = 0')
  RETURN
C 107 CONTINUE
  R1=SQRT(R1)
  V1 = ( -G(3) + R1 ) * 0.5 / G(2)
  V(1) = V1
  IF ( V1 .GT. VMIN .AND. V1 .LT. VMAX ) GO TO 8
  V2 = (-G(3) - R1) * 0.5 / G(2)
  V(2) = V2
  IF ( V2 .GT. VMIN .AND. V2 .LT. VMAX ) GO TO 8
  WRITE (6,902) VMIN, V1, V2, VMAX
  902 FORMAT (1H0,'BOTH ROOTS V1, V2 OF QUADRATIC EQ. 62.X2 + 63.X + 64
    = 0 OUTSIDE OF LIMIT VMIN, VMAX/ 12, VMIN =,E14.6, V1 =,
    2 E14.4, V2 =,E14.6, VMAX =,E14.6)
  RETURN
C 8 CONTINUE
  L = 1
  RETURN
C 9 THREE ROOTS
  CONTINUE
  A1 = G(2) / G(1)
  A2=G(3)/G(1)
  A3=G(4)/G(1)
  A=A2-A1/3.
  B=2.-A1-A1/27.-A1*A2/3.*A3
  RAD=B*B/4.*A*A/27.
  SIGN=1.
  IF(B.GE.0.) SIGN=-1.
  IF(RAD)10,20,20
    C 10 THREE UNEQUAL REAL ROOTS
    CONTINUE
    A3 = SQRT( ABS( A/3.0 ) )
    A2 = SQRT( ABS( A*3/3. ) )
    PHI = ACOS( ABS( 1.5*B/ A2 ) )
    DO 11 I=1,3
      X(I) = 2.0*SIGN*A3*COS((PHI+6.2831853*FLOAT(I-1))/3.0)
    11 CONTINUE
    GO TO 40
    C 20 EITHER ONE REAL AND TWO IMAG. OR THREE REAL ROOTS OF WHICH TWO
    ARE EQUAL
    CONTINUE
    V1 = 1.0 / 3.0
    B = A2*V1 - ( A1*V1)*0.2
    R = .5*A1*A2*V1 - .5 *A3 - ( A1*V1 ) *0.3
    RAD = 0.03 * R*R

```

```

IF ( RAD .GT. 0.0 ) 60 TO 120
WRITE (6,903)
903 FORMAT (1H0,'ERROR IN TESTING RADICAL, REDUNDANT TEST')
RETURN
120 CONTINUE
RAD = SQRT( RAD)
A2 = R * RAD
SIGN = 1.
IF ( A2 .LT. 0.0 ) SIGN = -1.
A2 = SIGN * ABS( A2 ) **V1
A3 = R - RAD
SIGN = 1.
IF ( A3 .LT. 0.0 ) SIGN = -1.
A3 = SIGN * ABS( A3 ) **V1
V0 = A2 * A3 - A1*V
35 CONTINUE
IF ( V0 .GT. VMIN .AND. V0 .LT. VMAX ) L = 1
IF ( L .EQ. 0 ) WRITE (6,905)
905 FORMAT (1H0,'THE ONLY UNEQUAL ROOT IS OUTSIDE OF VMIN, VMAX=/',L,
1 VMIN=,E16.4,X, V=,E16.4,X, VMAX=,E16.4)
V(1) = V0
RETURN
40 CONTINUE
DO 45 I=1,3
V0 = X(I) - A1/3.0
X(I) = V0
V(I) = V0
IF ( V0 .LT. VMIN .OR. V0 .GT. VMAX ) 60 TO 30
L=L+1
RETURN
50 CONTINUE
45 CONTINUE
WRITE (6,904) VMIN, V(1), V(2), V(3), VMAX
904 FORMAT (1H0,'ALL THREE UNEQUAL ROOTS OUTSIDE OF LIMIT VMIN, VMAX=/'
1 L, VMIN=,E16.4,X, V1=,E16.4,X, V2=,E16.4,X, V3=,
2 E16.4,X, VMAX=,E16.4)
RETURN
END
C
SUBROUTINE POINT(M)
COMMON/CONFIT/ B(2,4), 50), T( 50), SCALEF(2)
COMMON/ENDPT/ IP1ST, XLAST, XSAVE( 50)
COMMON/LOCTH/ V(3), L, R, A(4), VMAX, VMIN
COMMON/POINTS/ NPTS, NVAR
COMMON/ POINTP/ X(4), XPC( 4), XPP( 4)
DIMENSION B(4)
N1 = NPTS - 1
IF (N.NE.0) 60 TO 4
DO 1 I=1,N1
N2=I
IF ( V(I) .LE. T(I+1) ) 60 TO 2
1 CONTINUE
WRITE(6,100)
100 FORMAT(1H0,33H ERROR - PARAMETER EXCEEDS LIMITS)
RETURN
2 CONTINUE
DO 300 K=1,3
P = V(K)
IX(K) = 1.E10
IF ( P .LT. VMIN .OR. P .GT. VMAX ) 60 TO 300
IX(K) = ABS(((B(1,M))P + B(1,2,M))P + B(1,3,M))P + B(1,4,M))P
SCALEF(1) = XIND
300 CONTINUE

```

```

IF ( DX(1) .GT. DX(2) ) GO TO 302
IDENT = 1
IF ( DX(1) .GT. DX(3) ) IDENT = 3
GO TO 305
302 CONTINUE
IDENT = 2
IF ( DX(2) .GT. DX(3) ) IDENT = 3
305 CONTINUE
P = V(IDENT)
C
DO 3 I=1, NVAR
X(I) = ((B(1,1,M)*B(1,2,M)) + B(1,3,M)*B(1,4,M)) * SCALEF(I)
X(I) = ((B(1,1,M)*B(1,2,M)) + B(1,3,M)*B(1,4,M)) * SCALEF(I)
3 XPD(I) = (B(1,1,M)*B(1,2,M)) + B(1,3,M)*B(1,4,M) * SCALEF(I)
C
310 CONTINUE
RETURN
C
4 X(M) = X(M) / SCALEF(M)
DO 5 I=1, M1
M=I
K = I * I
IF ( XSAVE(K) .GE. X(M) ) GO TO 6
5 CONTINUE
XIND = X(M) * SCALEF(M)
XFT = XFIRST * SCALEF(1)
XLT = XLAST * SCALEF(1)
WRITE (6,101) XIND, XFT, XLT
101 FORMAT (1H0, ' THE GIVEN VALUE OF INDEPENDENT VARIABLE ', E12.4,
1
5X, ' IS OUT OF RANGE TO BE INTERPOLATED ', E12.4, 5X, E12.4)
RETURN
6 DO 7 I=1, 4
7 A(I) = B(M, I, M)
A(4) = A(4) - X(M)
IF ( XFIRST .EQ. X(M) ) GO TO 112
IF ( XLAST .EQ. X(1) ) GO TO 111
VMAX = T(M-1) * 1.0001
VMIN = T(M) * 0.9999
CALL CUBERT
XIND = X(1) * SCALEF(1)
GO TO 103
112 V(1) = 0.0
L=1
GO TO 103
111 V(1) = 1.0
L=1
103 CONTINUE
IF (L.NE.0) GO TO 2
OP = P
WRITE (6,10) OP
10 FORMAT (1X#OP =, E15.8)
DO 8 J=1, 2
K = M * J - 1
B1 = B(M, 1, K)
B2 = B(M, 2, K)
B3 = B(M, 3, K)
B4 = B(M, 4, K)
P = T(K)
SUM = B1*P**3 + B2*P**2 + B3*P + B4
OIF = SUM - X(M)
8 WRITE (6,9) K, P, B1, B2, B3, B4, X(M), SUM, OIF
9 FORMAT (1X13, E15.8)
WRITE (6,102)
102 FORMAT (1H0, 32H FAILURE IN CUBE ROOT EXTRACTION)
RETURN

```

```

END
SUBROUTINE SPLFIT
REAL L,M,MU
COMMON/CURFIT/ B(2,4,50), T( 50), SCALEF(2)
COMMON/EMPT/ XFIRST,XLAST,XSAVE( 50)
COMMON/INPUT / X( 2, 50)
COMMON/POINTS/ NPTS, NVAR
DIMENSION M(2, 50), S( 50), L( 50), MU( 50), P( 50), U( 50)
1 IF(NPTS .GT. 50) NPTS= 50
DO 200 I=1, NVAR
SCALEF(I)=0.
DO 201 J=1,NPTS
DO 201 J=1,NVAR
201 SCALEF(J)=MAX1(SCALEF(J),ABS(X(J,I)))
DO 101 J=1,NVAR
IF ( SCALEF(J) .LT. 1.E-10 ) SCALEF(J) = 1.0
101 CONTINUE
DO 202 I=1,NPTS
DO 202 J=1,NVAR
202 X(J,I)=X(J,I)/SCALEF(J)
DO 203 I=1,NPTS
XSAVE(I) = X(1,I)
203 CONTINUE
XFIRST = X(1,1)
XLAST = X(1,NPTS)
T(1)=0.
S(1)=0.
SUM=0.
L(1)=-2.
MU(1)=0.
L(NPTS)=0.
MU(NPTS)=-2.
O(1)=0.
O(NPTS)=0.
P(1)=2.
O(1)=1.
U(1)=0.
M1=NPTS-1
DO 2 I=2,NPTS
SUM=0.
DO 1 J=1,NVAR
1 SUM1=SUM1+(X(J,I)-X(J,I-1))**2
S(I)=SUM1/SUM
2 T(I)=T(I-1)+S(I)
DO 3 I=2,NPTS
S(I)=S(I)/T(NPTS)
3 T(I)=T(I)/T(NPTS)
DO 4 I=2,M1
L(I)=S(I-1)/(S(I)+S(I+1))
MU(I)=1.-L(I)
P(I)=MU(I)*O(I-1)+2.
4 O(I)=L(I)/P(I)
P(NPTS)=2.*(1.-O(M1))
DO 7 I=1,NVAR
DO 5 J=2,M1
O(J)=6.*(X(I,J+1)-X(I,J))/S(J+1)-(X(I,J)-X(I,J-1))/S(J))/(O(J)+
1S(J+1))
5 U(J)=(O(J)-MU(J)+U(J-1))/P(J)
U(NPTS)=MU(NPTS)+U(M1)/P(NPTS)
M(I,NPTS)=U(NPTS)
DO 6 J=1,M1
K=NPTS-J
M(I,K)=O(K)+M(I,K+1)+U(K)

```

```

B(I,1,K)=(M(I,K+1)-M(I,K))/(6.*S(K+1))
B(I,2,K)=S*(M(I,K)*T(K+1)-M(I,K+1)*T(K))/S(K+1)
B(I,3,K)=(S*(P(I,K+1)*T(K)+2*M(I,K)*T(K+1)+2)*X(I,K+1)-X(I,K)*
1*(M(I,K)-M(I,K+1))*S(K+1)+2/6.)/S(K+1)
6 B(I,4,K)=(M(I,K)*T(K+1)+3-M(I,K+1)*T(K)+3)/(6.*S(K+1))*X(I,K)+Y
1(K+1)-X(I,K+1)*T(K)/S(K+1)+S(K+1)/6.*(M(I,K+1)*T(K)-M(I,K)*T(K+1)
1)
7 CONTINUE
RETURN
END

```

DISTRIBUTION LIST

	<u>Copies</u>
<u>Navy</u>	
Commander	
Naval Surface Weapons Center	
Dahlgren, Virginia 22448	
G05 (Soper)	1
G40	1
G402	30
G41	1
G402 (Matra)	32
G42 (Graff)	1
N11 (Hardy)	1
K21 (Blackman)	1
K82 (Edwards)	1
X211	3
S20 (Reinhard)	1
K21 (Purvis)	1
R44 (Baltakis)	1
K21 (Moore)	1
Office of Chief of Naval Operations	
Operations Evaluation Group (OP-03EG)	
Washington, D. C. 20350	1
Defense Nuclear Agency	
Washington, D. C. 20301	2
Commander	
Naval Sea Systems Command	
Washington, D. C. 20362	
SEA-03A	
SEA-0333	1
SEA-035-13	2
SEA-035A	1
SEA-55212	1
SEA-0632	2
SEA-654312	1
PMS-403-02	1
Commander	
Naval Air Systems Command (AIR-320-B)	
Washington, D. C. 20361	1
Commander	
Pacific Missile Test Center	
Point Mugu, California 93042	
T. W. Elliott (Code 5331)	1

Copies

Commander
Naval Air Development Center
Johnsville, Warminister, Pennsylvania 18974

Dr. E. McQuillan (Code 3033)

1

Herman Rubin (Code 3033)

1

K. T. Yen (Code 30331)

1

Naval Research Laboratory

Washington, D. C. 20390

Dr. M. R. Achter (Code 6340)

1

Commander

Naval Weapons Center

China Lake, California 93557

Code 4062

1

Code 45703

1

Mr. E. L. Jeter

1

James E. Serpanos

1

Office of Naval Research

Department of the Navy

North Quincy Street

Arlington, Virginia 22217

Code 461

1

Code 439 (N. Perrone)

1

Code 470

1

Technical Library

2

Naval Postgraduate School

Monterey, California 93940

Code 2124

1

Dr. J. G. Cantin

1

Commanding Officer

Naval Ship Research and Development Center

Bethesda, Maryland 20034

Code 1844 (Jim McKee)

1

Code 1844 (Gordon Everstine)

1

Code 564

2

Air Force

AFFDL/PTS (CAPT Gilbert L. Camburn)

AF Flight Dynamics Laboratory

Wright-Patterson Air Force Base, Ohio 45433

1

AF Materials Laboratory

Wright-Patterson Air Force Base, Ohio 45433

Code MAAN (B. R. Emrich)

1

Copies

Flight Dynamics Laboratory
Structures Division
Wright-Patterson Air Force Base, Ohio 45433
Mr. Leo J. D. Bernier 1

AFML/LLM (W. G. Ramke)
Wright-Patterson Air Force Base, Ohio 45433 1

AFATL/DLRV
Eglin Air Force Base, Florida 32542
James M. Heard 2

AFAL/TEM-3
Wright-Patterson Air Force Base, Ohio 45433
Richard A. Ireland 1

Army

Commanding General
U. S. Army Missile Command
Redstone Arsenal, Alabama 35809 1

Other Agencies

NASA Scientific and Technical Information Facility
P. O. Box 33
College Park, Maryland 20740
Acquisitions Branch - Foreign Exchange 2

Industry

Aerospace Corporation
1111 East Mill Street
San Bernardino, California 92408
Dr. Robert L. Hallse 1

Booz Allen Applied Research, Inc.
P. O. Box 874
Shalimar, Florida 32579
William Day 1

Applied Physics Laboratory
The Johns Hopkins University
Johns Hopkins Road
Laurel, Maryland 20810
L. B. Weckesser 1
W. C. Caywood 1
R. M. Rivello 1

Copies

Battelle Memorial Institute
Defense Metals Information Center
505 King Avenue
Columbus, Ohio 43201
H. D. Moran

1

Brunswick Corporation
Technical Products Division
325 Brunswick Lane
Marion, Virginia 24354
Robert Copeland

1

General Dynamics/Pomona
P. O. Box 2507
Pomona, California 91766
R. A. Miller
D. A. Underhill

1

1

University of Maryland
College Park, Maryland 20740
Dr. Schaeffer

1

Raytheon Company
Missile Systems Division, VB1-3
Hartwell Road
Bedford, Massachusetts 01730
R. D. Howe
D. H. Sanders

1

1

McDonnell Douglas Astronautics Co.
5301 Bolsa Avenue
Huntington Beach, California 92647
W. F. Bozich
K. K. Wang
R. S. Lee

1

1

1

McDonnell Douglas Astronautics Co.
Eastern Division
P. O. Box 516
St. Louis, Missouri 63166
J. Gubser

1

Lockheed Missiles and Space Co., Inc.
Dept S1-12, Building 154
Sunnyvale, California 94088
J. A. Bailie

1

	<u>Copies</u>
Lockheed Missile and Space Co., Inc. Dept 52-30, Building 103 P. O. Box 504 Sunnyvale, California 94088 Robert M. Beasley Senior Staff Scientist	1 1
Lockheed Missile and Space Co., Inc. Technical Information Center 3251 Hanover Street Palo Alto, California 94034	1
Hughes Aircraft Company Missile System Division 8433 Fallbrook Avenue Canoga Park, California 91304 R. J. Oedy	1
Hughes Aircraft Company Centinela and Teale Streets Culver City, California 90230 L. E. Gates, M.S. H-164	1
The Boeing Company P. O. Box 3999 Seattle, Washington 98124 R. Blaisdell K. Brettmann	1 1
Martin-Marietta Corporation P. O. Box 5837 Orlando, Florida 32855 G. H. Sprague, Jr. (Director Aeormechnical Division) G. Fotieo	1 1
Honeywell, Incorporated 600 2nd Street North Hopkins, Minnesota 55343 J. D. Brannan	1
Fairchild Republic Co. Farmingdale, New York 11735 Jim Clarke Mail Stop Z-19A-08	1 1
Raytheon Co. Spenser Lab Burlington, Massachusetts 01803 Doris A. Holden Dept 7113	1 1
TOTAL	150