NASA CONTRACTOR REPORT

NASA CR-2676

NASA CR-2676

DEVELOPMENT OF A COMPUTER CODE FOR CALCULATING THE STEADY SUPER/HYPERSONIC INVISCID FLOW AROUND REAL CONFIGURATIONS

Volume II - Code Description

Frank Marconi and Larry Yaeger

Prepared by
GRUMMAN AEROSPACE CORPORATION
Bethpage, N.Y. 11714
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MAY 1976

1. Report No.
NASA CR-2676

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Development of a Computer Code for Calculating the Steady Super/Hypersonic Inviscid Flow Around Real Configurations - Volume II - Code Description

5. Report Date
May 1976

6. Performing Organization Code

7. Author(s)
Frank Marconi and Larry Yaeger

8. Performing Organization Report No.

10. Work Unit No.
505-26-10-07

9. Performing Organization Name and Address
Grumman Aerospace Corporation
Bethpage, NY 11714

11. Contract or Grant No.
NAS1-11525

13. Type of Report and Period Covered
Contractor Report

12. Sponsoring Agency Name and Address
National Aeronautics & Space Administration
Washington, DC 20546

14. Sponsoring Agency Code

15. Supplementary Notes
Langley Technical Monitor: Harris Hamilton
Final Report

16. Abstract

A numerical procedure has been developed to compute the inviscid super/hypersonic flow field about complex vehicle geometries accurately and efficiently. A second-order accurate finite difference scheme is used to integrate the three-dimensional Euler equations in regions of continuous flow, while all shock waves are computed as discontinuities via the Rankine-Hugoniot jump conditions. Conformal mappings are used to develop a computational grid. The effects of blunt nose entropy layers are computed in detail. Real gas effects for equilibrium air are included using curve fits of Mollier charts. Typical calculated results for shuttle orbiter, hypersonic transport, and supersonic aircraft configuations are included to demonstrate the usefulness of this tool.

17. Key Words (Suggested by Author(s))
Vehicle flow fields, three -dimensional flows, supersonic/hypersonic flows, inviscid flow, numerical flow field computations

18. Distribution Statement
Unclassified - Unlimited

Subject Category 12

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
153

22. Price*
$6.25

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

DEVELOPMENT OF A COMPUTER CODE FOR CALCULATING THE STEADY SUPER/HYPERSONIC INVISCID FLOW AROUND REAL CONFIGURATIONS

VOLUME II - CODE DESCRIPTION

by

F. Marconi and L.S. Yaeger

GRUMMAN AEROSPACE CORPORATION

SUMMARY

A set of four computer codes has been developed to compute the inviscid super/hypersonic flow field about complex vehicle geometries. The numerical procedures used in these codes are described in detail in Volume I of this report. Here the codes developed are described with two views; one oriented toward the user and the other toward the programmer.

The nomenclature used in the codes, the input and output formats, and the storage requirements and computer time are discussed in detail. A description of routines, over-all logic flow, and overlay structure are also presented.

Page intentionally left blank

CONTENTS

CONTENTS (Continued)

ORIENTATION

Volume I gave the approach to the numerics and this volume gives all the other matters closely related to the codes.

When handling real configurations, one matter of decisive importance is how the geometry is modeled, particularly when one solves partial differential equations (rather than integral equations as in other aerodynamic efforts). Therefore the user needs to have an idea of the approach to the geometry modeling before learning the operations of the codes. In this piece of work, geometry modeling is done with a technique developed by A. Vachris and L. Yaeger. For the reader unfamiliar with it, Appendix A gives a brief, self-contained description of this technique, called the QUICK Geometry System. Appendix A is couched in code oriented terms without indulging in dissertations of lofting techniques.

To compute the transonic flow over the nose of blunted vehicles, a three dimensional time asymptotic technique was used. The code (BLUNT) which was used for these calculations is briefly discussed in Appendix B. The computational procedure used is discussed in reference 1.

The typical user will be interested in Part 1, the 'user oriented documentation', which will give him the minimum amount of information necessary to operate, as 'black boxes', the codes developed or adapted under this contract. The large amount of nomenclature included here is to be used primarily as a dictionary; only a few symbols and terms need to be learned by the user, namely those that appear in the input/output data format. The nomenclature and non-dimensionalizations of Volume I of this report are used here.

To the programmer who wants to look into the 'black boxes', Part 2 of this volume is dedicated.

PART 1 USER-ORIENTED DOCUMENTATION

DESCRIPTION OF CODES AND THEIR INTERACTION

A series of five codes has been developed or adapted under this contract:

QUICK - written by A. Vachris and L.S. Yaeger, is a geometry system designed to allow the user to model a complex vehicle geometry in a quick, straightforward fashion. The QUICK geometry system also allows another code, which uses the modeled vehicle geometry as input, to interrogate the model for cross sectional information as efficiently as possible. QUICK consists of an initial defining and logical checkout group of routines, which actually set up the mathematical model, and a second group of routines (called SUB-QUICK throughout this report) which is used for interrogating the mathematical model. SUB-QUICK is used as a part of QUICK to inspect the modeled vehicle, and as a part of the supersonic flow field code (STEIN), along with an output data set from QUICK (the QUICK intermediate data deck), to supply all geometry information.

STEIN - written by F. Marconi and L.S. Yaeger, is a supersonic flow field code designed as a tool to allow the user to compute the super/hypersonic inviscid flow about realistic configurations. The numerical techniques utilized in STEIN are described in detail in Volume I of this report. STEIN reads control data, starting plane data, and geometry data, and computes the flow from the starting plane to a user prescribed axial station. The nose region of the vehicle must be computed with another code which generates starting plane data (where the axial Mach number is supersonic, see Volume I). In STEIN there is a routine which will compute the starting plane data for sharp circular cones at small angles of attack, so the initial data need not be generated elsewhere for this case. For blunt nose vehicles a BLUNT BODY code, developed by Professor Gino Moretti (ref. 1) which is compatible with QUICK and STEIN is used. STEIN computes the flow field, the aerodynamic coefficients and the metric coefficient from the starting plane to the end station.

STRMBL - written by L.S. Yaeger, is a code designed to utilize flow field data, output on tape from STEIN, to compute streamlines on the body, create pseudostream surfaces (p-s-s; defined by the body surface normals taken at each point along a given body streamline), and evaluate flow variables and their normal derivatives along the streamlines and in the p-s-s from the starting plane to the end station.

BOOM - written by L.S. Yaeger, is a code designed to utilize flow field data (from the same STEIN output tape used by STRMBL) to evaluate flow variables on a data cylinder (whose centerline is the z-axis and radius is user-specified) for sonic boom work.

BLUNT - developed by Moretti, uses a time dependent computational technique to asymptote to a steady transonic solution. Its results are used as an initial condition to compute three dimensional supersonic flow over blunt nose vehicles. Details of the technique used to compute the blunt nose flow fields are presented in reference 1. The geometry input for this blunt body code can be either supplied by the geometry package ("QUICK") or computed internally for simple noses. The output from this code is compatible with the three dimensional supersonic flow field code's (STEIN) requirements for initial data. The input for BLUNT is described in Appendix B.

The interaction of these codes (i.e., input-output flow) is described in figure 1.

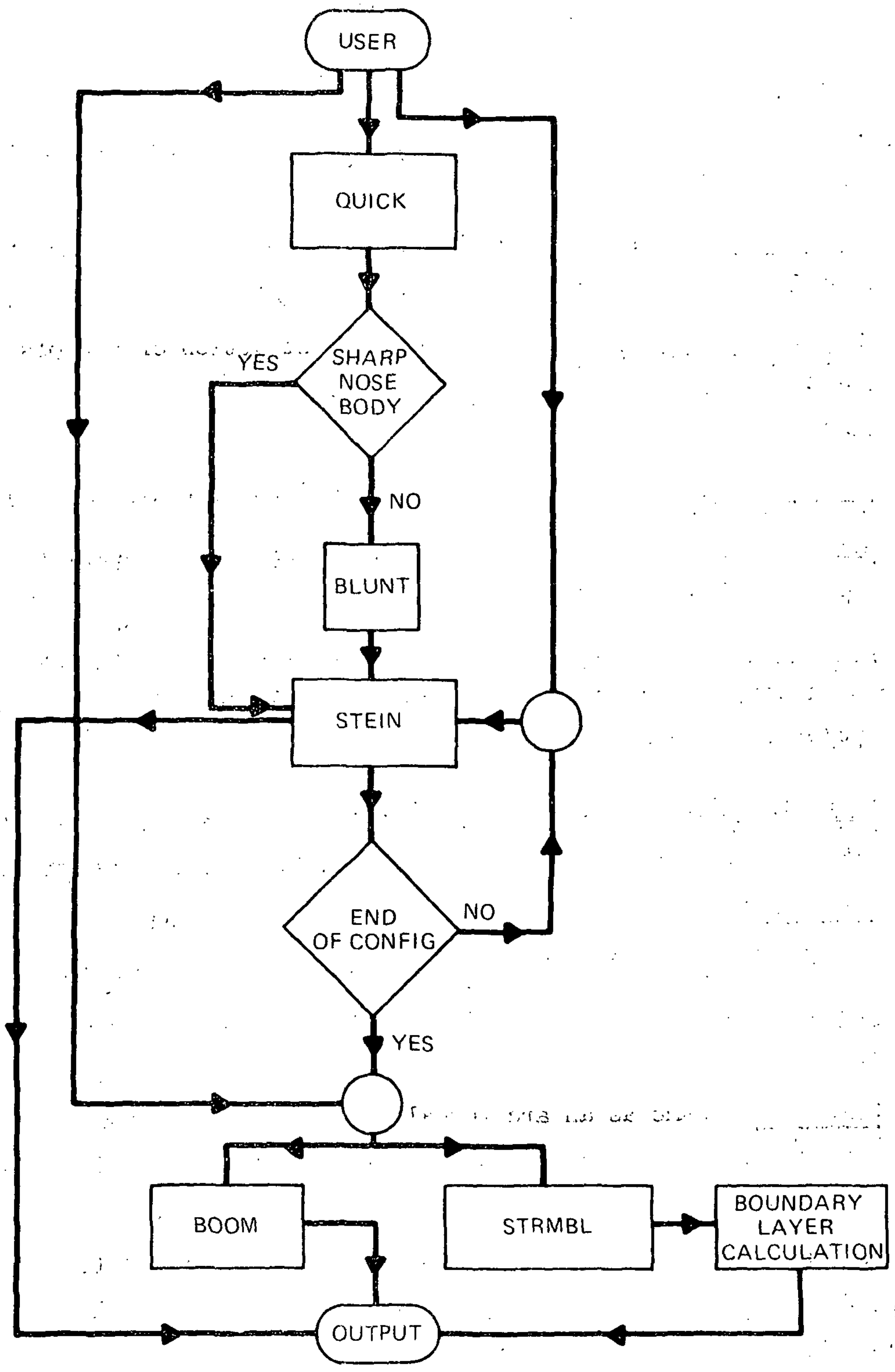


Figure 1 - INTERACTION OF SYSTEM OF CODES

NOMENCLATURE

QUICK TERMINOLOGY

During the discussion of QUICK, several terms will appear frequently, and as such, will be defined here:

1) Cross section - standard definition; a planar cut through the vehicle normal to the FRL at a given x-station.

2) Cross-sectional model - mathematical abstraction of a cross section, using simple curves to represent arcs between specified control points.

3) Control points - break or joining points for defining each arc.

4) Arc - a portion of one simple mathematical curve between two control points in cross section.

5) Body lines - the defining lines of the vehicle geometry in plan and profile views; x-running control points given as $y_i = y_i(x)$ and/or $z_i = z_i(x)$.

6) Body line model - mathematical abstraction of a body line, using simple curves to represent segments between specified match points.

7) Match or Key points - break or joining points between body line segments; initial and terminal points for defining each segment.

8) Segment - a portion of one simple mathematical curve between two match points of a body line model.

9) Component - same as an arc; usually considered to be a named portion of the vehicle geometry (e.g., a wing-upper-ellipse may be component WNGUPELL).

Body line segments are discussed in terms of an origin point at (x_1, v_1) (v standing for y or z), a termination point (x_2, v_2), an initial slope t_1 and a final slope t_2.

SYMBOL LIST FOR QUICK

ANAME — Hollerith input variable; body line (BL)/control point name to which BNAME is to be aliased, when applicable (blank when not)

ARCNAM — Hollerith input variable; cross section (CS) arc or component name

ARCNM(1) — Hollerith input variable; if type is FILET: the name of the most aft component arc to which the current arc's forward end is to be filleted

If type is other: the name of the most aft component arc which, in case of intersection with the current arc, is to update the forward end of the current arc and the aft end of the intersected arc

ARCNM(2) — Hollerith input variable; if type is FILET: the name of the most forward component arc to which the current arc's aft end is to be filleted

If type is other: the name of the most forward component arc which, in case of intersection with the current arc, is to update the aft end of the current arc and the forward end of the intersected arc

ASHAPE — Hollerith input variable; arc or component shape

ASPEC(1) — Hollerith input variable:

= blank yields no effect

= Y when type is FILET, and only y-values are to be specified for the next control point in order of input (z is computed on controlling component)

= Z when type is FILET, and only z-values are to be specified for the next control point in order of input (y is computed on controlling component)

= B to indicate that the next control point is the bottom centerline of the vehicle for the model currently being defined **(optional)**

= T to indicate that the next control point is the top centerline of the vehicle for the model currently being defined **(optional)**

ASPEC(2) Same as ASPEC(1)

ATYPE Hollerith input variable; arc or component type

AYORZ Hollerith input variable; the letter Y or Z to indicate which definition is to be used when aliasing (blank when not)

BLCOEF(I,N,M) I = 1 to 7; defining mathematical parameters for each segment and BL model

I = 1: x_1

I = 2: v_1

I = 3: A^2

I = 4: B^2

I = 5: C

I = 6: x_2

I = 7: v_2

BLMDEE(I,N,M) I = 1 to 8; points used to define each segment and BL model

I = 1: x_1

I = 2: v_1

I = 3: x_2

I = 4: v_2

I = 5: x_{3L}

I = 6: v_{3L}

I = 7: x_{3R}

I = 8: v_{3R}

(x_1, v_1) and (x_2, v_2) are the initial and final points, respectively, of the given segment. (x_{3L}, v_{3L}) establishes the slope at the initial side, (x_{3R}, v_{3R}) establishes the slope at the terminal side

BLMMAX(I)	I = 1 to KNTBLM; maximum x for each BL model
BLMMIN(I)	I = 1 to KNTBLM; minimum x for each BL model
BLMNAM(M)	Alphanumeric name of each BL model
BLMNYZ(M)	Alphanumeric y or z coordinate specification for each BL model
BNAME	Hollerith input variable; body line/control point name which is to be defined
BTITLE(I,II)	not used currently
BYORZ	Hollerith input variable; the letter Y or Z to indicate which data coordinate definition is to follow
COMPNM(I)	I = 1 to KCOMP; component names (alphanumeric)
CPNTNM(I)	I = 1 to KCPNT; control point names (alphanumeric)
CTITLE(I,K)	I = 1 to 10; alphanumeric CS model title or comments
D(1)	Input variable; if type is PIECE or FLINK, this is x_1. If type is ALINK, PATCH, or FILET, this is a floating point number equal to KSEG of the segment from which x_1 and/or v_1 are to be determined.
D(2)	Input variable; if type is PIECE or FLINK, this is v_1. If type is ALINK, PATCH, or FILET, this is a floating point number equal to KSEG of the segment from which t_1 is to be determined.

D(3) Input variable; if type is PIECE or ALINK, this is x_2. If type is FLINK, PATCH, or FILET, this is a floating point number equal to KSEG of the segment from which x_2 and/or v_2 are to be determined.

D(4) Input variable; if type is PIECE or ALINK, this is v_2. If type is FLINK, PATCH, or FILET, this is a floating point number equal to KSEG of the segment from which t_2 is to be determined.

D(5) Input variable; if SLP1 is blank:

If type is FILET, this is x_1; y_1 and t_1 are to be determined from the segment specified by D(1) and D(2). If type is other, this is x_3.

If SLP1 is other than blank, see definition of SLP1.

D(6) Input variable; if SLP2 is blank:

If type is FILET, this is x_2; y_2 and t_2 are to be determined from the segment specified by D(3) and D(4).

If type is other, this is x_3.

If SLP2 is other than blank, see definition of SLP2.

HDEL Input variable; increment size in degrees to establish interrogation points between HGO and HEND; not required for modes 1 or 3.

HEND Input variable; final value of theta (in degrees) to be interrogated; not required for modes 1 or 3.

HGO Input variable; initial value of theta (in degrees) to be interrogated; not required for modes 1 or 3.

HNOW Current value of θ in degrees (used in various exercising routines; e.g., MODE1, MODE2, etc).

HNOWR Current value of θ in radians (used in various exercising routines, e.g., MODE1, MODE2, etc).

IAMD IABS(MODE)

IANDV IABS(NDERV)

IBLCOR(I,J) I = 1 to 6; body line coordinate index for Y1(I = 1), Z1(I = 2), Y2(I = 3), Z2(I = 4), Y3 and/or Y4(I = 5), Z3 and/or Z4(I = 6).

IBLMIX(M) Index to the control point coordinate for which this BL model was first defined.

IBLMWD(I,N,M) I = 1 to 4; indicator for the shape (I = 1), type (I = 2), mode of definition (I = 3), and freed constraints (I = 4) of each segment and BL model

IBLMX(I) I = 1 to NBLCOR; index of the body line model for the I^{th} coordinate control point

IBLSSH(N,M) Shape index for each segment and BL model
= (1) LINE, (2) CIRC - not used, (3) ELLX, (4) ELLY, (5) XPAR, (6) YPAR, (7) RXPA, (8) RYPA, (9) CUBI, (10) ALL - not used, (11) NULL

IBLSX(I) I = 1 to KNTBLM: current segment number index for each BL model.

ICOMPX(J,K) Index of the component definition for each arc and CS model

ICRITE Output unit for error and checking messages, primarily for use on a time sharing computing system, otherwise, ICRITE = IRITE

ICSACC(I,J,K) I = 1,2; controlling component index for each arc and CS model

I = 1: information pertains to forward end of arc
I = 2: information pertains to aft end of arc
= -1: end of arc is unaffected
> 0: gives index of another arc which is to intersect the J^{th} arc for growing pieces, or which is to supply filleting information if J^{th} arc is a fillet.

ICSACP(I,J,K) I = 1 to 3: control point index for each arc and CS model
I = 1: initial point of arc
I = 2: final point of arc
I = 3: slope control point for arc

ICSAFR(J,K) Free constraint index for each arc and CS model (not currently used)

ICSASH(J,K) Shape index for each arc and CS model

= 1: LINE

= 2: CIRC, circle (not currently available)

= 3: ELLI, in-facing ellipse (concave to origin)

= 4: ELLO, out-facing ellipse (convex to origin)

ICSASQ(J,K) Sequencing index to establish the order in which cross sectional arcs are to be defined.

ICSATY(J,K) Type index for each arc and CS model

= 1: PIEC, piece

= 2: FLIN, forward link

= 3: ALIN, aft link

= 4: PATC, patch

= 5: FILE, fillet

ICSMX(KMODEL) Index of current CS model (from 1 to NCSM), describes use of library of CS models as applied to this vehicle.

IFREE Input variable; index of the datum quantity which is to be "free," i.e., determined by the code. IFREE ranges from 1 to 6 corresponding to x_1, v_1, x_2, v_2, t_1, t_2, as ordered. A line must have any one of these free; an x- or y-parabola must have either 5 or 6 free; other curves should have IFREE = 0.

IN(J) Indicator for each arc of the current CS model

= -1: arc not included at this station

= 1: arc included at this station

IPLOT I/O unit for plot mode output from GEMCHK, MODE1, MODE2, etc.

IPUNCH Assumed punch unit (= 7)

IREAD Input unit

IRITE Output unit

ISPEC(I,J,K) I = 1, 2; index to indicate what coordinate is to be specified at the initial control point (I = 1) and the final control point (I = 2)

= 1: y is to be specified (z is to be computed on the controlling component)

= 2: z is to be specified (y is to be computed on the controlling component).

= -1: for nonfillets

ITAPE I/O unit for QUICK intermediate data deck (math model) (note: called INREAD in GEOMIN)

IUORDR(J) Use order index to establish sequence of CS arcs after intersections and filets are completed.

IZBDEX(K) Index of the bottom center body line model for each CS model.

IZCDEX Index of the center body line model (mapaxis)

IZTDEX(K) Index of the top center body line model for each CS model.

J Index of current cross sectional arc for a given CS model (K) from 1 to KNTCSA(K).

JSEQ Input variable; definition sequence (order in which the CS arcs are to be defined)

K Index of current cross sectional definition (library) model (from 1 to NCSM)

KARC Input variable; number of arcs in current cross sectional model.

KCOMP Number of components used to define all CS models (entire vehicle).

KCPNT Number of control points used to define all CS models (entire vehicle).

KDUM Input variable; running count of the current cross section model.

KMODEL	Index of current cross sectional use model (from 1 to KNTCSM)
KNTARC	Number of arcs in the CS model corresponding to the current station
KNTBLM	Number of body line models
KNTBLS(M)	Number of segments for each body line model
KNTCSA(K)	Number of arcs for each cross sectional model
KNTCSM	Number of applications of cross section models to define entire vehicle
KSEG	Input variable; the order (in increasing x) in which this segment appears in this body line model. A KSEG = -1 (further arguments not required) terminates the data for a given body line.
KZBDEX	Control point index for bottom centerline.
KZCDEX	Control point index for mapaxis.
KZTDEX	Control point index for top centerline.
M	Index of current body line definition model (from 1 to KNTBLM)
MODE	Input variable; = ±1, creates body line traces = ±2, creates cross sectional cuts = +3, interrogates cross sections in neighborhood of control points -3, allows multiple body line traces to create plan and profile views = +4, comparison of analytic derivatives with numerically formed derivatives = +5, check of unit vectors normal to body surface

= +6, exercises modes 1, 2, and 3 at the limits of each cross sectional model

-6, exercises modes -2 and -7 at the limits of each cross sectional model

= -7, (plotting mode only) creates cross sectional cuts, but includes all arcs in their entirety (including growing pieces still contained within the basic skin)

MODEL — Index to the current CS library model definition

N — Index of current body line segment for a given BL model (M) from 1 to KNTBLS(M).

NBLCOR — Number of control point coordinates to define entire vehicle (y and z are distinct, thus NBLCOR = 2*KCPNT).

NCSM — Input variable; number of distinct cross section models.

NDERV — Input variable;

= $\pm$N, where N is the order of derivative to be calculated (N = 0, 1, or 2)

= +N, should always be used for checkout interrogations (means each call to a given location is new, thus the radius and all temporary variables must be computed)

= -N, should not be used for checkout interrogations; requires previous call to same location (x and θ); radius and certain temporary variables are not recomputed.

NHPTS — Number of θ points (used in various exercising routines; e.g., MODE1, MODE2, etc).

NXPTS — Number of x-stations (used in various exercising routines; e.g., MODE1, MODE2, etc).

PNTNAM(1) — Hollerith input variable; control point name for the beginning of the arc currently being defined.

PNTNAM(2) Hollerith input variable; control point name for the termination of the arc currently being defined.

PNTNAM(3) Hollerith input variable; slope control point name for the current arc when required, blank if not.

SDEF Hollerith input variable; segment definition mode (currently, only two point, two slope/slope control point method is available - input "KV").

SLP1 Hollerith input variable;

= blank yields no effect

= S when following item, D(5), is to be explicit t_1

= A when following item, D(5), is to be arctan t_1 (in degrees)

SLP2 Hollerith input variable;

= blank yields no effect

= S when following item, D(6), is to be explicit t_2

= A when following item, D(6), is to be arctan t_2 (in degrees)

SSHAPE Hollerith input variable; segment shape (including NULL, in which case this segment is essentially deleted, and no further parameters are required)

STYPE Hollerith input variable; segment type

THETA1(J) Value of θ at the initial control point location for each arc (at the current x-station)

THETA2(J) Value of θ at the final control point location for each arc

TITLE Hollerith input; any comments

UNX x-component of surface unit normal

UNY y-component of surface unit normal

UNZ z-component of surface unit normal

UTHET1(J)	Initial use θ for each arc (as affected by intersections and fillets)
UTHET2(J)	Final use θ for each arc
V(M)	Current (latest x-station) computed value of each BL model
VTITLE(I)	I = 1 to 15; alphanumeric vehicle or run title
VX(M)	Current computed slope (dv/dx) of each BL model
VXX(M)	Current computed derivative (d^2v/dx^2) of each BL model
W(I,J)	I = 1 to 4; defining mathematical parameters for each CS arc at a given station: $R_o(I = 1)$, $\theta_o(I = 2)$, $A^2(I = 3)$, $B^2(I = 4)$
WX(I,J)	I = 1 to 5; for I = 1 to 4, WX(I,J) = d(W(I,J))/dx; WX(5,J) = dr/dx for internal computations only
WXX(I,J)	I = 1 to 4; d(WX(I,J))/dx
XCSMS1(KK)	Starting x-station of the current cross section model
XCSMS2(KK)	Ending x-station of the current cross section model
XDEL	Input variable; increment size in x, to establish output stations between XGO and XEND
XEND	Input variable; final x-station to be interrogated
XGO	Input variable; initial x-station to be interrogated
XNOW	Current x-station (used in various exercising routines; e.g., MODE1, MODE2, etc).
Y1(J)	y of initial point for each CS arc
Y1X(J)	dY1(J)/dx
Y1XX(J)	$d^2Y1(J)/dx^2$
Y2(J)	y of final point for each CS arc

Y2X(J)	$dY2(J)/dx$
Y2XX(J)	$d^2Y2(J)/dx^2$
Y3(J)	y of slope control point for forward (initial) end of each CS arc
Y3X(J)	$dY3(J)/dx$
Y3XX(J)	$d^2Y3(J)/dx^2$
Y4(J)	y of slope control point for aft (final) end of each CS arc
Y4X(J)	$dY4(J)/dx$
Y4XX(J)	$d^2Y4(J)/dx^2$
ZCL(I)	I = 1 to 3; current value (z) of bottom center line (I = 1), top center line (I = 2), and mapaxis (I = 3)
ZCLX(I)	I = 1 to 3; current slope (dz/dx) of bottom center line (I = 1), top centerline (I = 2), and mapaxis (I = 3)
ZCLXX(I)	I = 1 to 3; current second derivative (d^2z/dx^2) of bottom centerline (I = 1), top centerline (I = 2), and mapaxis (I = 3)
ZMAPNM	Name of mapaxis
Z1(J)	z of initial point for each CS arc
Z1X(J)	$dZ1(J)/dx$
Z1XX(J)	$d^2Z1(J)/dx^2$
Z2(J)	z of final point for each CS arc
Z2X(J)	$dZ2(J)/dx$
Z2XX(J)	$d^2Z2(J)/dx^2$
Z3(J)	z of slope control point for forward (initial) end of each CS arc
Z3X(J)	$dZ3(J)/dx$
Z3XX(J)	$d^2Z3(J)/dx^2$

Z4(J)	z of slope control point for aft (final) end of each CS arc
Z4X(J)	dZ4(J)/dx
Z4XX(J)	$d^2Z4(J)/dx^2$

SYMBOL LIST FOR STEIN

AAA, BBB, CCC, DDD, EEE, FFF, AAAZ, BBBZ, CCCZ, DDDZ, EEEZ, FFFZ, AAAZZ, BBBZZ, CCCZZ, DDDZZ, EEEZZ, FFFZZ	The coefficients of the conformal mappings, and their first and second derivatives with respect to z
ACH	Free stream Mach number
APINF	Dimensional free-stream pressure (Note: dimensions must be consistent with choice of length scale; this is for the computation of aero-coefficients only.) Currently not used - leave blank.
AR(I,J)	I = 1 to KCOMP, J = 1 to KPIECE(I); integrated surface area for each component and piece
AREF	Reference area for aerodynamic coefficients
ARINF	Dimensional free-stream density (see note for APINF). Currently not used, leave blank.
ATTACK	Angle of attack (input in degrees)
B(M)	Radial position of the body in the mapped plane
BHH(M)	Second derivative of body radius with respect to θ
BHZ(M)	Cross derivative of body radius with respect to ζ and θ
BN(M)	Radial position of body in mapped plane at Z + DZ
BZZ(M)	Second derivative of body radius with respect to ζ
B2, B2Z	y position (in the physical plane) of the wing tip and its derivative with respect to z (Fig. 5)
C(M,L), CH(M,L), CZ(M,L)	Radial position of shock L in the mapped plane and its derivatives with respect to θ and ζ (mapped coordinates)

CC(M,L), CCY(M,L), CCZ(M,L)	Radial position of L^{th} wing type shock surface (CC(M,1) = B(M) and CC(M,L) = C(M,L-1) (L = 2 ... LC +1)) and its derivatives with respect to Y and Z
CFTITL(I)	I = 1 to 5; alphanumeric request for aerodynamic coefficients (i.e., CL, CD, CM, CN, and CA).
CMPTTL(I)	I = 1 to KCOMP, CMPTTL (KCOMP + 1) = TOTL (total); alphanumeric title for each component (above)
CN(M,L), CHN(M,L), CZN(M,L)	Radial position of shock L in the mapped plane at Z + DZ and its derivatives with respect to θ and $\mathcal{Z}$ (mapped coordinate at Z + DZ)
CONE	Cone half angle (input in degrees); only used for sharp cone calculations
DX(L)	Mesh spacing in the radial direction, in region L.
DY(I)	Mesh spacing in the circumferential direction, in region I.
DZ	Step size
DZFAC	Factor multipling DZ computed from CFL stability condition (usually DZFAC = .7)
DZGEOM	Interval for geometry test
DZWRIT	Interval for printed output
ERR(J)	J^{th} error generated in an iteration
GAMFR	Equivalent ratio of specific heats (γ) for frozen flow
GAMIN	Free stream ratio of specific heats (γ)
GAMLO (N,M)	Local value of $a^2/(p/\rho)$ (a= speed of sound, p = pressure, ρ = density)
H(N,M)	Mapped space polar angle
HCX(N,I), HCZ(N,I)	The X and Z derivatives of the I^{th} cross flow surface

HFN(I,J)	Same as HFO(I,J) but at current station (see Fig. 8)
HFO(I,J)	I = 1 to KCØMP, J = 1 to KPIECE(I); final value of θ′ defining each component and piece at previous station
HHL(M)	Value of θ in the mapped plane of entropy layer surface point M (Fig. 3)
HIN(I,J)	Same as HIO(I,J) but at current station (see Fig. 8)
HIO(I,J)	I = 1 to KCØMP, J = 1 to KPIECE(I); initial value of θ′ defining each component and piece at previous station
HO(M)	Cylindrical θ′ at mesh points on the body at Z (see Fig. 8)
HS(N,I), HSR(N,I), HSZ(N,I)	Circumferential position of cross flow surface I and its derivatives with respect to r and z at Z
HSN(N,I), HSRN(N,I), HSZN(N,I)	Circumferential position of cross flow surface I in the mapped plane and its derivative with respect to r and z at Z + DZ
HST	Free stream total enthalpy
H1(M)	Metric factor h_1 (spreading of streamlines) at Z
H1N(M)	Metric factor h_1 at Z + DZ
I	Counter for regions in the circumferential direction; I = 1 in the region adjacent to the bottom symmetry plane and I = IC is the region adjacent to the top symmetry plane. I is also a counter for cross flow type surfaces (I = 1, bottom symmetry plane; I = IC + 1, top symmetry plane).
IAERD	Indicator: IAERD = 0: Integrated forces and moments on the body are not read, and are set to 0. (This would be used to start an aero-coefficient run) IAERD = 1: Integrated forces and moments on the body are read. (This would be used to continue an aero-coefficient run)

IAERO	Indicator:
	IAERO = 0: No aero-coefficients to be computed
	IAERO = 1: At least one aero-coefficient to be computed
IBLOUT	Output (tape) unit for streamline/boundary layer code and sonic boom code - set equal to 0 if no boundary layer inputs are to be computed.
IBUG	Output indicator - IBUG = 0: no intermediate output, IBUG = 1: for intermediate output
IC	Number of regions in the circumferential direction
ICASE	Indicator - ICASE = 1: Initial flow field data are not read but computed in the code (i.e., first run for sharp nose vehicles)
	ICASE = 2: starting plane data will be read (i.e., first run for blunt nose body or continuation run)
ICF(K)	K = 1 to 5; indicates request and name location for each aerodynamic coefficient (K = 1 for CL, K = 2 for CD, K = 3 for CM, K = 4 for CN, and K = 5 for CA)
	ICF(K) = -1: coefficient not requested
	ICF(K) = N > 0: coefficient requested and
	CFTITL(N) = proper alphanumeric coefficient name (CL, CD, etc).
	(If ICF(3) = 4, then CM is to be computed and CFTITL(4) = 'CM')
IDIMEN	Maximum number of regions in the I direction
IENT(M)	Indicator for entropy layer IENT(M) = 0: surface not detected yet at M, IENT(M) = 1: surface detected at M, IENT(M) = 2: surface collapsed to body at M.

IENTE	Indicator IENTE = 0: no entropy layer to be detected, IENTE = 1: entropy layer to be detected. IENTE is set equal to 2 when an entropy layer is started.
IFCP(I,J)	I = 1 to KCOMP, J = 1 to KPIECE(I); final control point (in θ) for each component and piece (determined from QUICK modeling)
IGAS	Indicator; IGAS = 0: ideal gas; IGAS = 1: equilibrium; is set equal to 2 at Z = ZREEZ (freezing station)
IHS	Indicator: IHS = 0: metric factor h_1 not computed IHS = 1: h_1 initial plane data read and computed IHS = -1: h_1 initial plane data not read, but initialized by code to the body radius at each mesh point and computed
III	Indicator: III = 0: No component pieces were found between this Z and Z + DZ III = 1: At least one component piece was found between this Z and Z + DZ
INCP(I,J)	I = 1 to KCOMP, J = 1 to KPIECE(I); initial control point (in θ) for each component and piece (determined from QUICK modeling)
IPUNCH	Output unit for starting plane data for next run
IREADO	Set to 5 in data statement in INIT-read unit for read #1
IREAD1	Read unit for control data 1
IREAD2	Read unit for control data 2
IREAD3	Read unit for starting plane data
IREAD4	Read unit for QUICK intermediate data

ISHBEG(J) Indicators: J = 1 denotes the bottom symmetry plane, J = 2 wing plane, and J = 3 the top symmetry plane. ISHBEG(J) = 0 no sharp leading edge at the J^{th} plane. ISHBEG(J) = 1 there is a sharp leading edge at the J^{th} plane but the shock has not been detected yet. ISHBEG(J) is set equal to 2 when the shock has been detected. ISHBEG(J) is set equal to 3 when the shock is in.

ISHOK(M,L) Wing type shocks surface indication for shock L at M

ISHOK(M,L) = 0: arbitrary surface

ISHOK(M,L) = 1: shock point (detached)

ISHOK(M,L) = 2: sharp leading edge shock point

ISHTIP Indicator: ISHTIP = 0 no sharp leading edges; ISHTIP ≠ 0 sharp leading edges exist on the geometry

IWRIT Output unit for printed flow field data

IZ(I,J) I = 1 to KCOMP, J = 1 to KPIECE(I); Indicator:

= 0: Component piece is not present between this Z and Z + DZ

= 2: Component piece is present between this Z and Z + DZ and, thus, must be integrated over

JA Maximum number of steps between printed output

K Step counter, K = 0 at starting plane for each run

KA Maximum number of steps before punching output and stopping run

KCOMP Number of individual components for which aero-coefficients are to be computed

KNTCAL Number of consecutive calls to AEROCF from ARCONT; significant for initialization procedures

KPIECE(I) I = 1, KCOMP; see NP

L	Counter for regions in the radial direction; L = 1 is the region closest to the body, L = LC is the region closest to the bow shock. L is also a counter for wing type shocks (L = --1 inner most and L = LC--bow shock). Finally L is used as a counter for radial dividing surfaces (i.e., L = 1 => body and L = LC + 1 => bow shock.)
LC	Number of regions in the radial direction
LDIMEN	Maximum number of regions in the L direction
LOOP	Indicator: LOOP = 0: level one of the MacCormack scheme LOOP = 1: level two of the MacCormack scheme LOOP = 100: print one more station and stop
M	Counter in the circumferential direction; M = 1 is the bottom symmetry plane and M = MC(IC) + MREG(IC) is the top symmetry plane
MC(I), NSHK1(I), NSHK2(I), MREG(I)	Correspond to NC(L), MSHK1(L), MSHK2(L). NREG(L) but for cross flow type surfaces
MCIR	Minimum number of points in the "M" direction in any region I (usually MCIR = 5)
MC1	Number of points in the "M" direction in region I = 1.
MDIMEN	Maximum number of points in the "M" direction
MDZ	The value of M at which the minimum step size was found
MSHK1(L), MSHK2(L)	Values of M at end shock points of shock L (Fig. 4)
MSHOK(N,I)	Crossflow shock surface indicator MSHOK(N,I) = 0: arbitrary surface

MSHOK(N,I) = 1: cross flow shock point

MSHOK(N,I) = 2: for points at a sharp leading edge shock

N	Counter in the radial direction (Fig. 2); N = 1 is the body and N = NC(LC) + NREG(LC) is the bow shock
NC(L)	Number of points in region L (radial direction)
NC1	Number of points in the radial direction in region L = 1
NDIMEN	Maximum number of points in N direction
NDZ	The value of N at which the minimum step size was found
NLOOK	Indicator: = 0: wing type shock is first detected in any circumferential region I. = 1: wing type shock is first detected in region I = 1. = 2: wing type shock is first detected outside of region I = 1.
NP	Number of pieces or segments into which a given aerodynamic component is to be divided (stored in KPIECE(I), I = 1 to KCOMP)
NREG(L)	NREG(L) = NC(L-1) + NREG(L-1) (NREG(1) = 0)
NRUN	Run number, used to order runs
NSOUT	Number of specific values of z at which there is to be printed output (NSOUT $\leq$ 10)
P(N,M)	$\ell n(p/p_\infty)$ (where p is the pressure)
PFT(I,J,K)	I = 1 to KCOMP, J = 1 to KPIECE(I), K = 1, 2, 3; x, y, and z components, respectively, of the integrated pressure force for each component and piece
PHL(M)	$\ell n(p/p_\infty)$ on the entropy layer surface (Fig. 3)

PHLN(M)	$\ell n(p/p_\infty)$ on the entropy layer surface (Fig. 3) at Z + DZ
PIN	p_∞/p_{SL} (free stream pressure/sea level pressure)
PMT (I,J,K)	I = 1 to KCOMP, J = 1 to KPIECE(I); Cartesian components of the integrated moments for each component and piece
PN(N,M)	$\ell n(p/p_\infty)$ (where p is the pressure) at Z + DZ
PO(M)	In ARCONT and AEROCF only, $\ell n(p/p_\infty)$ on the body at Z
PO(N,M)	$\ell n(p/p_\infty)$ (where p is the pressure) at Z - DZ
R(N,M)	Mapped space radial coordinate
RHL(M)	Radial position of the entropy layer surface (Fig. 3)
RHLN(M)	Radial position of entropy layer surface in the mapped plane at M and Z + DZ (Fig. 3)
RQRI	Ratio of the freezing plane gas constant to its free stream value
S(N,M)	Entropy
SFR	Reference entropy at the freezing plane
SHL(M)	Entropy on the entropy layer surface (Fig. 3)
SHLN(M)	Entropy on the entropy layer surface (Fig. 3) at Z + DZ
SN(N,M)	Entropy at Z + DZ
SO(N,M)	Entropy at Z - DZ
T(N,M)	Local value of (pressure/density)
TIN	T_∞/T_{SL} (free stream temperature/sea level temperature)
TRY(J)	J^{th} guess in an iteration
U(N,M), V(N, M), W(N,M)	Cartesian velocity components
UHL(M), VH (M), WHL(M,	Cartesian velocity components on the entropy layer surface (Fig. 3)

UHLN(M), VHLN(M), WHLN(M)	Cartesian velocity components on the entropy layer surface (Fig. 3) at Z + DZ
UN(NM), VN(N, M), WN(N,M)	Cartesian velocity components at Z + DZ
UNOR(I,J)	The three (UNOR(1,J), UNOR(2,J) and UNOR(3,J)) Cartesian components of the unit normal to the body at the J^{th} sharp leading edge (Fig. 6)
VIN	Free stream velocity
VMO(I)	I = 2, 3; y and z positions of line about which moments are computed
X(NN,L)	Computational plane coordinate (X(1,L) = 0 and X(NC(L),L) = 1) (Fig. 2)
XO(M)	Cartesian x′ at mesh points on the body at Z (see Fig. 7)
XTIP, XTIPZ	x position (in the physical plane) of the wing tip and its derivative with respect to z (Fig. 5)
Y(MM,L)	Computational plane coordinate (Y(1, I) = 0 and Y(MC(I),I) = 1) (Fig. 2)
YD, YDZ, YB, YBZ	Position and z derivatives in the physical space, in the symmetry plane, of the top and bottom of the body. These roles depend on whether the configuration is high wing or low wing (Fig. 5)
YO(M)	Cartesian y′ at mesh points on the body at Z (see Fig. 7)
Z	Axial station
ZCOMP	z station immediately prior to start of sharp leading edge
ZEND	Last axial station to be computed before punching output and stopping

ZFINL(I,J)	I = 1 to KCØMP, J = 1 to KPIECE(I); final station (z) for each component and piece (Note: ZINIT and ZFINL may overlap or coincide for different pieces of the same component, thus allowing for disjoint cross sectional members)
ZFREEZ	Value of z at which the thermodynamics is to be converted from equilibrium to frozen.
ZGEOM1	First axial station at which a "geometry test" will be printed
ZGEOM2	Last station of geometry test
ZINIT(I,J)	I = 1 to KCOMP, J = 1 to KPIECE(I); initial station (z)
ZMADD, MDEL	MDEL points will be added at Z = ZMADD. In the circumferential direction
ZMAP1, ZMAP2	The conformal mappings are not used for $Z \leq$ ZMAP1 and they are fully developed for $Z \geq$ ZMAP2. (ZMAP1 = starting plane station for the first supersonic flow run and ZMAP2 = ZMAP1 + a number of nose radii, usually)
ZN	Updated axial station ZN = Z + DZ
ZNADD, NDEL	NDEL points will be added at Z = ZNADD. In the "radial" direction
ZO	Z (in ARCONT and AEROCF)
ZSHRP	z station immediately following start of sharp leading edge ($\approx$ ZCOMP)
ZSOUT(I)	Specific values at z at which there is to be printed output I = 1 $\rightarrow$ NSOUT (if NSOUT $\leq$ 0 no values of ZSOUT are read or stored)
ZSTART	Starting value of z for run
ZTIPS	Value of z at which wing tip surface (Fig. 7) is inserted (usually ZTIPS $\leq$ ZWING). This surface is used to control the grid.

ZWING	Axial station at which wing starts (used in mappings) (Fig. 7)
ZWRIT1	Axial station at which output is begun (ZWRIT1 $\geq$ ZSTART usually)
ZWRIT2	Last axial station at which output is printed (ZWRIT2 $\leq$ ZEND usually)
Z1MSH, Z2MSH	Same as Z1NSH, Z2NSH but for cross flow shocks (See Fig. 7)
Z1NSH, Z2NSH	A wing type shock will be looked for between z = Z1NSH(J) and z = Z2NSH(J). After detection, Z1NSH(J) is set to 1×10^6 and Z2NSH(J) is set to -1×10^6 so that shock J is not found again. (See Fig. 7)

SYMBOL LIST FOR STRMBL

ACHINF	Free stream mach number, read from data tape
ATTACK	Angle of attack, in degrees, read from tape
DZ	Current step size, Δz
DZO	previous step size
FNU	Nondimensional kinematic viscosity $FNU = \nu_\infty' = \bar{\nu}_\infty/\bar{\nu}_{ref}$ where $(\bar{\ })$ = dimensional quantity and $\bar{\nu}_{ref} = \sqrt{\bar{p}_\infty/\bar{\rho}_\infty}$
GAMMA	Free stream ratio of specific heats, read from tape
HCUT(IS,ICUT)	θ' -location of each streamline at each cut for body normals
HP(N,M)	Angle from x' -axis (see Fig. 8) to mesh points (θ' in Fig. 8)
HPO(N,M)	HP(N,M) at previous data plane
HZNP	$d\theta'/dz$ for the current streamline and data plane
HZOP(IS)	$d\theta'/dz$ for each streamline at the previous data plane
H1(M)	Metric coefficient h_1 at mesh points on the body
H1S	Metric coefficient h_1 for the current streamline and data plane
IC	Number of regions in the circumferential direction*
ICO	IC at previous data plane
ICUT	Indicator of current pseudo-stream-surface normal cut, from 1 to NCUT

*As in STEIN

ICUTMX	Largest ICUT currently in storage
IDUM1, IDUM2, IDUM3	Not used
IIC(IICUT)	Indicates which ICUT (= IIC(IICUT)) is currently stored in location referred to by IICUT
IICUT	Index (between 1 and NIICUT) to dynamic storage locations for pseudo-stream-surface data
INPT	Index/counter for points taken along body surface normals, from 1 to NNPT
IR	Read unit for card input
IRT	Not currently in use
IS	Streamline index/counter, from 1 to NS
ITP	I/O unit for data tape input
IW	Write unit for printed output
JCUT	Output and pseudo-stream-surface (p-s-s) parameter, normals to body are taken and p-s-s data is output every JCUT data planes
JS	Output parameter, streamline flow variables are output every JS data planes
LC	Number of regions in the radial directions*
LCO	LC at previous data plane
M	Circumferential mesh point counter, from 1 to MC(IC) + MREG(IC)
MC(I)	I = 1 to IC; number of points in region I (circumferential direction)*
MCO(I)	MC(I) at previous data plane

*As in STEIN

MREG(I)	MREG(I) = MC(I-1) + MREG(I-1), MREG(1) = 0*
MREGO(I)	MREG(I) at previous data plane
N	Radial mesh point counter, from 1 to NC(LC) + NREG(LC)
NC(L)	L = 1 to LC; number of points in region L*
NCO(L)	NC(L) at previous data plane
NCUT	Number of pseudo-stream-surface normal cuts
NFLG(INPT,IS, IICUT)	Flag set to indicate whether a point on the normal for a given streamline has been computed (= 1) or not (= -1)
NIICUT	Number of cuts permitted to be in storage simultaneously (must be sufficiently large, now equal to 5, to prevent body normal from the K + NIICUT data plane from extending past the K data plane or vice versa)
NNPT	Number of points taken along body surface normal to establish data in pseudo-stream-surface
NREG(L)	NREG(L) = NC(L-1) + NREG(L-1), NREG(1) = 0*
NREGO(L)	NREG(L) at previous data plane
NS	Number of streamlines to be traced (up to 50)
NUM(IICUT)	Number of points successfully computed for the IICUT cut (when NUM(IICUT) = NS*NNPT, all points on all normals taken at the ICUT corresponding to this IICUT have been computed, and thus may be output and the storage locations used for the next cut)
P(N,M)	$\ln(\bar{p}/\bar{p}_\infty)$ at mesh points (where $\bar{p}$ is pressure)
PI	π

*As in STEIN

PNORM(INPT,IS, IICUT)	$\bar{p}/\bar{p}_\infty$ (where $\bar{p}$ is pressure) at each point along the normal to each streamline for each cut currently being stored
PO(N,M)	P(N,M) at previous data plane
PS	$\ell n(\bar{p}/\bar{p}_\infty)$ for the current streamline and data plane (where $\bar{p}$ is pressure)
RP(N,M)	Radial distance from mapaxis (B_2 line) to mesh points (r' in Fig. 7)
RPO(N,M)	RP(N,M) at previous data plane
S(N,M)	entropy at mesh points
SLNG(IS)	Integrated arc length along each streamline
SNORM(INPT,IS, IICUT)	Entropy stored the same as PNORM(INPT, IS, IICUT)
SO(N,M)	S(N,M) at previous data plane
SR(IS)	r for each streamline
SS	entropy for the current streamline and data plane
STHE(IS)	θ for each streamline
TESTA	Angle of attack, in degrees, read from cards
TESTG	Free stream ratio of specific heats, read from cards
TESTM	Free stream mach number read from cards
TESTZ	Initial value of z, read from cards
THEOP(IS)	θ' for each streamline
U(N,M)	x-velocity component at mesh points
UNORM(INPT, IS, IICUT)	x-component of velicty stored the same as PNORM(INPT, IS, IICUT)

UNX	x-component of body surface unit normal
UNY	y-component of body surface unit normal
UNZ	z-component of body surface unit normal
UO(N,M)	U(N,M) at previous data plane
US	x-component of velocity for the current streamline and data plane
V(N,M)	y-velocity component at mesh points
VNORM(INPT, IS,IICUT)	y-component of velocity stored the same as PNORM(INPT,IS, IICUT)
VO(N,M)	V(N,M) at previous data plane
VS	y-component of velocity for the current streamline and data plane
W(N,M)	z-velocity component at mesh points
WNORM(INPT, IS,IICUT)	z-component of velocity stored the same as PNORM(INPT,IS, IICUT)
WO(N,M)	W(N,M) at previous data plane
WS	z-component of velocity for the current streamline and data plane
YCL(J)	J = 1 to 3; y-position of body bottom center line (J = 1), body top centerline (J = 2), mapaxis or B_2 line (J = 3)
YCLZ(J)	dYCL(J)/dz
YCLZZ(J)	d^2YCL(J)/dz^2
Z	Current z
ZCUT(ICUT)	z-locations at which cuts for body normals were made
ZO	z at previous data plane
ZSTAR	Initial value of z, read from tape

SYMBOL LIST FOR BOOM

ACHINF	Free stream Mach number, read from data tape
ATTACK	Angle of attack, in degrees, read from tape
DZ	Current step size, Δz
GAMMA	Free stream ratio of specific heats, read from tape
HP(N,M)	Angle from x′ -axis (see Fig. 8) to mesh points (θ' in Fig. 8)
IC	Number of regions in the circumferential direction*
IR	Read unit for card input
IRT	Not currently in use
ITP	I/O unit for data tape input
IW	Write unit for printed output
JA	Output parameter, data are computed and output every JA data planes
KZBDEX, KZTDEX, KZCDEX	See symbol list for QUICK (not used here)
LC	Number of regions in the radial direction*
M	Circumferential mesh point counter, from 1 to MC(IC) + MREG(IC)*
MC(I)	I = 1 to IC; number of points in region I (circumferential direction)*
MREG(I)	MREG(I) = MC(I-1) + MREG(I-1), MREG(1) = 0*
N	Radial mesh point counter, from 1 to NC(LC) + NREG(LC)*
NC(L)	L = 1 to LC; number of points in region L (radial direction)*

*As in STEIN

NHPTS	Number of circumferential points on the data cylinder at which values of the flow variables are to be determined
NREG(L)	NREG(L) = NC(L-1) + NREG(L-1), NREG(1) = 0*
P(N,M)	$\ell n(p/p_\infty)$ at mesh points (where p is pressure)
RCYL	Radius of data cylinder
RP(N,M)	Radial distance from mapaxis (B_2 line) to mesh points (r′ in Fig. 8)
S(N,M)	entropy at mesh points
TESTA	Angle of attack, in degrees, read from cards
TESTG	Free stream ratio of specific heats, read from cards
TESTM	Free stream Mach number, read from cards
TESTZ	Initial value of z, read from cards
U(N,M)	x-velocity component at mesh points
V(N,M)	y-velocity component at mesh points
W(N,M)	z-velocity component at mesh points
YCL(J)	J = 1 to 3; y-position of body bottom center line (J = 1), body top centerline (J = 2), mapaxis or B_2 line (J = 3)
YCLZ(J)	dYCL(J)/dz
YCLZZ(J)	d^2YCL(J)/dz^2
Z	Current z
ZSTAR	Initial value of z, read from tape

*As in STEIN

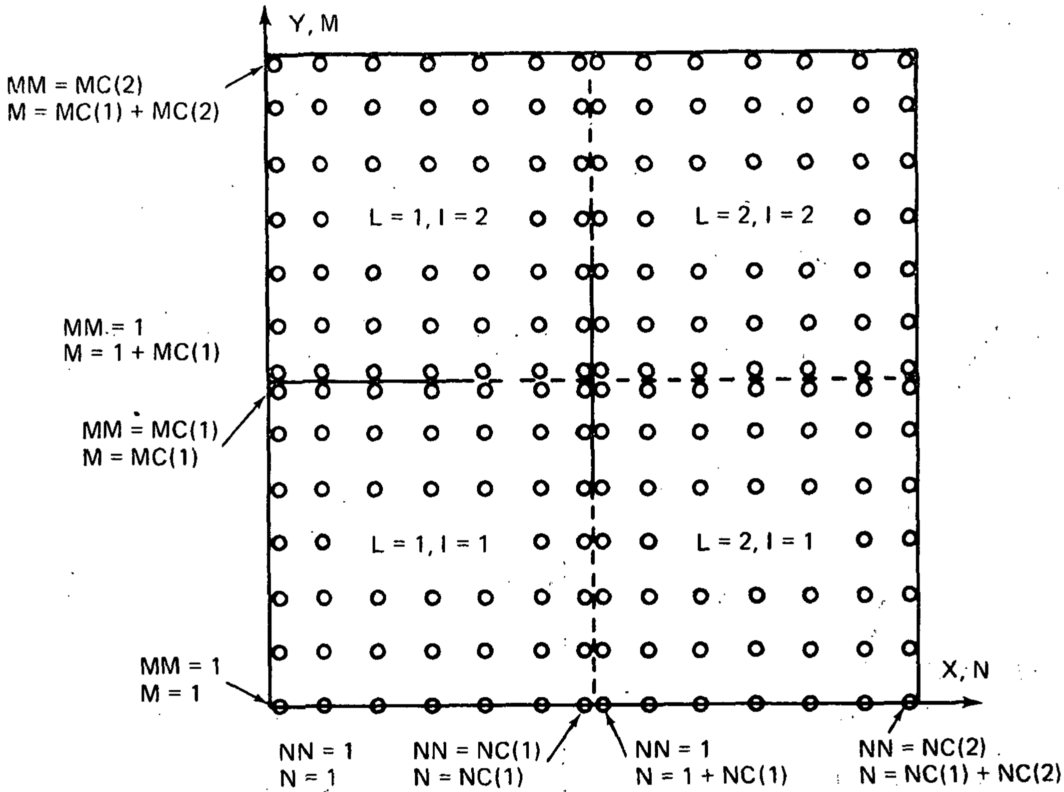


Figure 2 - COMPUTATIONAL GRID

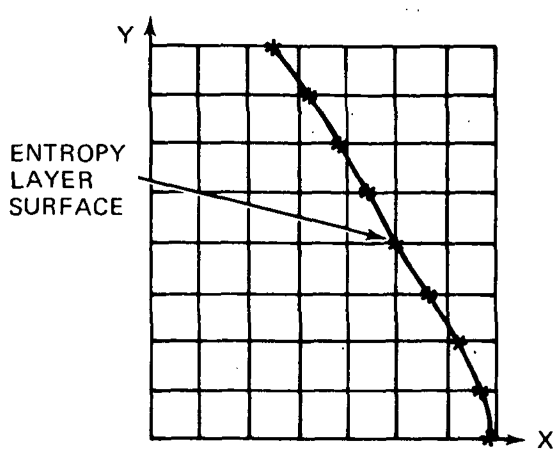


Figure 3 - ENTROPY LAYER SURFACE

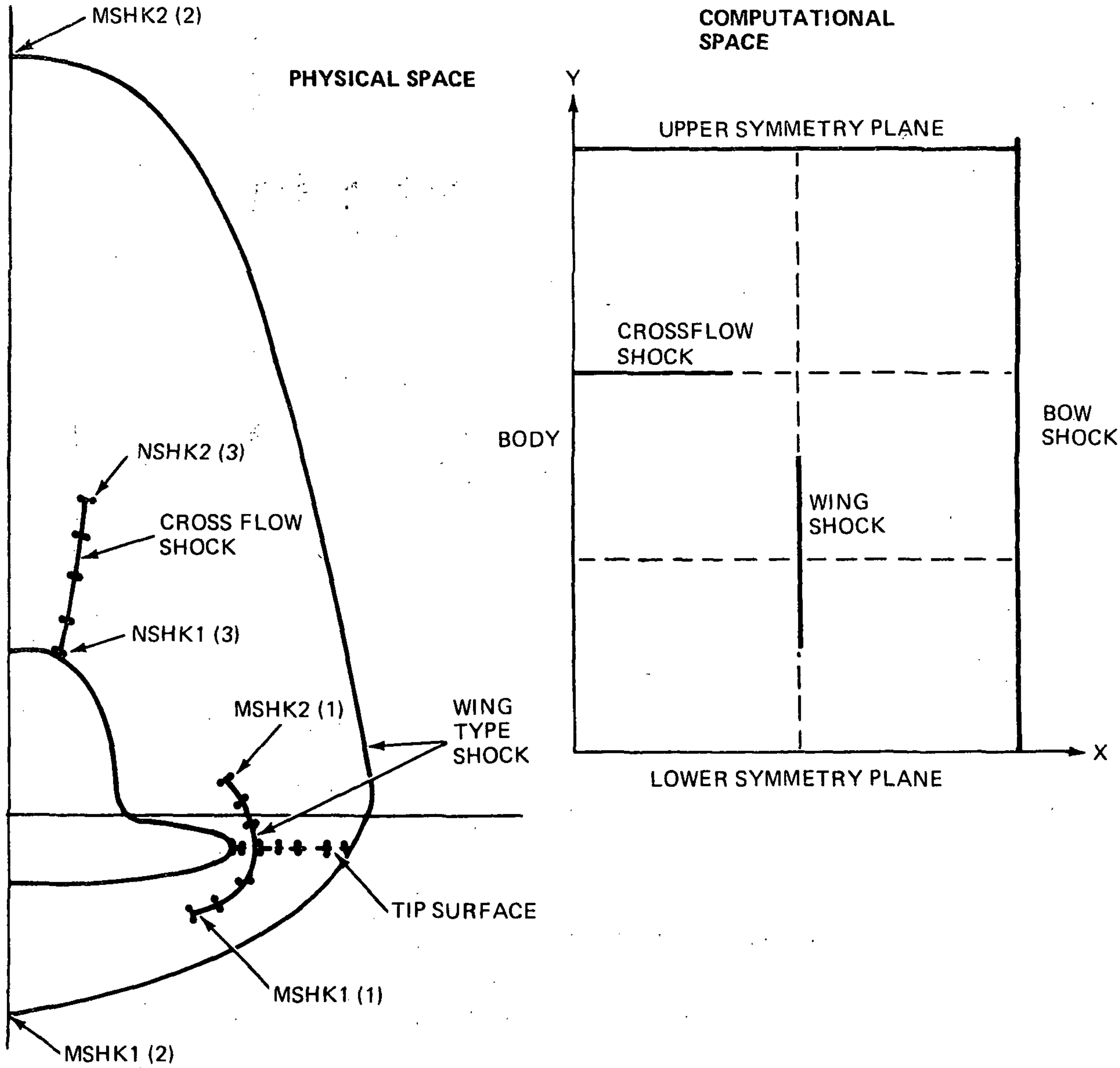


Figure 4 - CROSS SECTIONS IN THE PHYSICAL AND COMPUTATIONAL SPACES

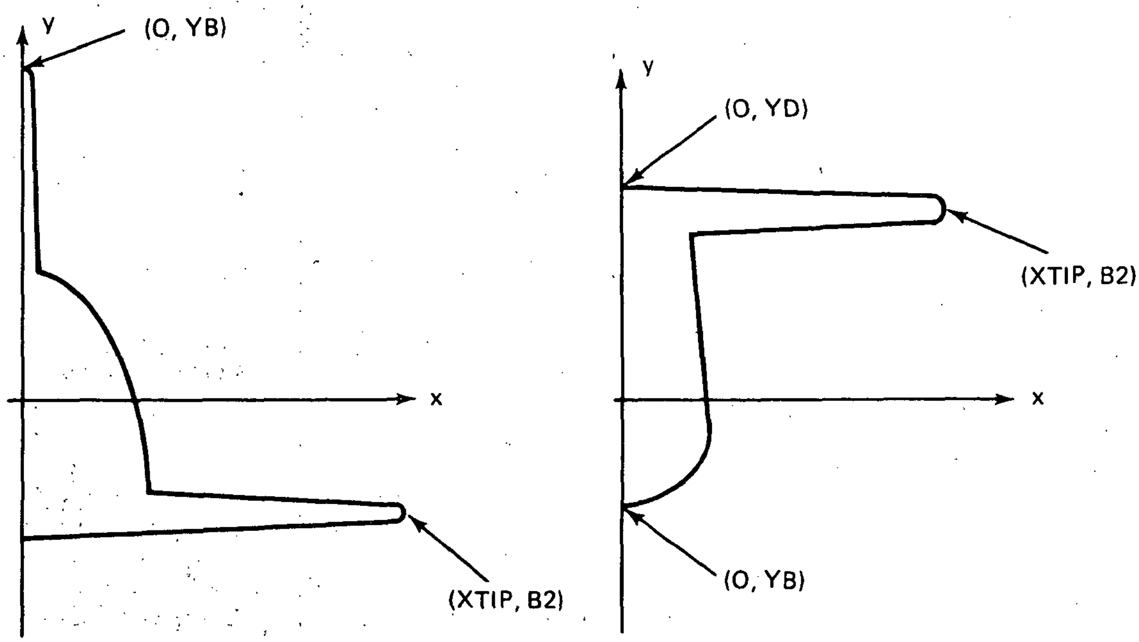


Figure 5 - MAPPING PARAMETERS

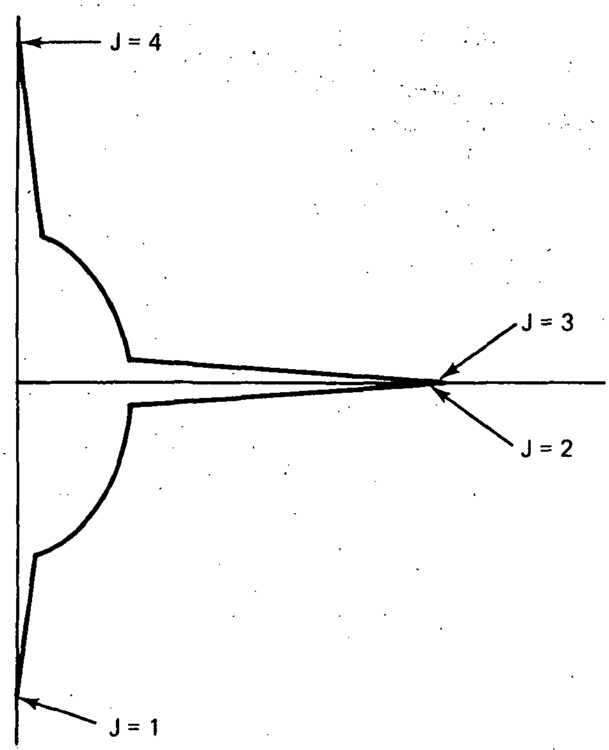


Figure 6 - SHARP LEADING EDGES

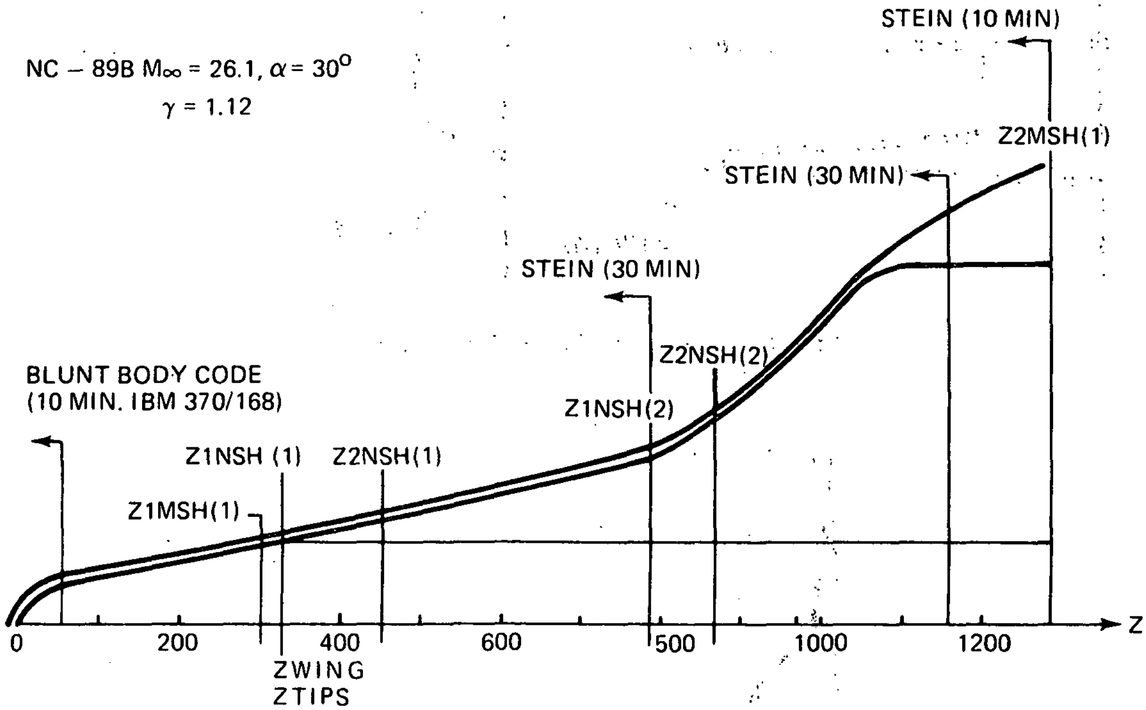


Figure 7 - SHUTTLE ORBITER TOP VIEW

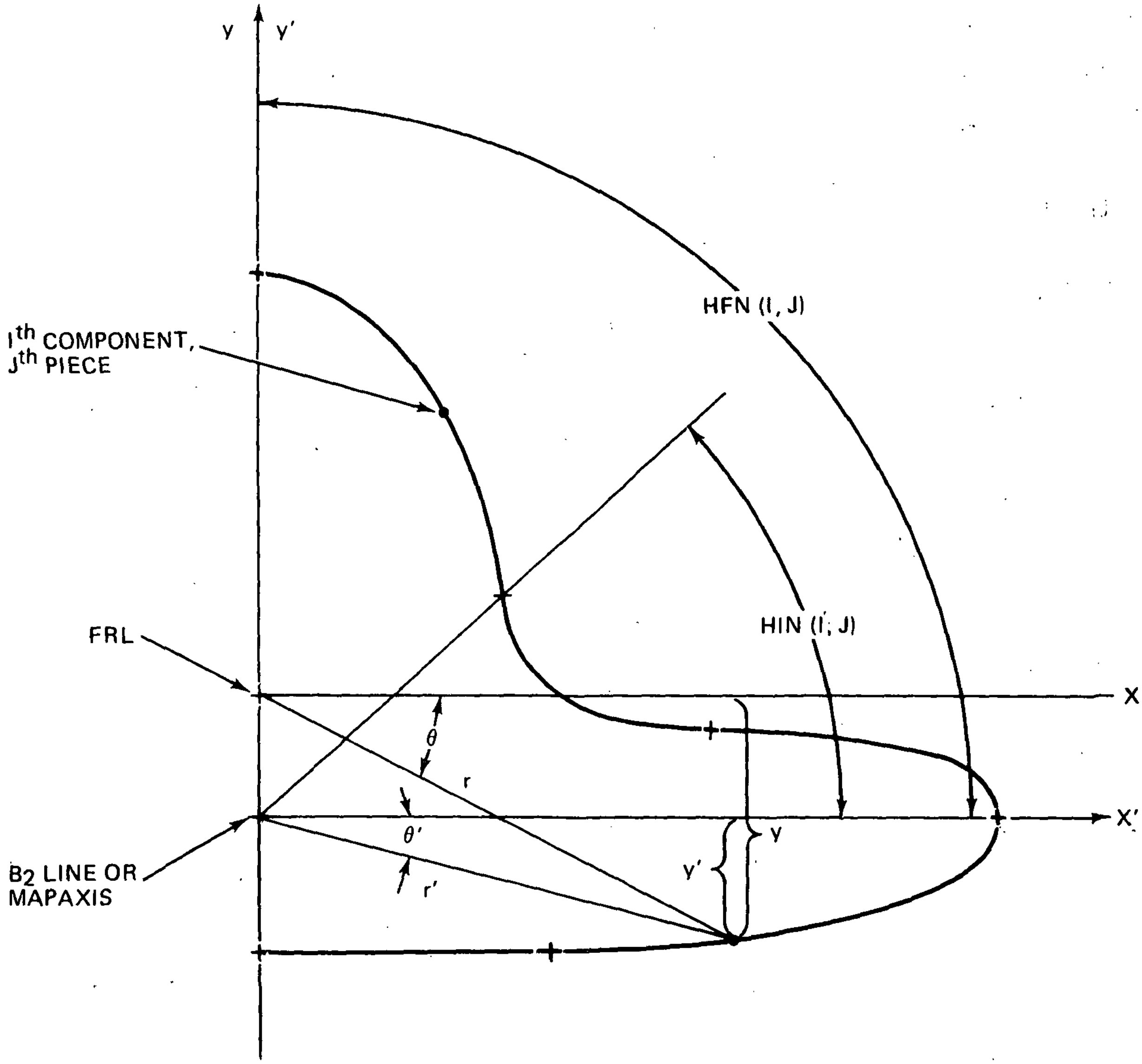


Figure 8 - AERODYNAMIC COEFFICIENT COMPONENT DEFINITION

INPUT DATA FORMAT FOR QUICK

QUICK input may be divided into three basic blocks: data input for (1) cross section modeling, (2) body line modeling, and (3) exercising the model. The first block may also be subdivided into (1a) - a cross section library definition, and (1b) - an application of this library to construct the total vehicle. For another presentation of QUICK input see Appendix A.

(1) - Cross Section Modeling

(a) - Library

Card Type	Format	Variable Names
1	15A4	VTITLE(I) (I = 1, 15)
2	I2	NCSM
3	2I2,6X,10A4	KDUM, KARC, CTITLE(I) (I = 1, 10)

(Note: There will be exactly NCSM cards of type 3 appearing together with the appropriate cards of type 4.)

Card Type	Format	Variable Names
4	A8,I2,A4,2X, A4,4X,A1,A8, 1X,A1,4A8	ARCNAM, JSEQ, ASHAPE, ATYPE, ASPEC(1), PNTNAM(1), ASPEC(2), PNTNAM(2), PNTNAM(3), ARCNM(1), ARCNM(2)

(Note: There will be exactly KARC cards of type 4 per model, and they will be grouped together for a given model after a card of type 3.)

(b) - Application (Note: These cards appear after NCSM blocks of one card 3 and KARC card 4's.)

Card Type	Format	Variable Names
5	I2,8X,A8	KNTCSM, ZMAPNM

Card Type	Format	Variable Names
6	2I2,6X, 2F10.5	KDUM, MODEL, XCSMS1(KDUM), XCSMS2(KDUM)

(Note: There will be exactly KNTCSM cards of type 6.)

(2) - Body Line Modeling

Card Type	Format	Variable Names
1	A1,A8,1X, A1,A8	BYORZ, BNAME, AYORZ, ANAME

(Note: There will be as many cards of type 1, followed by its cards of type 2 and 3, as there are body line models, and as many cards of type 1, alone, as there are aliased control point coordinates, plus one blank card to terminate modeling input.)

Card Type	Format	Variable Names
2	I2,1X,A4, 3X,A4,2X, A2, I1	KSEG, SSHAPE, STYPE, SDEF, IFREE

(Note: There will be as many cards of type 2 and 3 as there are segments in a given body line, plus one card type 2 with KSEG = -1. These cards are deleted when aliasing.)

Card Type	Format	Variable Names
3	3F10.5, 2(F9.4, A1), F10.5	D(1), D(2), D(3), D(4), SLP1, D(5), SLP2, D(6)

(Note: If SSHAPE is NULL, this card type 3 is deleted; also see Note for card type 2.)

(3) - Exercising the Model

Card Type	Format	Variable Names
1	I2, 1X, I2, 5X, 6F10.5	MODE, NDERV, XGO, XEND, XDEL, HGO, HEND, HDEL

(Note: MODE = 0, or blank, terminates all input.)

An example of the input deck for a simple sharp-nose cone (10^{o} half-angle) with afterbody follows in Fig. 9. Figure 11f also shows the intermediate data deck for this geometry.

INPUT DATA FORMAT FOR STEIN

There are five separate data sets read by the STEIN code. They are read on different read units because they may be generated in different places (i.e., some may be user-generated and others are generated by other codes). These data sets are shown in Fig. 10.

Control data (0) is read for every run of STEIN. This data set is generated by the user and read in on unit IREADO (set in a data statement in INIT). The data in control data (0) are

Card No.	Format	Variable Names
1	16I5	IREAD1, IREAD2, IREAD3, IREAD4, IWRIT, IPUNCH, ICASE, IBUG, MCIR, NRUN, KA, JA, NLOOK, NSOUT, IBLOUT, IAERO

Control Data (1) is read for every run of STEIN. This data set is generated by the user and read in on unit IREAD1. Its data are

Card No.	Format	Variable Names
2	5F10.5	ZEND, ZWRIT1, ZWRIT2, DZWRIT, DZFAC
3	6F10.5	ZGEOM1, ZGEOM2, DZGEOM, ZWING, ZTIPS, ZFREEZ
4	2(F10.5,I5)	ZNADD, NDEL, ZMADD, MDEL
5 & 5-a	8F10.5	ZSOUT(I) (I = 1, NSOUT) (if NSOUT $\leq$ 0 these cards are not read)

Control Data (2) is read for every run of STEIN. This data set is generated by the user for the first run of a configuration (geometry and free stream conditions). This data set is output (on IPUNCH) by STEIN for continuation runs of the same configuration but can be modified by the user. These data are read in on IREAD2 and consist of:

Card No.	Format	Variable Names
6	5E15.5	Z1NSH(I) (I = 1, 5)
7	5E15.5	Z2NSH(I) (I = 1, 5)
8	5E15.5	Z1MSH(I) (I = 1, 5)
9	5E15.5	Z2MSH(I) (I = 1, 5)
10	2E15.5	ZMAP1, ZMAP2
11	7I5	IENTE, IGAS, ISHTIP, ISHBEG(I) (I = 1, 3) IHS

The following data are read if and only if IAERO ≠ 0

Format	Variable Names
8X,I1,1X, 5(A2, 3X)	IAERD, CFTITL(I) (I = 1, 5)
5E15.6	VMO(2), VMO(3), APINF, ARINF, AREF
I2	KCOMP
I2,3X,A4	NP, CMPTTL(I) (Note: NP is stored in KPIECE(I))
I2,1X,I2, 2F10.4	INCP(I,J), IFCP(I,J), ZINIT(I,J) ZFINL(I,J) (I = 1, KCOMP; J = 1, NP = KPIECE(I))

The following data are read if and only if IAERD $\neq$ 0 (set and used by code for continuation runs).

Format	Variables Names
6E13.7	PFT(I,J,K), PMT(I,J,K), AR(I,J) (I = 1, KCOMP; J = 1, NP = KPIECE(I); K = 1, 3)

Starting plane control data are read for every run of STEIN. These data are generated by another code** or the user for the first run of a configuration. It is output from STEIN for continuation runs of the same configuration. These data are read on IREAD3 and consist of:

Card No.	Format	Variable Names
12	4I5	LC, IC, NC1, MC1
13 & 14	5E15.5	ZSTART, ACH, GAMIN, ATTACK, CONE, PIN, TIN
15	3E15.5	GAMFR, RQRI, SFR (Only read if IGAS = 2 i.e., the flow has been frozen in a previous run of STEIN.)

The starting plane flow field data are read by STEIN only if ICASE $\neq$ 1, since if ICASE = 1 the starting plane flow field data are computed in STEIN (vehicle having a sharp circular nose of half angle CONE with axis the same as the Z axis). This data set is generated by another code** or the user for the first STEIN run and is output by STEIN for continuation runs. These data are received on unit IREAD3 and consist of:

**These data are output by the BLUNT body code used to compute the flow over the nose of blunt vehicles.

Format	Variable Names
4I5	NC(L), MSHK1(L), MSHK2(L), NREG(L) (L = 1, LC)
4I5	MC(I), NSHK1(I), NSHK2(I), MREG(I) (I = 1, IC)
80I1	ISHOK (M, L) (L = 1, LC) (M = 1, MC(IC) + MREG(IC))
	MSHOK(N, I) (I = 1, IC + 1) (N = 1, NC(LC) + NREG(IC))
4E13.5	BN(M), CN(M,1), CHN(M,1) CZN(M,1) (M = 1, MC(IC) + MREG(IC))
3E13.5	CN(M,L), CHN(M,L), CZN(M,L) (L = 2, LC) (M = 1, MC(IC) + MREG(IC))
3E13.5	HSN(N,I), HSRN(N,I), HSZN(N,I) (I = 2, IC) (N = 1, NC(LC) + NREG(LC))
5E13.5	VN(N,M), UN(N,M), WN(N,M), PN(N,M), SN(N,M), (N = 1, NC(LC) + NREC(LC)) and (M = 1, MC(IC) + MREC(IC))

The following data is read if and only if IENTE = 2 (i.e., entropy layer points have been detected):

Format	Variable Names
80I1	IENT(M) (M = 1, MC (IC) + MREG (IC))
6E13.5	RHLN(M), PHLN(M), UHLN(M), VHLN(M), WHLN(M) SHLN(M) (M = 1, MC (IC) + MREG(IC))

The following data are read if and only if IHS > 0 (i.e., metric coefficient h_1 is being computed, and is not to be initialized by the code).

6E13.5	H1N(M) (M = 1, MC(IC) + MREG(IC))

The QUICK intermediate data set is read by STEIN for every run and is output by the QUICK code. These data are read on unit IREAD4. Since the user need not interact with these data, they will not be described in detail here.

INPUT DATA FORMAT FOR STRMBL

STRMBL input consists of user input control data, geometry data in the form of the QUICK intermediate data deck, and a flow field data tape generated by STEIN upon request. All control input is from unit IR, set in subroutine INOUT.

Card No.	Format	Variable Names
1	4F10.5	TESTM, TESTA, TESTG, TESTZ
2	E13.6	FNU
3	3I5	NS, JS, JCUT

Since the user need not alter the QUICK intermediate data deck, and the flow field data tape cannot be altered by the user, neither of these inputs need be described in detail. Geometry input is from unit IR; flow field data input is from unit ITP, also set in subroutine INOUT.

INPUT DATA FORMAT FOR BOOM

BOOM input consists of user input control data, geometry data in the form of the QUICK intermediate data deck, and a flow field data tape generated by STEIN upon request. All control inputs are from unit IR, set in subroutine INOUT.

Card No.	Format	Variable Name
1	4F10.5	TESTM, TESTA, TESTG, TESTZ
2	F10.4, 2I5	RCYL, NHPTS, JA

Since the user need not alter the QUICK intermediate data deck, and the flow field data tape cannot be altered by the user, neither of these inputs need be described in detail. Geometry input is from unit IR; flow field data input is from unit ITP, also set in subroutine INOUT.

```
 SCONE10: TEN DEGREE SHARP CONE
 1
 1 2
BDYLOWER 1ELLI  PIECE     BDYBOT     BDYSID     BDYLSCP
BDYUPPER 2ELLI  PIECE     BDYSID     BDYTOP     BDYUSCP
 1         MAPAXIS
 1 1       0.         20.
YBDYBOT
 1 LINE    PIECE KV5
0.         0.         20.        0.
-1
ZBDYBOT
 1 LINE    PIECE KV4
0.         0.         15.                   A-10.
 3 LINE    PIECE KV5
10.        -2.        20.        -2.
 2 ELLX    FILET KV0
1.         1.         3.         3.          10.        15.
-1
YBDYSID
 1 LINE    PIECE KV4
0.         0.         15.                   A10.
 3 LINE    PIECE KV5
10.        2.         20.        2.
 2 ELLX    FILET KV0
1.         1.         3.         3.          10.        15.
-1
ZBDYSID    YBDYBOT
YBDYTOP    YBDYBOT
ZBDYTOP    YBDYSID
YBDYLSCP   YBDYSID
ZBDYLSCP   ZBDYBOT
YBDYUSCP   YBDYSID
ZBDYUSCP   ZBDYTOP
YMAPAXIS   YBDYBOT
ZMAPAXIS   YBDYBOT

 1  2      5.         20.        5.
 2  2      5.         20.        5.          -90.       90.        10.
 3  2      5.         20.        5.
 4  2      5.         20.        5.          -90.       90.        30.
 5  1      5.         20.        5.          -90.       90.        30.
```

Figure 9 - SAMPLE INPUT DATA FOR QUICK

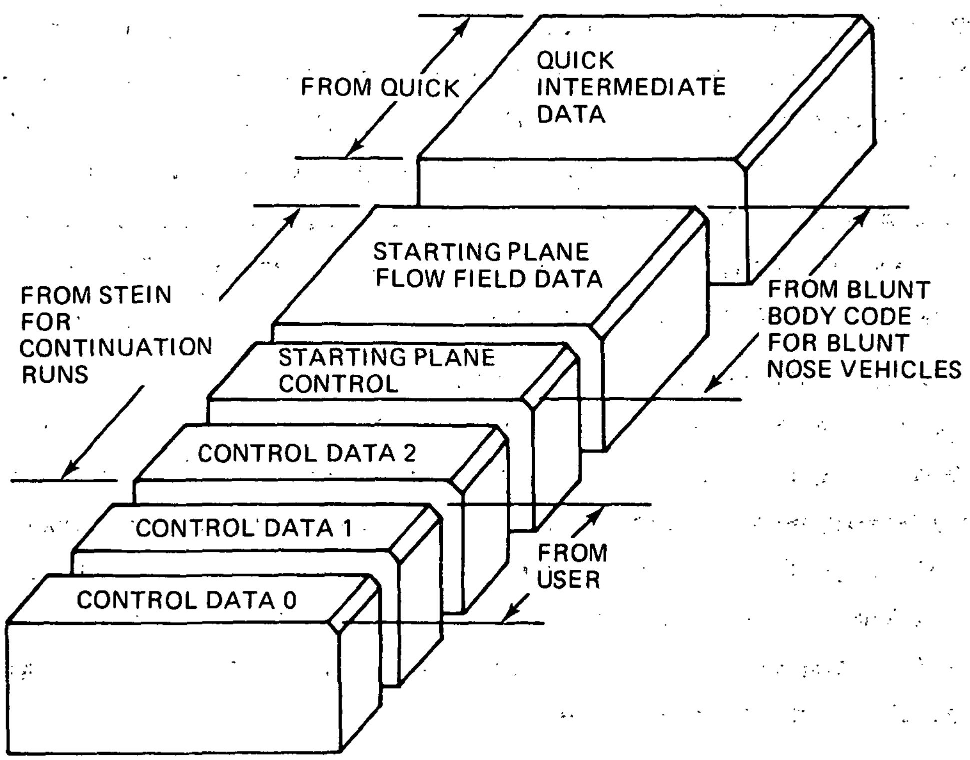


Figure 10 - STEIN INPUT

OUTPUT FORMAT FOR QUICK

QUICK generates several modes of printed output, output suitable for external plotting codes, and an intermediate data deck (the mathematical model) to be used as input to other codes using SUB-QUICK.

The math model is output on unit ITAPE (set in QUICK - the main routine) from subroutine GEMOUT. ITAPE may, of course, correspond to the punch unit in which case a card deck will be generated that may easily be used (with SUB-QUICK) with any other code. This data set need not be altered (configuration changes should be made in the initial QUICK input data which should then be rerun through QUICK, thus generating a new math model), and as such, will not be described in detail. This data deck is also included in the printed output and may be seen in Fig. 11f.

QUICK prints several cross section and body line checks with every run. Fig. 11a shows a correlation check between the cross section input data and the math model. Labels and names make this and all printed output self-explanatory. Note that the indices in parentheses correspond to the indices in the tables. Any misspelled names will show up as additional items in the component and/or control point tables and thus are easily detected on the first pass. A blank is always loaded into the first position of the control point index table.

Figure 11b shows a check list menu for body line models, output strictly for user convenience. In modeling a vehicle, the user may first define the logical cross section library and its application (see input data description) with subsequent blank cards to terminate input (thus, initially no body line models would be defined) and by filling in this table he could ensure that all control points were defined, either as a separate model or as an alias.

The output shown in Fig. 11c provides an important cross reference between the control point coordinates and the body line models (the indices in the parentheses) which define them. Model numbers are repeated because aliasing was used. The left hand sequential index bears a direct relation to the control point index table in Fig. 11a. Each control point has two coordinates which must be defined ($y = f(x)$, $z = g(x)$), and in Fig. 11c, the index for a particular control point's (n in Fig. 11a) z definition is $m_z = 2n$ and for its y definition, $m_y = 2n-1$. Any control point coordinates that were not defined will have a zero (0) in the parentheses, thus providing a quick check for complete definition. The first two blanks correspond to the initial blank in the control point index table of Fig. 11a.

The output shown in Figs. 11d and e provides a correlation check between the body line input data and the math model. The index in parentheses represents the shape of that segment, a negative value indicating that a line between the initial and final points of that segment has a negative slope. The output of Fig. 11e is completely annotated. In the column marked GAP, if two consecutive segments were not continuous in either x or v (v standing for y or z) the symbols X* or Y* would appear, accordingly. The last two lines in Fig. 11e are generated in GEMOUT, and indicate that a successful check was performed to ensure that all control points are defined throughout the range of the cross section models in which they are to be used.

Figure 11f shows a listing of the math model. Figure 12 gives an example of the output, generated at user request only, from MODEl. The first line is an echo of the user's input which requested this exercising of the math model (MODE, NDERV, etc. ... see input data description). This line appears at the start of each piece of user requested output. INXBLM is the body line model number, INXBLS is the segment number, and V represents y or z ($VX = dV/dx$, etc.). If MODE = - 1, no printed output will be generated, but the following output will be written on unit IPLOT:

Line	Variables	Format
1	IAM, IANDV	2I5
2	NXPTS, KNTBLM	2I5
3	XNOW	F10.4
4	V(I), VX(I), VXX(I) (I = 1, KNTBLM)	3F10.4

Blocks of lines 3 and 4 are repeated NXPTS times and line 4 is repeated KNTBLM times for each line 3.

Figure 13 shows an example of user requested output from MODE2. The use of a "G" suffix (THETAG, RAD-G, ZGCORD) denotes variables referenced to the "geometric" coordinate system; i.e., the x (not x′) axis which does not include the shifting due to the mapaxis (most often the FRL). Note that in general, ZGCORD = Z-CORD + ZCL(3). Here, since z of the mapaxis (ZCL(3)) is zero, ZGCORD = Z-CORD. Variables without the "G" are of course referenced to the mapaxis. H is used to represent θ′, so RH, RX, RXH, and RXX are the first and second derivatives of the radius R with respect to θ′ and x. All labels with "SUB" indicate derivatives formed numerically in SLOPE. Where "SUB" appears together with "D" the variables shown are the differences between the analytically formed and numerically formed derivatives. Plotting output from MODE2 (MODE = - 2) is in the following form:

Line	Variables	Format
1	IAMD, IANDV	2I5
2	NXPTS, NHPTS	2I5
3	XNOW	F10.4
4	YP, ZPG, HNOW*, RPX*, RPH*, RPXX**, RPXH**	7F10.4

*written if and only if IANDV ≥ 1

**written if and only if IANDV ≥ 2

Lines 1 and 2 are output once per call to MODE2; line 3 is output NXPTS times per call; line 4 is output NHPTS times for each line 3. Line 4 output consists of y, x, θ, r_x, $r_{\theta'}$, r_{xx}, and $r_{x\theta'}$.

Output from MODE3 is shown in Fig. 14. ZBCL, ZTCL, and ZMAP are ZCL(1), ZCL(2), and ZCL(3), respectively. J is an index reference for each arc, but it may not appear sequentially since the arcs will be listed in increasing θ′ after all intersections and fillets have been computed and inserted in their proper location. If J is positive the arc is in (IN(J) = 1); if J is negative the arc is not in (IN(J) = - 1) - this occurs, for example, when a growing piece is still completely contained by the basic skin or a fillet was unable to be inserted. U/THETA1 and U/THETA2 are the theta limits of the arc, UTHET1(J) and UTHET2(J) if J > 0, THETA1(J) and THETA2(J)(original definition theta limits - unaffected by intersections or fillets) if J < 0. RO, HO, AA, and BB are curve parameters R_o, θ'_o, A^2, and B^2. The second portion of MODE3 output is a cross-sectional interrogation in the neighborhood of each control point; labels are self-explanatory.

Plotting output for MODE = - 3 is generated in subroutine MODE1 (multiple body line traces may be used to create plan and profile views). Output format is the same as for MODE = - 1 except for line 4 which will consist of just V(I), I = 1, KNTBLM (no VX(I) or VXX(I)).

MODE4 output is shown in Fig. 15. Labels are the same as those used in the output of MODE2.

Output from MODE5 may be seen in Fig. 16. NORM-X, NORM-Y, and NORM-Z are the x, y, and z components of the unit normal to the body surface at the x, r′, θ′ location indicated.

There is also a mode of output for MODE = 6, but no separate subroutine is involved. When MODE = 6 is specified, GEMCHK exercises modes 1, 2, and 3 at x-stations near the limits of each cross section model. For plotting purposes, if MODE = - 6, GEMCHK exercises modes -2 and -7 at these same stations.

MODE7 output is for graphical purposes only. Output is again on unit IPLOT, and is in the form of cross-sectional cuts which show all arcs over their entire definition range (THETA1 to THETA2) rather than their limited use range (UTHET1 to UTHET2). For MODE = - 7, output is in the following format:

Line	Variables	Format
1	IAMD, IANDV	2I5
2	NXPTS, NHPTS	2I5
3	KARC, KNTARC	2I5
4	KNOW	F10.5
5	Y, ZG, HNOWR*, RX*, RH*, RXX**, RXH**	7F10.5

Lines 1 and 2 are written once per call to MODE7. Lines 3 and 4 are written NXPTS times per call. Line 5 is written KARC*NHPTS times for each write of lines 3 and 4. NHPTS is the number of points on each arc, KNTARC is the total number of arcs at the current station, and KARC is the number of arcs minus any fillets that were unable to be defined at this station (and also the number of arcs output from this mode for plotting purposes).

OUTPUT FORMAT FOR STEIN

STEIN generates three types of output. On unit IPUNCH STEIN will output (only if IPUNCH > 0) starting plane data to continue a run. This output is generated at Z = ZEND or at K = KA (i.e., the final axial station or step of a run).

*written if and only if IANDV ≥ 1
**written if and only if IANDV ≥ 2

The second type of output from STEIN is on unit IBLOUT (if and only if IBLOUT > 0) and is used as input for both BOOM and STRMBL. IBLOUT should usually correspond to a tape unit, since a great deal of output is to be expected. This output consists of body and shock position, the flow field variables, and the various region sizing and control parameters (IC, LC, MREG(I), etc.) at each computational step. The formats are not important as long as they are consistent with the input formats of STRMBL and BOOM, and since all the formats are consistent they need not be discussed further.

The last type of output from STEIN is usually printed on unit IWRIT. The input data is printed as shown in Fig. 17. The flow field data at the first axial station (Z = ZSTART) is always printed as in Fig. 16. Where X & Y are the Cartesian coordinates of the mesh point, P is the pressure (p/p_∞) U, V & W are the three Cartesian velocity components, S is the entropy, M is the total Mach number and MA is the axial component of the Mach number. This flow field data will be printed in this format at every axial station between ZWRIT1 and ZWRIT2 at an interval of DZWRIT; the maximum number of steps between outputs is JA. Figure 18 shows a "Geometry Test" of the body in the mapped space. Here Y is the circumferential position in the computational space, B is the body radius in the mapped space, BH and BZ are the body derivatives with respect to the polar angle and axial position in the mapped space and finally XX and YY are the Cartesian coordinates in the physical space. Figure 19 shows the output format for the variables on the entropy layer surface.

Aerodynamic coefficients are also written on unit IWRIT following the flow field output at each z-station. An example of the aero-coefficient output follows in Fig. 20a and b. The first piece of output, 20a, is computed using a reference area which is the integrated surface area of a given component up to the current station. The second piece of output, 20b is computed with a user input reference area. Labels make the output self-explanatory but it is important to note that the input reference area must be in the same units as the geometry is model.

OUTPUT FORMAT FOR STRMBL

Output from STRMBL is of two main types. The first of these is associated with the tracing of streamlines on the body, and consists of the location (θ′, θ, r, x, and y) of each streamline and the value of the flow variables (u, v, w, p, and S) at these locations in various data planes. Also included are the index, the integrated arc length, and the value of the metric coefficient h_1 for each streamline at the current z-station, see Fig. 21a.

The second type of output from STRMBL corresponds to the development of the pseudo-stream-surfaces. Locations and values of flow variables and their derivatives are output at NNPT points along the body normals originating from each of the previously traced streamlines at selected data planes. For each data plane (which, along with the θ′ location of each streamline and the geometry model, establishes the origin points for the body normals) there are two blocks of output associated with each streamline. The first block gives the location of and flow variable values at the points equally distributed along the body normal. The second block gives the length along the normal, the derivatives of the flow quantities in the normal direction (DUDN = du/dn, etc.) and the component of velocity in the normal direction (VELDTN = $\vec{Q} \cdot \hat{n}$ or $\vec{Q} \cdot \hat{\zeta}$) at the same points; see Fig. 21b.

OUTPUT FORMAT FOR BOOM

Output from BOOM, see Fig. 22, is a simple presentation of flow variables (p, S, u, v, w) on the surface of the data cylinder of user specified raidus with centerline at x = y = 0 (the z axis, not the z′ axis). HC is the angle θ to the points on the cylinder, measured from the windward symmetry plane.

```
SCONE10: TEN DEGREE SHARP CONE

  CHECK CROSS SECTION DEFINITION
 1
 1 2
BDYLOWER( 1)  1 ELLI( 3)PIEC( 1)0BDYBOT   ( 2)BDYSID   ( 3)BDYLSCP  ( 4)
BDYUPPER( 2)  2 ELLI( 3)PIEC( 1)0BDYSID   ( 3)BDYTOP   ( 5)BDYUSCP  ( 6)

 1          MAPAXIS   ( 7)
 1 1  BDYBOT   ( 2)  BDYTOP    ( 5)              0.0        20.00000

  COMPONENT INDEX TABLE
  1BDYLOWER
  2BDYUPPER

  CONTROL POINT INDEX TABLE
  1
  2BDYBOT
  3BDYSID
  4BDYLSCP
  5BDYTOP
  6BDYUSCP
  7MAPAXIS
```

Figure 11a - CROSS SECTION MODEL CHECK

CHECK LIST MENU FOR BODY LINE MODELS

CONTROL POINT NAMES	PLANFORM DEFINITION Y-COORDINATE	PROFILE DEFINITION Z-COORDINATE
1		
2 BDYBOT		
3 BDYSID		
4 BDYLSCP		
5 BDYTOP		
6 BDYUSCP		
7 MAPAXIS		

Figure 11b - CHECK LIST MENU FOR BODY LINE MODELS

SCONE10: TEN DEGREE SHARP CONE

CHECK BODY LINE DEFINITION

BODY LINE COORDINATE INDEX

1	Y		(0)
2	Z		(0)
3	Y	BDYBOT	(1)
4	Z	BDYBOT	(2)
5	Y	BDYSID	(3)
6	Z	BDYSID	(1)
7	Y	BDYLSCP	(3)
8	Z	BDYLSCP	(2)
9	Y	BDYTOP	(1)
10	Z	BDYTOP	(3)
11	Y	BDYUSCP	(3)
12	Z	BDYUSCP	(3)
13	Y	MAPAXIS	(1)
14	Z	MAPAXIS	(1)

BODY LINE MODEL TABLES

BODY LINE MODEL NUMBER	1	1				
1	LINE (1)	0.0	0.0	0.0	-0.20000000E 02	0.0
BODY LINE MODEL NUMBER	2	3				
1	LINE (-1)	0.0	0.0	-0.26449032E 01	-0.15000000E 02	0.0
2	ELLX (-3)	10.00000	-1.76327	-0.48419382E-01	-0.27460033E 00	0.48419386E-02
3	LINE (1)	10.00000	-2.00000	0.0	-0.10000000E 02	0.0
BODY LINE MODEL NUMBER	3	3				
1	LINE (1)	0.0	0.0	0.26449032E 01	-0.15000000E 02	0.0
2	ELLX (3)	10.00000	1.76327	-0.48419382E-01	0.27460033E 00	0.48419386E-02
3	LINE (1)	10.00000	2.00000	0.0	-0.10000000E 02	0.0

Figure 11d - BODY LINE MODEL CHECK TABLE 1

***** YBDYBOT   BODY LINE MODEL   MODEL NUMBER = 1   NUMBER OF SEGMENTS = 1

BOUNDARY CONDITIONS

SEG	SHAPE	CONN	DEF	FREE	GAP	X-ORIGIN	Y-ORIGIN	X-TERM	Y-TERM	X-LEFT SCP	Y-LEFT SCP	X-RIGHT SCP	Y-RIGHT SCP
1	LINE	PIEC	KV	5		0.0	0.0	20.00000	0.0	10.00000	0.0	10.00000	0.0

SEGMENT EQUATIONS

SEG	SHAPE	EQUATION	A-COEFFICIENT	B-COEFFICIENT	C-COEFFICIENT
1	LINE	0.=AX+BY	0.0	-0.20000000E 02	0.0

ALIAS LIST

ZBDYSID   YBDYTOP   YMAPAXIS   ZMAPAXIS

***** ZBDYBOT   BODY LINE MODEL   MODEL NUMBER = 2   NUMBER OF SEGMENTS = 3

BOUNDARY CONDITIONS

SEG	SHAPE	CONN	DEF	FREE	GAP	X-ORIGIN	Y-ORIGIN	X-TERM	Y-TERM	X-LEFT SCP	Y-LEFT SCP	X-RIGHT SCP	Y-RIGHT SCP
1	LINE	PIEC	KV	4		0.0	0.0	10.00000	-1.76327	7.50000	-1.32245	7.50000	-1.32245
2	ELLX	FILE	KV	0		10.00000	-1.76327	15.00000	-2.00000	12.50000	-2.20409	12.50000	-2.00000
3	LINE	PIEC	KV	5		15.00000	-2.00000	20.00000	-2.00000	15.00000	-2.00000	15.00000	-2.00000

SEGMENT EQUATIONS

SEG	SHAPE	EQUATION	A-COEFFICIENT	B-COEFFICIENT	C-COEFFICIENT
1	LINE	0.=AX+BY	-0.26449032E 01	-0.15000000E 02	0.0
2	ELLX	0.=AX+BY+CXX+YY	-0.48419382E-01	-0.27460033E 00	0.48419386E-02
3	LINE	0.=AX+BY	0.0	-0.10000000E 02	0.0

ALIAS LIST

ZBDYLSCP

***** YBDYSID   BODY LINE MODEL   MODEL NUMBER = 3   NUMBER OF SEGMENTS = 3

BOUNDARY CONDITIONS

SEG	SHAPE	CONN	DEF	FREE	GAP	X-ORIGIN	Y-ORIGIN	X-TERM	Y-TERM	X-LEFT SCP	Y-LEFT SCP	X-RIGHT SCP	Y-RIGHT SCP
1	LINE	PIEC	KV	4		0.0	0.0	10.00000	1.76327	7.50000	1.32245	7.50000	1.32245
2	ELLX	FILE	KV	0		10.00000	1.76327	15.00000	2.00000	12.50000	2.20409	12.50000	2.00000
3	LINE	PIEC	KV	5		15.00000	2.00000	20.00000	2.00000	15.00000	2.00000	15.00000	2.00000

SEGMENT EQUATIONS

SEG	SHAPE	EQUATION	A-COEFFICIENT	B-COEFFICIENT	C-COEFFICIENT
1	LINE	0.=AX+BY	0.26449032E 01	-0.15000000E 02	0.0
2	ELLY	0.=AX+BY+CXX+YY	-0.48419382E-01	0.27460033E 00	0.48419386E-02
3	LINE	0.=AX+BY	0.0	-0.10000000E 02	0.0

ALIAS LIST

YBDYLSCP   ZBDYTOP   YBDYUSCP   ZBDYUSCP

CROSS SECTION CHECK AGAINST BODY LINES
CROSS SECTION DEFINITION CHECK IS FINISHED

Figure 11e - BODY LINE MODEL CHECK TABLE 2

```
 SCONE10: TEN DEGREE SHARP CONE
 1  1  7
 2  2  5
 1  1       1     3     1     0     2     3     4    -1    -1    -1    -1
 1  2       2     3     1     0     3     5     6    -1    -1    -1    -1
 1  1     0.0       20.00000
14
 1  0
 2  0
 3  1
 4  2
 5  3
 6  1
 7  3
 8  2
 9  1
10  3
11  3
12  3
13  1
14  1
 3
 1  1         0.0             20.00000
 1  1     0.0        0.0        20.00000   0.0
 1  1  1   0.0            -0.20000000E 02 0.0
 2  3         0.0             20.00000
 2  1     0.0        0.0        10.00000  -1.76327
 2  1 -1  -0.26449032E 01-0.15000000E 02 0.0
 2  2    10.00000  -1.76327   15.00000  -2.00000
 2  2 -3  -0.48419379E-01-0.27460033E 00 0.48419349E-02
 2  3    10.00000  -2.00000   20.00000  -2.00000
 2  3  1   0.0            -0.10000000E 02 0.0
 3  3         0.0             20.00000
 3  1     0.0        0.0        10.00000   1.76327
 3  1  1   0.26449032E 01-0.15000000E 02 0.0
 3  2    10.00000   1.76327   15.00000   2.00000
 3  2  3  -0.48419379E-01 0.27460033E 00 0.48419349E-02
 3  3    10.00000   2.00000   20.00000   2.00000
 3  3  1   0.0            -0.10000000E 02 0.0
```

Figure 11f - SAMPLE QUICK INTERMEDIATE DATA DECK (MATH MODEL)

1	2	5.00000	20.00000	5.00000	0.0	0.0	0.0

XSTATION = 5.00000

INXBLM	INXBLS	V	VX	VXX
1	1	0.0	0.0	0.0
2	1	-0.88163	-0.17633	0.0
3	1	0.88163	0.17633	0.0

XSTATION = 10.00000

INXBLM	INXBLS	V	VX	VXX
1	1	0.0	0.0	0.0
2	1	-1.76327	-0.17633	0.0
3	1	1.76327	0.17633	0.0

XSTATION = 15.00000

INXBLM	INXBLS	V	VX	VXX
1	1	0.0	0.0	0.0
2	2	-2.00000	-0.00000	0.01295
3	2	2.00000	0.00000	-0.01295

XSTATION = 20.00000

INXBLM	INXBLS	V	VX	VXX
1	1	0.0	0.0	0.0
2	3	-2.00000	0.0	0.0
3	3	2.00000	0.0	0.0

Figure 12 - QUICK OUTPUT FOR MODE = 1

2 2 5.00000 20.00000 5.00000 -90.00000 90.00000 10.00000

SCONE10: TEN DEGREE SHARP CONE

GEOMETRY CHECK

STATION = 5.00000

ZMAP = 0.0 ZXMAP = 0.0 ZXXMAP = 0.0

THETA	RADIUS	THETAG	RAD-G	Y-CORD	Z-CORD	ZGCORD
-0.900000E 02	0.881634E 00	-0.900000E 02	0.881634E 00	0.0	-0.881634E 00	-0.881634E 00
-0.800000E 02	0.881634E 00	-0.799999E 02	0.881634E 00	0.153094E 00	-0.868240E 00	-0.868240E 00
-0.700000E 02	0.881634E 00	-0.699999E 02	0.881634E 00	0.301537E 00	-0.828465E 00	-0.828465E 00
-0.600000E 02	0.881634E 00	-0.599999E 02	0.881634E 00	0.440817E 00	-0.763517E 00	-0.763517E 00
-0.500000E 02	0.881634E 00	-0.500000E 02	0.881634E 00	0.566704E 00	-0.675371E 00	-0.675371E 00
-0.400000E 02	0.881634E 00	-0.400000E 02	0.881634E 00	0.675371E 00	-0.566703E 00	-0.566703E 00
-0.300000E 02	0.881634E 00	-0.300000E 02	0.881634E 00	0.763518E 00	-0.440817E 00	-0.440817E 00
-0.200000E 02	0.881634E 00	-0.200000E 02	0.881634E 00	0.828465E 00	-0.301537E 00	-0.301537E 00
-0.100000E 02	0.881634E 00	-0.999999E 01	0.881634E 00	0.868240E 00	-0.153094E 00	-0.153094E 00
0.0	0.881634E 00	0.0	0.881634E 00	0.881634E 00	0.0	0.0
0.100000E 02	0.881634E 00	0.999999E 01	0.881634E 00	0.868240E 00	0.153094E 00	0.153094E 00
0.200000E 02	0.881634E 00	0.200000E 02	0.881634E 00	0.828465E 00	0.301537E 00	0.301537E 00
0.300000E 02	0.881634E 00	0.300000E 02	0.881634E 00	0.763518E 00	0.440817E 00	0.440817E 00
0.400000E 02	0.881634E 00	0.400000E 02	0.881634E 00	0.675371E 00	0.566703E 00	0.566703E 00
0.500000E 02	0.881634E 00	0.500000E 02	0.881634E 00	0.566704E 00	0.675371E 00	0.675371E 00
0.600000E 02	0.881634E 00	0.599999E 02	0.881634E 00	0.440817E 00	0.763517E 00	0.763517E 00
0.700000E 02	0.881634E 00	0.699999E 02	0.881634E 00	0.301537E 00	0.828465E 00	0.828465E 00
0.800000E 02	0.881634E 00	0.799999E 02	0.881634E 00	0.153094E 00	0.868240E 00	0.868240E 00
0.900000E 02	0.881634E 00	0.900000E 02	0.881634E 00	0.0	0.881634E 00	0.881634E 00

DERIVATIVES CHECK

STATION = 5.00000

ZMAP = 0.0 ZXMAP = 0.0 ZXXMAP = 0.0

THETA	RADIUS	RH	RSUBH	DSUBH	RX	RXH	RXSUBH	DXSUBH	RXX
-89.99997	0.88163	0.0	0.102453E-05	-0.10E-05	0.176327E 00	0.0	-0.683019E-06	0.68E-06	0.152213E-06
-79.99997	0.88163	0.766837E-07	0.341510E-06	-0.26E-06	0.176327E 00	-0.271810E-07	0.0	-0.27E-07	0.228320E-06
-69.99997	0.88163	0.766837E-07	-0.170755E-06	0.25E-06	0.176327E 00	0.0	0.512264E-06	-0.51E-06	0.125032E-06
-59.99997	0.88163	0.0	-0.170755E-06	0.17E-06	0.176327E 00	0.0	0.0	0.0	0.103288E-06
-49.99998	0.88163	0.0	0.0	0.0	0.176327E 00	0.0	-0.341510E-06	0.34E-06	0.233756E-06
-39.99998	0.88163	-0.766837E-07	0.0	-0.77E-07	0.176327E 00	-0.153367E-07	0.0	-0.15E-07	0.239192E-06
-29.99998	0.88163	0.0	0.0	0.0	0.176327E 00	0.0	0.170755E-06	-0.17E-06	0.206575E-06
-19.99998	0.88163	0.0	0.170755E-06	-0.17E-06	0.176327E 00	0.0	0.0	0.0	0.125032E-06
-10.00000	0.88163	-0.766837E-07	-0.170755E-06	0.94E-07	0.176327E 00	0.118442E-07	-0.170755E-06	0.18E-06	0.924152E-07
0.0	0.88163	0.0	0.0	0.0	0.176327E 00	0.0	0.0	0.0	0.152213E-06
10.00000	0.88163	0.766837E-07	0.170755E-06	-0.94E-07	0.176327E 00	-0.118442E-07	0.170755E-06	-0.18E-06	0.157650E-06
19.99998	0.88163	0.0	-0.170755E-06	0.17E-06	0.176327E 00	0.0	0.0	0.0	0.190267E-06
29.99998	0.88163	0.0	0.0	0.0	0.176327E 00	0.0	0.0	0.0	0.190267E-06
39.99998	0.88163	0.766837E-07	0.0	0.77E-07	0.176327E 00	0.153367E-07	0.341509E-06	-0.33E-06	0.190267E-06
49.99998	0.88163	0.0	0.0	0.0	0.176327E 00	0.0	0.170755E-06	-0.17E-06	0.761067E-07
59.99997	0.88163	0.0	0.170755E-06	-0.17E-06	0.176327E 00	0.0	-0.170755E-06	0.17E-06	0.869791E-07
69.99997	0.88163	-0.766837E-07	0.170755E-06	-0.25E-06	0.176327E 00	0.0	-0.341510E-06	0.34E-06	0.163086E-06
79.99997	0.88163	-0.766837E-07	-0.341510E-06	0.26E-06	0.176327E 00	0.271810E-07	0.0	0.27E-07	0.179394E-06
89.99997	0.88163	0.0	-0.102453E-05	0.10E-05	0.176327E 00	0.0	0.170755E-06	-0.17E-06	0.190267E-06

Figure 13 - QUICK OUTPUT FOR MODE = 2

3 2 5.00000 20.00000 5.00000 0.0 0.0 0.0

XSTATION = 5.00000

ZBCL	ZBCLX	ZBCLXX	ZTCL	ZTCLX	ZTCLXX	ZMAP	ZMAPX	ZMAPXX
-0.88163	-0.17633	0.0	0.88163	0.17633	0.0	0.0	0.0	0.0

J	U/THETA1	U/THETA2	RO	ROX	ROXX	HO	HOX	HOXX	AA	AAX	AAXX	BB	BBX	BBXX
1	-1.5708	0.0	0.0	0.0	0.0	-.16E 01	0.0	0.0	0.78E 00	0.31E 00	0.62E-01	0.78E 00	0.31E 00	0.62E-01
2	0.0	1.5708	0.0	0.0	0.0	0.16E 01	0.0	0.0	0.78E 00	0.31E 00	0.62E-01	0.78E 00	0.31E 00	0.62E-01

XSTATION = 5.00000

THETA	RADIUS	RX	RH	RXX	RXH	Y-CORD	Z-CORD	ZGCORD
-1.57080	0.88163	0.17633	0.0	0.00000	0.0	0.0	-0.88163	-0.88163
-1.57079	0.88163	0.17633	0.0	0.00000	0.0	0.00001	-0.88163	-0.88163
-0.00001	0.88163	0.17633	0.0	0.00000	0.0	0.88163	-0.00001	-0.00001
0.0	0.88163	0.17633	0.0	0.00000	0.0	0.88163	0.0	0.0
0.00001	0.88163	0.17633	0.0	0.00000	0.0	0.88163	0.00001	0.00001
1.57079	0.88163	0.17633	0.0	0.00000	0.0	0.00001	0.88163	0.88163
1.57080	0.88163	0.17633	0.0	0.00000	0.0	0.0	0.88163	0.88163

Figure 14 - QUICK OUTPUT FOR MODE = 3

4 2 5.00000 20.00000 5.00000 -90.00000 90.00000 30.00000

SCONE10: TEN DEGREE SHARP CONE

GEOMETRY CHECK

THETA = -90.00000

X-CORD	RADIUS	RAD-G	Y-CORD	Z-CORD	ZGCORD	THETAG	ZMAP
0.500000E 01	0.881634E 00	0.881634E 00	0.0	-0.881634E 00	-0.881634E 00	-0.900000E 02	0.0
0.100000E 02	0.176327E 01	0.176327E 01	0.0	-0.176327E 01	-0.176327E 01	-0.900000E 02	0.0
0.150000E 02	0.200000E 01	0.200000E 01	0.0	-0.200000E 01	-0.200000E 01	-0.900000E 02	0.0
0.200000E 02	0.200000E 01	0.200000E 01	0.0	-0.200000E 01	-0.200000E 01	-0.900000E 02	0.0

Figure 15 - QUICK OUTPUT FOR MODE = 4

SCONE10: TEN DEGREE SHARP CONE

DERIVATIVES CHECK

THETA = -90.00000

X-CORD	RADIUS	RX	RSUBX	DSUBX	RH	RXH	RXSUBH	DXSUBH	ZXMAP
5.00000	0.88163	0.176327E 00	0.230966E 00	-0.55E-01	0.0	0.0	0.0	0.0	0.0
10.00000	1.76327	0.176328E 00	0.932421E-01	0.83E-01	0.0	0.0	0.0	0.0	0.0
15.00000	2.00000	0.398393E-07	0.236734E-01	-0.24E-01	0.0	0.0	0.0	0.0	0.0
20.00000	2.00000	0.0	-0.236735E-01	0.24E-01	0.0	0.0	0.0	0.0	0.0

SCONE10: TEN DEGREE SHARP CONE

DERIVATIVES CHECK

THETA = -90.00000

X-CORD	RADIUS	RX	RXX	RXSUBX	DXSUBX	RSUBXX	DSUBXX	DXX	ZXXMAP
5.00000	0.88163	0.176327E 00	0.152213E-06	0.105801E-01	-0.11E-01	-0.335697E-01	0.34E-01	-0.44E-01	0.0
10.00000	1.76327	0.176328E 00	-0.565365E-06	-0.176327E-01	0.18E-01	-0.194722E-01	0.19E-01	-0.18E-02	0.0
15.00000	2.00000	0.398393E-07	-0.129452E-01	-0.105797E-01	-0.24E-02	-0.114440E-01	-0.15E-02	-0.86E-03	0.0
20.00000	2.00000	0.0	0.0	0.176328E-01	-0.18E-01	-0.686174E-02	0.69E-02	-0.24E-01	0.0

Figure 15 (continued)

5 1 5.00000 20.00000 5.00000 -90.00000 90.00000 30.00000

SURFACE NORMALS CHECK

STATION = 5.00000

THETA	RADIUS	Y-CORD	Z-CORD	ZGCORD	NORM-X	NORM-Y	NORM-Z
-0.900000E 02	0.881634E 00	0.276814E-06	-0.881634E 00	-0.881634E 00	-0.173648E 00	0.0	-0.984808E 00
-0.600000E 02	0.881634E 00	0.440817E 00	-0.763517E 00	-0.763517E 00	-0.173648E 00	0.492404E 00	-0.852868E 00
-0.300000E 02	0.881634E 00	0.763518E 00	-0.440817E 00	-0.440817E 00	-0.173648E 00	0.852868E 00	-0.492404E 00
0.0	0.881634E 00	0.0	0.0	0.0	-0.173648E 00	0.984808E 00	0.0
0.300000E 02	0.881634E 00	0.763518E 00	0.440817E 00	0.440817E 00	-0.173648E 00	0.852868E 00	0.492404E 00
0.600000E 02	0.881634E 00	0.440817E 00	0.763517E 00	0.763517E 00	-0.173648E 00	0.492404E 00	0.852868E 00
0.900000E 02	0.881634E 00	0.276814E-06	0.881634E 00	0.881634E 00	-0.173648E 00	0.0	0.984808E 00

Figure 16 - QUICK OUTPUT FOR MODE = 5

```
THREE-DIMENSIONAL SUPERSONIC FLOW,PROGRAM 148 /                                   RUN NUMBER  1
ZSTART=  50.00000  ZEND= 790.00000
KA= 1000  JA=  100  ICASE=  2  IBUG=  1  IENTE=  1  IGAS=  0
ACH=  26.09999  ATTACK=  30.00000  GAMMA=  1.12000  PIN=  0.10000E 01  TIN=  0.10000E 01
ZWING=  0.36000E 03  ZTIPS=  0.36000E 03  ZFREE=  0.12800E 04  ZNADD=  0.60000E 03  NDFL=  4  ZMADD=  0.35000E 03  MDEL=  10
ZINSH=  0.36000E 03  0.79000E 03  0.12800E 04  0.12800E 04  0.12800E 04
Z2NSH=  0.46000E 03  0.84000E 03  0.12800E 04  0.12800E 04  0.12800E 04
Z1MSH=  0.31000E 03  0.12800E 04  0.12800E 04  0.12800E 04  0.12800E 04
Z2MSH=  0.12800E 04  0.12800E 04  0.12800E 04  0.12800E 04  0.12800E 04
THERE WERE NO SPECIAL OUTPUT STATIONS RLQUESTFD

                    GEOMETRY TEST
  ZGEOM1=     0.50000E 02  ZGEOM2=    -0.10000E 01  DZGEOM=    0.10000E 01
  CONE=   9.99999  ZMAP1=     0.50000E 02  ZMAP2= 100.00000
 QUICK GEOMETRY FOR SHUTTLE ORBITER
```

INITIAL DATA

```
STEP    0 Z=  0.5000E 02DZ=  0.0        N=    0 M=    0

 0.0      0.0      0.0      0.0      0.0      0.0      0.0
SHOCK L=  1  MSHK1=    1  MSHK2=   22
SHOCK I=  1  NSHK1=    0  NSHK2=    0
SHOCK I=  2  NSHK1=    0  NSHK2=    0
REGION L=  1  NC=   11  NREG=    0
REGION I=  1  MC=   22  MREG=    0
```

NN= 1 (THE BODY)

	X	Y	P	U	V	W	S	M	MA
1	0.00002	-49.81158	166.50075	0.0	-0.30138	10.90100	3.38960	1.72520	1.72454
2	7.50792	-49.81158	170.52805	0.56185	-0.36261	10.79100	3.38960	1.70822	1.70495
3	15.21516	-49.32620	183.20393	1.21350	-0.53098	10.46000	3.38960	1.66946	1.64632
4	22.96836	-47.69427	196.76292	2.04690	-0.59396	10.06600	3.38960	1.61325	1.57826
5	30.57936	-44.65166	192.92462	3.02520	-0.16488	9.90740	3.38960	1.62611	1.55503
6	37.82416	-40.76476	182.14438	4.02220	0.62314	9.80660	3.38960	1.67195	1.54427
7	44.44624	-35.44469	158.99921	4.70930	1.80280	9.89680	3.38960	1.76153	1.56954
8	50.16840	-26.96478	128.16819	4.97600	3.27390	10.17500	3.38960	1.89153	1.63239
9	54.71463	-21.47394	96.55374	4.77710	4.88970	10.62100	3.38960	2.05737	1.73001
10	57.83978	-13.20156	69.07549	3.94930	6.16840	11.29800	3.38960	2.23287	1.87360
11	59.36259	-4.44665	46.56734	2.68430	7.10910	12.17600	3.38960	2.43099	2.06231
12	59.37671	4.44965	32.51495	1.42240	7.61980	13.02300	3.38960	2.61680	2.24863
13	58.07111	13.25429	25.74910	0.58844	7.83000	13.61700	3.38960	2.74822	2.38076
14	55.37816	21.73428	21.68887	0.03434	7.87630	14.11500	3.38960	2.85215	2.49062
15	51.38741	29.66846	18.30730	-0.49772	7.67400	14.56400	3.38960	2.93260	2.59329
16	46.22726	36.86496	15.71475	-0.84636	7.30710	14.95100	3.38960	2.99134	2.68406
17	40.05321	43.16699	13.82528	-0.97638	6.90040	15.28300	3.38960	3.03623	2.76256
18	33.03564	48.45430	12.54472	-1.05450	6.58070	15.59400	3.38960	3.08143	2.83349
19	25.35007	52.63989	11.70012	-1.00470	6.33750	15.86700	3.38960	3.12156	2.89388
20	17.17046	55.66507	10.96490	-0.82131	6.12600	16.08800	3.38960	3.15423	2.94441
21	8.66573	57.49313	10.23077	-0.48179	5.96490	16.25200	3.38960	3.18145	2.98549
22	0.00007	58.10455	9.59363	0.0	5.94930	16.36499	3.38960	3.20977	3.01662

BOTTOM SYMMETRY PLANE (row 1) — *TOP* (row 22)

NN= 2 (NEXT RING OUT TO THE BOW SHOCK)

	X	Y	P	U	V	W	S	M	MA
1	0.00002	-50.46562	169.68121	0.0	-0.38499	11.37100	3.35930	1.82247	1.82143
2	7.60195	-50.43541	173.86653	0.73330	-0.43134	11.28400	3.35850	1.81102	1.80589
3	15.39378	-49.90523	186.43620	1.55420	-0.55460	11.02200	3.35620	1.77879	1.75918
4	23.23566	-48.24930	199.61651	2.51880	-0.54554	10.71300	3.35280	1.75479	1.70612
5	30.92877	-45.36418	195.87941	3.52800	-0.04649	10.59700	3.34920	1.78349	1.69216
6	38.26649	-41.24147	165.06355	4.50280	0.78404	10.53200	3.34490	1.84244	1.69015
7	45.00838	-35.89297	162.47105	5.15790	1.96650	10.59500	3.34400	1.93138	1.71284
8	50.69017	-29.38150	132.43604	5.40030	3.39900	10.79000	3.34660	2.04650	1.76153
9	55.64162	-21.63775	101.28111	5.20230	4.94060	11.11000	3.35110	2.18594	1.83633
10	58.96587	-13.46315	73.95615	4.46330	6.18130	11.68500	3.35070	2.34572	1.96452
11	60.75407	-4.55293	51.13657	3.33860	7.11850	12.45100	3.34860	2.52754	2.13710

Figure 17 - INITIAL PLANE OUTPUT

```
THREE-DIMENSIONAL SUPERSONIC FLOW.PROGRAM 148 /                    RUN NUMBER 1
ZSTART=  47.00000  ZEND=  66.70000
KA= 1000  JA=  100  ICASE=  2  IBUG=  1  ILNTE=  0  IGAS=  0
ACH=  6.00000  ATTACK=  0.0  GAMMA=  1.20000  PIN=  0.10000E 01  TIN=  0.10000E 01
ZWING=  0.31200E 02  ZTIPS=  0.31200E 02  ZFREF=  0.66700F 02  ZNADD=  0.10000E 02  NDEL=  15  ZMADD=  0.20000E 02  MDEL=  15
ZINSH=  0.10000E 07  0.10000E 07  0.47400E 02  0.70000E 02  0.70000E 02
Z2NSH= -0.10000E 07 -0.10000E 07  0.70000E 02  0.70000E 02  0.70000E 02
ZIMSH=  0.70000F 02  0.70000E 02  0.70000E 02  0.70000E 02  0.70000E 02
Z2MSH=  0.70000E 02  0.70000E 02  0.70000E 02  0.70000E 02  0.70000E 02
THERE WERE NO SPECIAL OUTPUT STATIONS REQUESTED
```

GEOMETRY TEST

```
ZGEOM1=  0.0   ZGEOM2=  0.66700E 02  DZGEOM=  0.10000E 01
CONE=  15.19999  ZMAP1=  0.32000E 00  ZMAP2=  10.00000
HSRA   QUICK GEOMETRY FOR MODEL HSRA CONFIGURATION
```

Z= 0.0

	Y	B	BH	BZ	XX	YY
1	0.0	0.00018	0.0	0.27169	0.00000	-0.00018
2	0.04762	0.00018	-0.00000	0.27169	0.00002	-0.00018
3	0.09524	0.00018	-0.00000	0.27169	0.00003	-0.00018
4	0.14286	0.00018	-0.00000	0.27169	0.00005	-0.00017
5	0.19048	0.00018	-0.00000	0.27169	0.00006	-0.00017
6	0.23810	0.00018	-0.00000	0.27169	0.00008	-0.00016
7	0.28571	0.00018	-0.00000	0.27169	0.00009	-0.00016
8	0.33333	0.00018	-0.00000	0.27169	0.00010	-0.00015
9	0.38095	0.00018	-0.00000	0.27169	0.00012	-0.00014
10	0.42857	0.00018	0.00000	0.27169	0.00013	-0.00013
11	0.47619	0.00018	0.00000	0.27169	0.00014	-0.00012
12	0.52381	0.00018	0.00000	0.27169	0.00015	-0.00010
13	0.57143	0.00018	0.0	0.27169	0.00016	-0.00009
14	0.61905	0.00018	0.00000	0.27169	0.00016	-0.00008
15	0.66667	0.00018	0.00000	0.27169	0.00017	-0.00006
16	0.71429	0.00018	0.00000	0.27169	0.00018	-0.00005
17	0.76190	0.00018	0.00000	0.27169	0.00018	-0.00003
18	0.80952	0.00018	0.00000	0.27169	0.00018	-0.00001
19	0.85714	0.00018	-0.00000	0.27169	0.00018	0.00000
20	0.90476	0.00018	-0.00000	0.27169	0.00018	0.00002
21	0.95238	0.00018	0.0	0.27169	0.00018	0.00003
22	1.00000	0.00018	-0.00000	0.27169	0.00017	0.00005
1	0.0	0.00018	-0.00000	0.27169	0.00017	0.00005
2	0.08333	0.00018	-0.00000	0.27169	0.00017	0.00007
3	0.16667	0.00018	-0.00000	0.27169	0.00016	0.00009
4	0.25000	0.00018	-0.00000	0.27169	0.00015	0.00010
5	0.33333	0.00018	-0.00000	0.27169	0.00014	0.00012
6	0.41667	0.00018	0.00000	0.27169	0.00012	0.00013
7	0.50000	0.00018	0.00000	0.27169	0.00011	0.00014
8	0.58333	0.00018	0.0	0.27169	0.00009	0.00016
9	0.66667	0.00018	0.00000	0.27169	0.00008	0.00016
10	0.75000	0.00018	0.00000	0.27169	0.00006	0.00017
11	0.83333	0.00018	0.00000	0.27169	0.00004	0.00018
12	0.91667	0.00018	0.00000	0.27169	0.00002	0.00018
13	1.00000	0.00018	0.0	0.27169	0.00000	0.00018

WING TIP SURFACE (points to rows 22 and 1 of Z= 0.0)

Z= 1.000

	Y	B	BH	BZ	XX	YY
1	0.0	0.19964	0.0	0.06187	0.00000	-0.20000
2	0.04762	0.19968	0.00102	0.06247	0.01745	-0.19927
3	0.09524	0.19982	0.00204	0.06427	0.03478	-0.19708
4	0.14286	0.20004	0.00305	0.06721	0.05190	-0.19344
5	0.19048	0.20035	0.00406	0.07125	0.06868	-0.18838
6	0.23810	0.20075	0.00506	0.07630	0.08502	-0.18192
7	0.28571	0.20124	0.00605	0.08226	0.10081	-0.17410
8	0.33333	0.20182	0.00703	0.08900	0.11595	-0.16495
9	0.38095	0.20248	0.00800	0.09640	0.13033	-0.15453
10	0.42857	0.20322	0.00895	0.10432	0.14386	-0.14289
11	0.47619	0.20405	0.00988	0.11260	0.15642	-0.13011
12	0.52381	0.20496	0.01078	0.12111	0.16794	-0.11624
13	0.57143	0.20594	0.01165	0.12969	0.17832	-0.10139
14	0.61905	0.20701	0.01249	0.13820	0.18748	-0.08562
15	0.66667	0.20814	0.01329	0.14650	0.19534	-0.06905
16	0.71429	0.20934	0.01404	0.15447	0.20181	-0.05177
17	0.76190	0.21061	0.01472	0.16198	0.20685	-0.03388
18	0.80952	0.21193	0.01534	0.16895	0.21038	-0.01552
19	0.85714	0.21330	0.01587	0.17527	0.21235	0.00320
20	0.90476	0.21473	0.01768	0.18061	0.21274	0.02214
21	0.95238	0.21653	0.02316	0.18512	0.21182	0.04124
22	1.00000	0.21880	0.02858	0.18996	0.20966	0.06046
1	0.0	0.21880	0.02856	0.18996	0.20966	0.06046
2	0.08333	0.22224	0.03503	0.19637	0.20525	0.08416
3	0.16667	0.22635	0.04105	0.20329	0.19877	0.10789
4	0.25000	0.23107	0.04636	0.21068	0.19007	0.13150
5	0.33333	0.23632	0.05059	0.21846	0.17897	0.15476
6	0.41667	0.24194	0.05328	0.22649	0.16531	0.17736
7	0.50000	0.24774	0.05394	0.23457	0.14896	0.19887
8	0.58333	0.25349	0.05203	0.24240	0.12984	0.21878
9	0.66667	0.25887	0.04714	0.24965	0.10799	0.23646
10	0.75000	0.26355	0.03909	0.25590	0.08360	0.25121
11	0.83333	0.26719	0.02806	0.26074	0.05707	0.26236
12	0.91667	0.26952	0.01468	0.26381	0.02895	0.26932
13	1.00000	0.27032	0.0	0.26487	0.00000	0.27169

Z= 2.000

	Y	B	BH	BZ	XX	YY
1	0.0	0.26553	0.0	0.07252	0.00000	-0.26250
2	0.04762	0.26601	0.01093	0.07301	0.02193	-0.26186
3	0.09524	0.26745	0.02170	0.07449	0.04395	-0.25992
4	0.14286	0.26982	0.03219	0.07691	0.06616	-0.25663
5	0.19046	0.27310	0.04227	0.08021	0.08862	-0.25191
6	0.23810	0.27724	0.05183	0.08432	0.11135	-0.24565

Figure 18 - GEOMETRY TEST OUTPUT

2	79.33952	448.63208	4.72362	0.70695	14.11835	21.85757	1.02882	14.29816	12.00609
3	41.05798	467.10400	4.55149	0.36147	14.23296	21.88606	1.02292	14.40807	12.07743
4	0.00057	475.33911	4.50313	0.0	14.32294	21.92487	0.99756	14.62465	12.24360

NN= 11 (BOW SHOCK)

	X	Y	P	U	V	W	S	M	MA	
1	0.00005	-80.74327	213.06577	0.0	-0.27951	23.19080	2.23424	6.06484	6.06440	
2	21.56699	-80.49380	212.82309	0.39258	-0.26604	23.19316	2.23319	6.06948	6.06821	
3	43.13039	-79.58731	213.93681	0.85797	-0.28065	23.15503	2.23797	6.04824	6.04365	
4	64.35666	-77.94235	214.22591	1.52602	-0.23146	23.11455	2.23920	6.04275	6.02932	
5	84.64880	-75.10149	209.46503	2.69020	0.10711	23.11810	2.21866	6.13432	6.09314	
6	103.07898	-70.36879	193.92238	3.94398	0.96417	23.27303	2.14850	6.45133	6.35534	
7	119.10098	-64.45012	202.91034	5.30011	1.15734	22.78577	2.18967	6.26449	6.09415	
8	132.58316	-57.84842	200.65240	6.25977	1.70725	22.55429	2.18039	6.30639	6.06059	
9	144.28645	-51.14046	180.48079	6.43929	4.14587	21.99799	2.08358	6.75092	6.20763	
10	151.52809	-41.58492	140.14218	9.46943	7.91225	21.50980	1.85890	7.64168	6.80185	
11	156.15782	-31.43024	105.10097	6.67434	9.51181	22.05115	1.61205	9.15500	7.90634	WING TIP SURFACE
1	156.15782	-31.43016	105.10097	8.67434	9.51181	22.05115	1.61205	9.15500	7.90634	
2	161.77187	-21.11966	82.02573	7.85327	10.37369	22.51817	1.40844	10.34845	8.96026	
3	166.64258	-7.46992	53.12871	6.61281	12.04101	22.76358	1.07677	12.55670	10.75074	
4	169.66884	6.63100	37.65988	5.63917	13.34877	22.65520	0.84145	14.37049	12.10587	
5	169.49632	26.15759	30.09792	4.97108	13.99172	22.60011	0.70332	15.55446	12.99973	
6	168.33221	44.91669	24.82166	4.50161	13.99840	22.81683	0.59490	16.56812	13.91822	
7	167.82368	65.15973	21.40724	4.19900	13.74998	23.10300	0.51842	17.31187	14.69830	
8	166.78944	87.34184	19.32893	3.98762	13.55445	23.30276	0.46915	17.82043	15.23828	
9	170.57242	111.66858	17.83224	3.81505	13.39802	23.45566	0.43229	18.21388	15.66018	
10	174.01126	139.46260	16.23734	3.61498	13.30534	23.57582	0.39170	18.66141	16.10889	
11	178.44298	171.64221	14.45413	3.38822	13.37151	23.61218	0.34467	19.20085	16.57907	
12	182.51793	209.75119	12.57969	3.13460	13.51763	23.60617	0.29334	19.81895	17.08568	
13	185.46001	254.96970	11.15231	2.91800	13.76394	23.52364	0.25301	20.33020	17.44749	
14	183.53244	305.69233	9.71319	2.63344	14.12461	23.37556	0.21138	20.88675	17.79413	
15	172.99237	358.28711	8.54089	2.27966	14.52652	23.19254	0.17691	21.37503	18.05252	
16	151.92928	407.04883	7.62833	1.86347	14.87296	23.03067	0.14989	21.77988	18.25420	
17	121.16148	449.56714	6.57712	1.38516	15.06923	22.96129	0.11883	22.27684	18.60054	CROSS FLOW SHOCK
1	121.16148	449.56714	6.57712	1.38518	15.06923	22.96129	0.11883	22.27684	18.60054	
2	85.63585	481.53003	5.70026	0.91852	15.16127	22.94481	0.09327	22.72012	18.94514	
3	44.37018	502.05176	5.13784	0.47957	15.19975	22.94609	0.07724	23.02026	19.18872	
4	0.00061	511.20776	5.05600	0.0	15.24089	22.92577	0.07495	23.06511	19.20792	

LNTROPY LAYER SURFACE DATA

	IENT	X	Y	P	U	V	W	S	M	MA	
1	2	0.00005	-63.68585	215.36320	0.00000	-0.65837	23.81189	2.08821	6.64501	6.64247	
2	2	20.49835	-63.68571	214.23940	0.59591	-0.65738	23.77602	2.09589	6.61620	6.61160	
3	2	41.23151	-63.68576	215.03462	1.24421	-0.65494	23.68781	2.11091	6.55327	6.54175	
4	2	62.02864	-63.68584	214.95773	2.01714	-0.65106	23.54749	2.13183	6.46870	6.44266	
5	2	82.16475	-63.28842	204.34387	2.92222	-0.43031	23.39738	2.15198	6.41199	6.36150	
6	2	100.55553	-61.05701	209.23164	3.72278	-0.14934	23.22018	2.16446	6.35051	6.27031	
7	2	116.36403	-57.22383	214.11888	4.42653	0.07900	23.05029	2.17214	6.30871	6.19547	
8	2	129.31517	-52.10840	207.60854	5.33235	0.47271	22.85452	2.17550	6.31006	6.14377	
9	2	139.44995	-45.71214	183.81647	6.42090	1.45284	22.59378	2.17627	6.36545	6.11132	
10	2	146.11684	-38.04332	126.43452	6.67176	3.19969	22.53745	2.17670	6.54489	6.21833	
11	2	148.20976	-30.38263	60.61606	5.09676	4.70925	23.04443	2.17709	6.90578	6.61248	
1	2	148.20975	-30.38258	60.61606	5.09676	4.70925	23.04443	2.17709	6.90578	6.61248	
2	2	146.02238	-24.24310	22.90605	1.44571	5.13264	23.88921	2.17779	7.39712	7.21946	
3	2	138.04704	-17.84221	2.17630	-0.42204	3.03251	25.09500	2.18298	8.64638	8.58274	
4	2	121.61044	-12.62506	2.38770	-0.80813	1.33578	25.16127	2.19996	8.51487	8.49853	
5	2	108.78307	-0.34836	3.23441	-0.17538	3.15622	24.85277	2.22233	8.24273	8.17685	
6	2	107.99992	18.56909	4.48628	0.00000	4.62078	24.44789	2.24396	7.96655	7.82796	
7	2	108.00000	37.02184	4.37018	0.00000	5.76259	24.16156	2.26122	7.90328	7.68766	
8	2	108.00009	55.20229	5.09376	-0.00000	6.79354	23.79066	2.27530	7.75897	7.46075	
9	2	106.90633	72.90488	3.66230	-1.15320	7.81828	23.52036	2.29142	7.86318	7.45368	
10	2	103.08160	89.32474	1.56690	-2.66114	8.27735	23.47543	2.31475	8.21085	7.69972	
11	2	96.92870	104.30865	1.13606	-4.16798	8.24604	23.30995	2.34007	8.27859	7.69605	
12	2	86.86649	117.66864	0.69640	-5.68354	8.00095	23.22232	2.34730	8.51729	7.84544	
13	2	79.33260	129.24648	0.36654	-6.94141	7.30577	23.42648	2.30495	9.08722	8.34760	
14	2	68.78067	138.95255	0.16551	-7.65617	6.14862	23.98468	2.20174	10.09131	9.33894	
15	2	57.65984	146.78294	0.14120	-7.94973	4.88008	24.43568	2.08644	10.81380	10.10272	
16	2	46.38301	152.62306	0.11619	-8.12339	3.75590	24.69318	2.04197	11.19310	10.52326	
17	2	35.28726	157.23668	0.15963	-7.88487	2.65016	24.60596	2.17550	10.25249	9.71251	CROSS FLOW SHOCK
1	2	35.28726	157.23668	2.14961	-0.41697	0.14015	24.36610	2.49334	7.26156	7.26038	
2	0	79.33952	448.63208	4.72362	0.70695	14.11835	21.85757	1.02882	14.29816	12.00609	
3	0	41.05798	467.10400	4.55149	0.36147	14.23296	21.88606	1.02292	14.40807	12.07743	
4	0	0.00057	475.33911	4.50313	0.00000	14.32294	21.92487	0.99756	14.62465	12.24360	

Figure 19 - BOW SHOCK AND ENTROPY LAYER SURFACE OUTPUT

AERODYNAMIC COEFFICIENTS

USING

```
PINF  =     1.0000
RHOIN = 0.10000E 01
VIN   =     9.4066
QIN   =    44.2417
```

MOMENTS ARE TAKEN ABOUT A LINE THROUGH

```
YO =     0.0
ZO =    10.0000
```

CONE PARAMETERS

FOR PIECE(S)	IN Z-RANGE	BETWEEN CONT. PTS.
1	1.0000 , 20.0000	2 , 5

```
CL =  0.0098

CD =  0.0365

CM = -0.0002

CN =  0.0110

CA =  0.0362

AREA =          398.941 SQ. UNITS
```

TOTL PARAMETERS

```
CL =  0.0098

CD =  0.0365

CM = -0.0002

CN =  0.0110

CA =  0.0362

AREA =          398.941 SQ. UNITS
```

Figure 20a - AERODYNAMIC COEFFICIENTS OUTPUT 1

```
AERODYNAMIC  COEFFICIENTS


USING

PINF   =    10.0000
RHOIN  = 0.10000E-06
VIN    =94065.6250
QIN    =   442.4167
AND AREA(REF) =            12.566 SQ. UNITS


MOMENTS  ARE  TAKEN  ABOUT  A  LINE  THROUGH

YO =     0.0
ZO =    10.0000


CONE PARAMETERS

    FOR PIECE(S)                   IN Z-RANGE                      BETWEEN CONT. PTS.
         1                    1.0000 ,        20.0000                    2  ,   5

CL =  0.3104

CD =  1.1598

CM = -0.0355

CN =  0.3507

CA =  1.1483


TOTL PARAMETERS

CL =  0.3104

CD =  1.1598

CM = -0.0355

CN =  0.3507

CA =  1.1483
```

Figure 20b - AERODYNAMIC COEFFICIENTS OUTPUT 2

STREAMLINE AND PSEUDO-STREAM SURFACE
CALCULATIONS FOR BOUNDARY LAYER INPUT

FREE STREAM MACH NO. = 15.0000
ANGLE OF ATTACK = 5.0000
GAMMA = 1.2000
STARTING AT Z = 50.0000

STREAMLINE DATA (ON THE BODY) ...

AT STATION Z # 320.5479
WITH NEW STEP SIZE DZ = 3.7304

INDEX	THETAP	THETA	R	X	Y	LENGTH	U	V	W	P	S	HI
1	-1.5708	-1.5708	57.2919	0.0001	-57.2919	272.0122	0.0000	-0.3194	11.5511	5.5060	2.6774	47.7406
2	-1.4935	-1.5046	57.4179	3.8008	-57.2919	272.0210	0.0107	-0.3191	11.5411	5.5575	2.6774	46.4543
3	-1.4179	-1.4396	57.7884	7.5587	-57.2919	272.0466	0.0200	-0.3189	11.5342	5.5931	2.6774	46.8249
4	-1.3412	-1.3733	58.4276	11.4640	-57.2919	272.0991	0.0284	-0.3188	11.5302	5.6140	2.6774	48.8464
5	-1.2692	-1.3106	59.2881	15.2552	-57.2919	272.1897	0.0283	-0.3188	11.5311	5.6095	2.6774	50.8368
6	-1.1878	-1.2386	60.6044	19.7617	-57.2919	272.3445	0.0307	-0.3189	11.5342	5.5933	2.6774	55.6113
7	-1.0986	-1.1566	62.5283	25.0485	-57.2919	272.5933	0.0449	-0.3190	11.5378	5.5742	2.6774	65.6125
8	-1.0157	-1.0828	64.8648	30.4151	-57.2919	272.9436	0.0441	-0.3196	11.5602	5.4631	2.6774	79.8609
9	-0.9123	-0.9858	68.7192	37.9469	-57.2919	273.5190	0.1059	-0.3194	11.5519	5.5005	2.6774	102.7747
10	-0.8256	-0.9023	73.0088	45.2540	-57.2919	274.2915	0.3014	-0.3168	11.4591	5.9683	2.6774	104.5546
11	-0.7105	-0.7880	80.8133	56.9950	-57.2919	275.7317	0.2801	-0.2546	11.5060	5.7359	2.6774	195.8138
12	-0.5227	-0.5947	99.3927	82.3276	-55.6873	275.1797	0.6030	-0.8251	11.4215	5.9959	2.6774	505.4995
13	-0.3235	-0.3971	106.3815	98.1054	-41.1383	275.5969	1.8402	0.0338	11.1872	6.7020	2.6774	432.1084
14	-0.2213	-0.2979	105.1215	100.4899	-30.8594	275.8284	1.7279	0.5532	11.2292	6.4847	2.6774	316.5696
15	-0.1316	-0.2112	102.7889	100.5047	-21.5491	275.6011	1.6940	0.5795	11.1767	6.8238	2.6774	202.2810
16	-0.0558	-0.1371	101.4565	100.5047	-13.8645	275.2688	1.6933	0.6542	11.1721	6.8254	2.6774	137.9717
17	0.0260	-0.0560	100.6625	100.5047	-5.6327	275.0295	1.6964	0.7659	11.1927	6.6611	2.6774	98.2827
18	0.1004	0.0187	100.5222	100.5047	1.8750	274.8149	1.6962	0.7942	11.1913	6.6578	2.6774	75.0562
19	0.1727	0.0921	100.9327	100.5047	9.2846	274.6719	1.6937	0.7946	11.1749	6.7565	2.6774	64.1091
20	0.2501	0.1717	102.0040	100.5047	17.4244	274.6609	1.6936	0.8287	11.1740	6.7468	2.6774	61.9957
21	0.3271	0.2517	103.7747	100.5047	25.8453	274.7473	1.6930	0.8726	11.1704	6.7489	2.6774	64.6987
22	0.3950	0.3231	105.9877	100.5047	33.6480	274.9158	1.6918	0.9067	11.1625	6.7808	2.6774	69.4127
23	0.4525	0.3840	108.4006	100.5047	40.6139	275.1938	1.6741	0.9134	11.1621	6.7964	2.6774	73.1690
24	0.5056	0.4406	111.1189	100.5047	47.3941	275.7000	1.6752	0.9060	11.1592	6.8162	2.6774	78.0765
25	0.5578	0.4965	114.3055	100.5013	54.4539	276.4241	1.7245	0.8732	11.1496	6.8448	2.6774	85.8739
26	0.6135	0.5562	117.9249	100.1481	62.2626	277.3293	1.8093	0.8474	11.1286	6.9012	2.6774	96.4982
27	0.6727	0.6197	121.7406	99.1067	70.7011	278.3882	1.9272	0.8635	11.0910	7.0011	2.6774	111.7105
28	0.7350	0.6863	125.6847	97.2315	79.6409	279.5820	2.0521	0.9316	11.0352	7.1684	2.6774	125.2473
29	0.8010	0.7568	129.7497	94.3370	89.0816	280.8962	2.1780	1.0621	10.9570	7.4317	2.6774	132.9985
30	0.8697	0.8299	133.8092	90.3139	98.7335	282.2842	2.2726	1.2684	10.8602	7.7848	2.6774	139.4341
31	0.9392	0.9036	137.6992	85.1849	108.1879	283.6846	2.3183	1.5319	10.7496	8.2216	2.6774	145.6222
32	1.0088	0.9777	141.3334	78.9922	117.1979	285.0564	2.3047	1.8328	10.6300	8.7324	2.6774	151.7067
33	1.0789	1.0520	144.6786	71.7396	125.6396	286.3726	2.2257	2.1557	10.5046	9.3105	2.6774	157.0196
34	1.1492	1.1263	147.6788	63.4998	133.3297	287.5950	2.0771	2.4803	10.3819	9.9111	2.6774	160.6155
35	1.2192	1.2003	150.2784	54.4125	140.0817	288.6863	1.8609	2.7881	10.2679	10.4998	2.6774	164.2103
36	1.2890	1.2739	152.4478	44.5972	145.7787	289.6184	1.5839	3.0638	10.1678	11.0395	2.6774	168.0408
37	1.3589	1.3477	154.1744	34.1137	150.3530	290.3726	1.2478	3.2976	10.0865	11.4882	2.6774	172.2601
38	1.4293	1.4218	155.4346	23.0669	153.7134	290.9294	0.8609	3.4762	10.0252	11.8372	2.6774	175.8249
39	1.4999	1.4961	156.2006	11.6527	155.7654	291.2710	0.4403	3.5878	9.9871	12.0578	2.6774	177.8100
40	1.5707	1.5707	156.4590	0.0087	156.4590	291.3867	0.0003	3.6259	9.9739	12.1361	2.6774	178.0621

Figure 21a - STREAMLINE OUTPUT FROM STRMBL

NORMAL STREAM SURFACE CALCULATION BEGINS
USING GIVEN COEFFICIENT OF VISCOSITY = 0.117000E-03

PSEUDO STREAM SURFACE DATA
FROM CUT 1
AT Z = 95.4640
TOTAL NORMAL LENGTH = 0.391078

VALUES ALONG NORMAL TO BODY...

FROM STREAMLINE 1
AT THETA (ON BODY) = -89.9999 DEG

POINT	Z	R	THETA	X	Y	P	S	U	V	W
1	95.4640	51.0686	-89.9999	0.0001	-51.0686	11.6509	2.6774	-0.0000	-0.2948	10.6624
2	95.4628	51.1121	-89.9999	0.0001	-51.1121	11.6455	2.6768	-0.0000	-0.2958	10.6686
3	95.4616	51.1555	-89.9999	0.0001	-51.1555	11.6401	2.6763	-0.0000	-0.2968	10.6748
4	95.4604	51.1989	-89.9999	0.0001	-51.1989	11.6348	2.6757	-0.0000	-0.2978	10.6809
5	95.4592	51.2424	-69.9999	0.0001	-51.2424	11.6295	2.6752	-0.0000	-0.2989	10.6870
6	95.4580	51.2858	-89.9999	0.0001	-51.2858	11.6242	2.6746	-0.0000	-0.2999	10.6931
7	95.4568	51.3293	-69.9999	0.0001	-51.3293	11.6189	2.6741	-0.0000	-0.3010	10.6991
8	95.4556	51.3727	-89.9999	0.0001	-51.3727	11.6137	2.6735	-0.0000	-0.3020	10.7051
9	95.4544	51.4161	-89.9999	0.0001	-51.4161	11.6085	2.6730	-0.0000	-0.3031	10.7110
10	95.4532	51.4596	-89.9999	0.0001	-51.4596	11.6034	2.6724	-0.0000	-0.3042	10.7170

NORMAL DERIVATIVES AT SAME LOCATIONS...

POINT	N LENGTH	DPDN	DSDN	DUDN	DVDN	DWDN	DVELDN	VELDTN
1	0.0	-0.1249	-0.0127	0.0000	-0.0227	0.1430	0.1435	0.0000
2	0.0435	-0.1242	-0.0127	0.0000	-0.0230	0.1423	0.1429	0.0008
3	0.0869	-0.1233	-0.0127	0.0000	-0.0233	0.1415	0.1421	0.0017
4	0.1304	-0.1226	-0.0128	0.0000	-0.0235	0.1406	0.1412	0.0025
5	0.1736	-0.1219	-0.0128	0.0000	-0.0238	0.1399	0.1405	0.0034
6	0.2173	-0.1210	-0.0128	0.0000	-0.0241	0.1391	0.1397	0.0042
7	0.2607	-0.1205	-0.0128	0.0000	-0.0244	0.1382	0.1389	0.0051
8	0.3042	-0.1197	-0.0128	0.0000	-0.0247	0.1375	0.1381	0.0060
9	0.3476	-0.1186	-0.0128	0.0000	-0.0250	0.1367	0.1374	0.0069
10	0.3911	-0.1173	-0.0129	0.0000	-0.0253	0.1361	0.1368	0.0079

FROM STREAMLINE 2
AT THETA (ON BODY) = -85.4220 DEG

POINT	Z	R	THETA	X	Y	P	S	U	V	W
1	95.4640	51.2321	-85.4220	4.0890	-51.0686	11.7583	2.6772	-0.0163	-0.2945	10.6518
2	95.4628	51.2753	-65.4259	4.0890	-51.1120	11.7530	2.6767	-0.0157	-0.2956	10.6579
3	95.4616	51.3186	-85.4298	4.0690	-51.1555	11.7478	2.6762	-0.0151	-0.2967	10.6639
4	95.4604	51.3620	-85.4337	4.0890	-51.1989	11.7425	2.6756	-0.0146	-0.2979	10.6700
5	95.4592	51.4052	-85.4375	4.0890	-51.2424	11.7374	2.6751	-0.0140	-0.2990	10.6760
6	95.4580	51.4466	-85.4414	4.0890	-51.2858	11.7322	2.6745	-0.0134	-0.3002	10.6820
7	95.4568	51.4918	-85.4452	4.0890	-51.3292	11.7271	2.6740	-0.0129	-0.3014	10.6879
8	95.4556	51.5351	-85.4490	4.0890	-51.3727	11.7220	2.6734	-0.0123	-0.3026	10.6938
9	95.4544	51.5784	-85.4529	4.0690	-51.4161	11.7169	2.6729	-0.0118	-0.3038	10.6997
10	95.4532	51.6217	-65.4567	4.0890	-51.4595	11.7119	2.6723	-0.0112	-0.3050	10.7055

NORMAL DERIVATIVES AT SAME LOCATIONS...

POINT	N LENGTH	DPDN	DSDN	DUDN	DVDN	DWDN	DVELDN	VELDTN

Figure 21b - PSEUDO STREAM SURFACE OUTPUT

SONIC BOOM DATA

FREE STREAM MACH NO. = 26.1000
ANGLE OF ATTACK = 30.0000
GAMMA = 1.1200
STARTING AT Z = 50.0000

DATA TO BE FOUND ON CYLINDER OF RADIUS = 250.0000
AT 40 EVENLY DISTRIBUTED POINTS
OUTPUT EVERY 10 DATA PLANES (COMPUTATIONAL STEPS)

AT STEP 700
Z = 0.660402E 03

INDEX	HC	P	S	U	V	W
1	-1.5708	1.0000	0.0	0.0	13.8108	23.9210
2	-1.4902	1.0000	0.0	0.0	13.8108	23.9210
3	-1.4097	1.0000	0.0	0.0	13.8108	23.9210
4	-1.3291	1.0000	0.0	0.0	13.8108	23.9210
5	-1.2486	1.0000	0.0	0.0	13.8108	23.9210
6	-1.1680	1.0000	0.0	0.0	13.8108	23.9210
7	-1.0875	1.0000	0.0	0.0	13.8108	23.9210
8	-1.0069	1.0000	0.0	0.0	13.8108	23.9210
9	-0.9264	1.0000	0.0	0.0	13.8108	23.9210
10	-0.8458	1.0000	0.0	0.0	13.8108	23.9210
11	-0.7653	1.0000	0.0	0.0	13.8108	23.9210
12	-0.6847	1.0000	0.0	0.0	13.8108	23.9210
13	-0.6042	1.0000	0.0	0.0	13.8108	23.9210
14	-0.5236	1.0000	0.0	0.0	13.8108	23.9210
15	-0.4430	1.0000	0.0	0.0	13.8108	23.9210
16	-0.3625	1.0000	0.0	0.0	13.8108	23.9210
17	-0.2819	1.0000	0.0	0.0	13.8108	23.9210
18	-0.2014	1.0000	0.0	0.0	13.8108	23.9210
19	-0.1208	1.0000	0.0	0.0	13.8108	23.9210
20	-0.0403	1.0000	0.0	0.0	13.8108	23.9210
21	0.0403	1.0000	0.0	0.0	13.8108	23.9210
22	0.1208	1.0000	0.0	0.0	13.8108	23.9210
23	0.2014	1.0000	0.0	0.0	13.8108	23.9210
24	0.2819	1.0000	0.0	0.0	13.8108	23.9210
25	0.3625	1.0000	0.0	0.0	13.8108	23.9210
26	0.4430	1.0000	0.0	0.0	13.8108	23.9210
27	0.5236	1.0000	0.0	0.0	13.8108	23.9210
28	0.6042	1.0000	0.0	0.0	13.8108	23.9210
29	0.6847	1.0000	0.0	0.0	13.8108	23.9210
30	0.7653	8.6333	0.8962	2.7111	12.4647	24.3062
31	0.8458	4.6277	1.6488	1.2488	10.7615	23.4038
32	0.9264	2.6368	2.0147	-0.0347	10.1836	23.0840
33	1.0069	1.6773	2.2607	-1.1147	9.8309	22.3604
34	1.0875	1.2045	2.4387	-2.2003	9.5024	21.4749
35	1.1680	0.9108	2.5899	-3.2451	9.0350	20.5599
36	1.2486	0.7451	2.7254	-4.1283	8.3530	19.6477
37	1.3291	0.7149	2.8538	-4.6716	7.3395	19.0988
38	1.4097	1.9142	3.0574	-1.3095	4.5444	18.7060
39	1.4902	2.7432	3.2551	-1.3244	3.9880	18.2131
40	1.5708	2.5597	3.3075	-0.0001	3.7832	18.4587

Figure 22 - SONIC BOOM DATA CYLINDER OUTPUT

STORAGE REQUIREMENTS AND COMPUTER TIME FOR QUICK

Using the IBM G-compiler, QUICK requires approximately $128K_{10}$ bytes of core to compile ($\approx 40K_8$ words), and $176K_{10}$ bytes to execute ($\approx 54K_8$ words). CDC requirements may somewhat exceed the figures in parentheses since CDC machines do not use half-word instructions and IBM machines do.

These core requirements are true with the code dimensioned to allow a maximum of:

10 arcs pre-cross section (maximum value of J*)

10 segments per body line model (maximum value of N*)

10 cross-sectional models (maximum value of K*)

25 body line models (maximum value of M*)

Of course, these may be adjusted if required.

QUICK run time varies greatly with the user requested output options. On the IBM 370/168, a sample run for a simple 10° cone with afterbody, exercising modes 1, 2, 3, 4 and 5 at four x-stations each, nineteen (19) circumferential points per station in mode 2, and seven circumferential points per station in modes 4 and 5 required approximately 30 cpu seconds (of which, less than a third would be attributable to the initial defining and checking tasks). On a more complex vehicle, exercising only mode 2, assembly of the model and output of data for thirteen cross-sectional stations, using theta increments of one degree (181 points), required approximately 20 cpu seconds.

*Each dimensioned variable in QUICK is defined in the Symbol list for QUICK in terms of these integers, unless otherwise specified.

STORAGE REQUIREMENTS AND COMPUTER TIME FOR STEIN

The storage used in STEIN is divided, of course, between logic and variables. Using fixed dimensions at a maximum grid of 40 x 50 (which could be required for very complex vehicles) the core needed to store the variables is $180K_{10}$ bytes (on the IBM 370/168). The core required for logic without overlay is $400K_{10}$. So that $580K_{10}$ bytes of computer core is needed to run STEIN in this configuration. When STEIN is overlayed, the core required for the logic becomes $160K_{10}$ bytes. And if the dimensions of the variables were made to vary with the problem the expression for core required for this part of the code would be (NDIMEN x MDIMEN) x 17 + MDIMEN x 70 + NDIMEN x 40 + $50K_{10}$ where NDIMEN is the number of points in the radial direction and MDIMEN is the number of points in the circumferential direction. For simple geometries with small shock layers these can be as small as 10 x 10.

Presently the code is dimensioned to allow a maximum of:

40 grid points in the radial direction (maximum value of N*)

50 grid points in the circumferential direction (maximum value of M*)

4 regions in the radial direction (maximum value of L*)

4 regions in the circumferential direction (maximum value of I*)

The computer time required by STEIN depends in general upon length of vehicle and free stream condition. One of the longest running calculations was that of a shuttle orbiter flying at $M_\infty = 10$ and an angle of attack of 30°. This calculation took about 2 hours on the CDC 6600. Some of the reasons for this running time are:

(1) At large angle of attack the shock layer on top of the body becomes large (requiring 25 mesh points in the radial direction for accuracy). These mesh points are also across the

*Each dimensioned variable in STEIN, STRMBL and BOOM is defined in the appropriate symbol list in terms of these integers, unless otherwise specified.

shock layer on the bottom of the body which makes the physical distance between mesh points small and caused DZ (stable marching step) to become very small. With this small value of DZ it takes 3000 steps to compute the entire vehicle.

(2) On blunt nose vehicles the body entropy is very large causing small Mach numbers on the body. As the local axial Mach number approaches one, DZ approaches zero. On the forebody of blunt nose vehicles this condition exists causing the calculation to slow down there.

The computer time required to compute the flow field about an H.R.A. configuration at $M_\infty = 6$ and $\alpha = 0$, was about 1 hour of CDC 6600 time. The same number of mesh points at each axial station were computed in this case and the Shuttle orbiter case but the step size DZ was doubled because of the small angle of attack and the low body entropy. Finally, the time required to compute the flow field about a simple slab delta wing ($M_\infty = 9.6$ and $\alpha = 30^\circ$) from the nose to 15 nose radii down stream was about 15 min.

The computer time/mesh points depend significantly upon two parameters:

(1) Vehicle geometry (Shuttle orbiter or simple slab delta wing)

(2) Gas model used in thermodynamics (ideal gas or chemical equilibrium)

There is also a slight dependence on the number of imbedded shocks in the flow field, but this comparison is hard to make since one cannot run the same vehicle with and without imbedded shocks.

STORAGE REQUIREMENTS AND COMPUTING TIMES FOR STRMBL

With the IBM 370/168 H-compiler, STRMBL requires roughly $240K_{10}$ bytes of core to compile ($\approx 74K_8$ words), and approximately $354K_{10}$ bytes ($\approx 131K_8$ words) to execute.

Approximately eight cpu minutes were required to run STRMBL on the IBM 370/168 for an 89B shuttle calculation of about 225 computational steps (from Z = 50 to Z = 790; this piece of the flow field computation required approximately 22 cpu minutes.)

STORAGE REQUIREMENTS AND COMPUTING TIMES FOR BOOM

BOOM requires (for the IBM G-compiler) approximately $122K_{10}$ bytes to compile ($\approx 37K_8$ words), and $190K_{10}$ bytes to execute ($\approx 60K_8$ words).

In the same shuttle calculation as above, BOOM required about 3.6 cpu minutes.

PART 2 PROGRAMMER-ORIENTED DOCUMENTATION

OVERALL FLOW OF LOGIC

QUICK consists of three basic sets of routines with distinct functions. The first of these reads the input data and begins to assemble the mathematical model - this is the defining portion of QUICK. The second set of routines perform some logical checking of the math model, and correlates it to the input data - this is the checking portion of QUICK. Included in this set is a routine which reads user requests to exercise the math model, and calls upon the third and remaining portion of QUICK - the interrogating or exercising section, called SUB-QUICK in this report; see Fig. 23.

STEIN utilizes a finite difference marching technique, so that given the flow field at one axial station z the code computes the flow field at $z = z + Dz$. This process is repeated until the desired station is reached. Figure 24 shows a flow chart of the overall logic used in STEIN.

STRMBL performs two basic functions in two nearly independent steps. The first step reads all of the flow field data planes from tape and traces streamlines for the length of the vehicle in this run. Flow variables are evaluated and output along these streamlines. The link with the second step is the establishing of the cutting planes at which body surface normals will be taken to determine the pseudo-stream-surfaces (p-s-s). The data tape is rewound and control transferred to the second portion of the code which, reading through the entire data tape a second time, uses SUB-QUICK to establish the body normals and then evaluate the flow variables and their derivatives in the constructed p-s-s. An end-of-file (EOF) mark on the tape terminates the job.

<u>BOOM</u> simply reads through the same flow field data tape used by STRMBL, and interpolates for flow variables on the data cylinder every JA data planes. (JA is a user input.) An end-of-file (EOF) mark on the tape terminates the job.

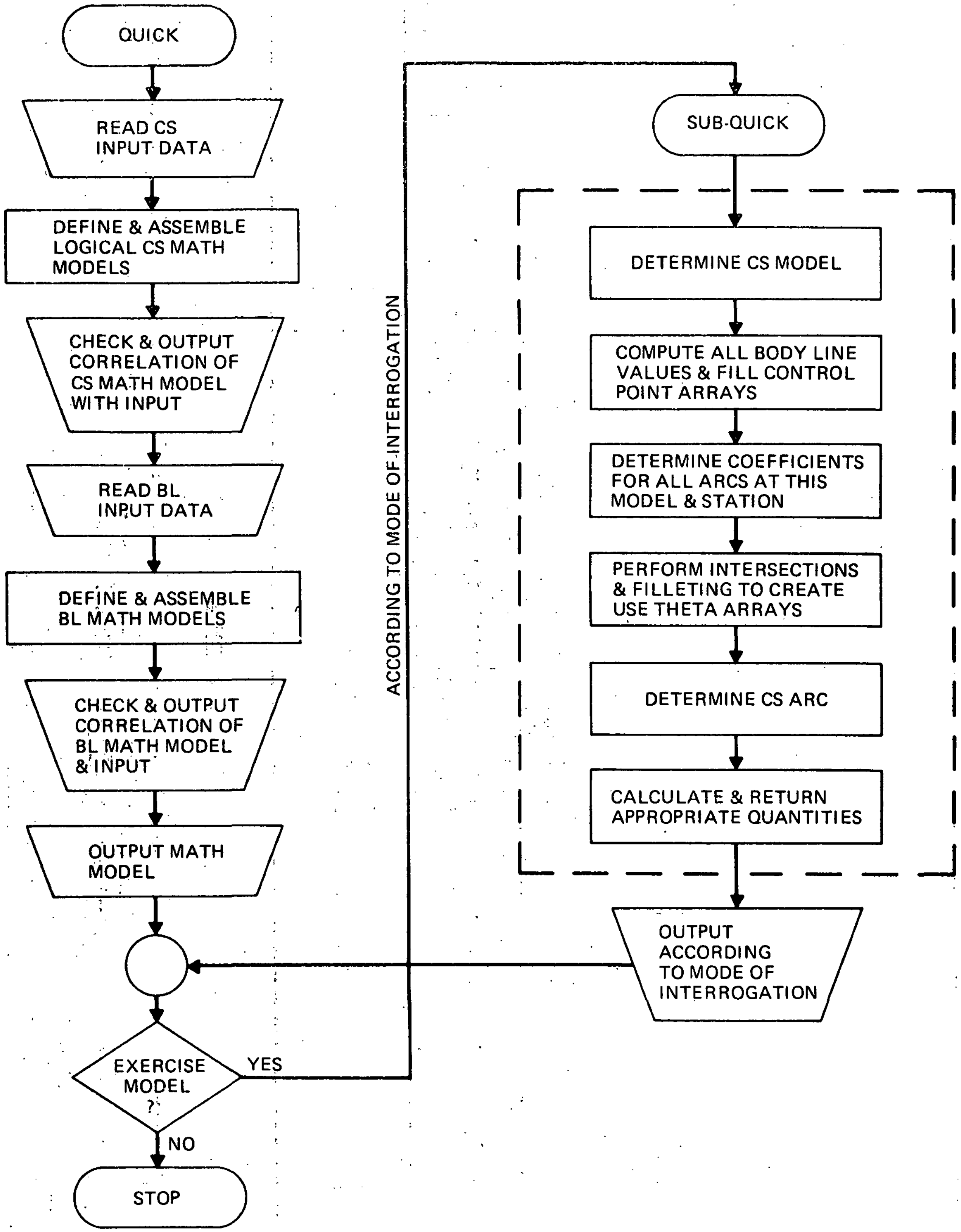


Figure 23 - QUICK OVERALL LOGIC

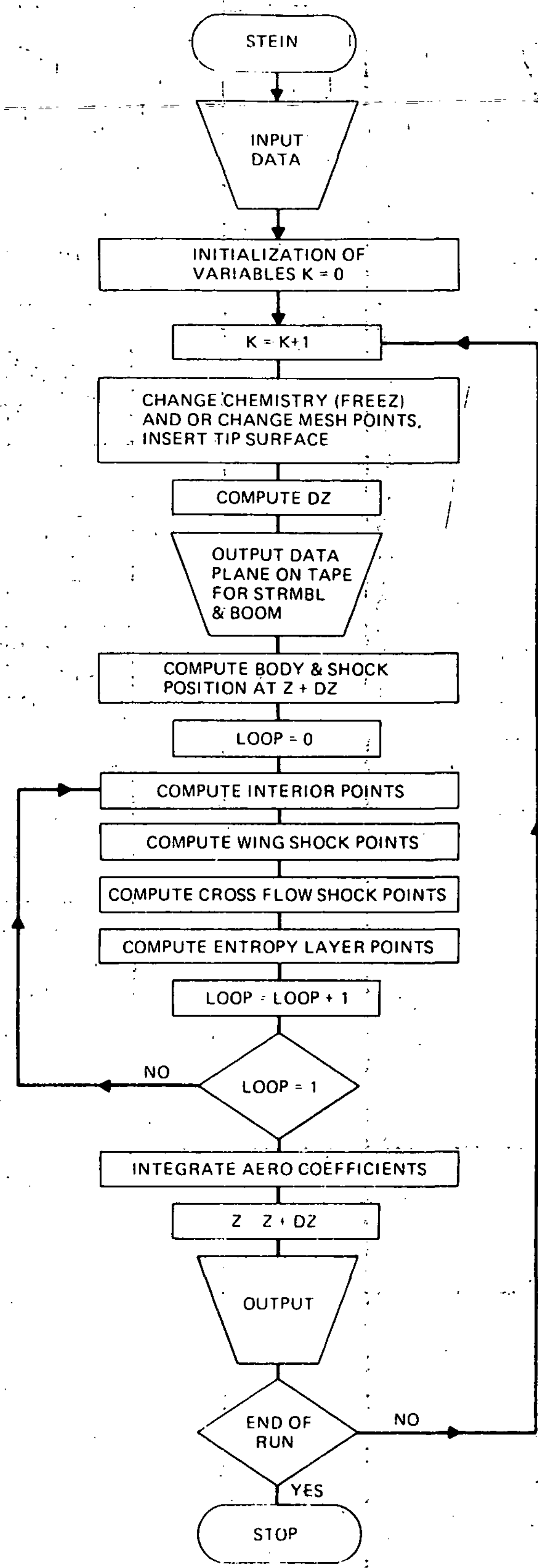


Figure 24 - STEIN OVERALL LOGIC

OVERLAY DESCRIPTION

The only code we found it necessary to overlay was STEIN. It was found that the core requirements could be reduced by 50% using a simple overlay.

The routines in the root segment (No. 1) (always in core) are:

STEIN (main routine), TIPSUR, UPDATE, CSGEOM, BLGEOM, CSCALC, IMAP, MAP, BODY, NINTER, MINTER, PRAN, RANK, GAS, MOLEH, MOLES, EXPAN, OBSHK, SHTEST, SHTIP, VDOTV, MDOTV, THELIM, CSMINT, CSCALC, CURVES, CSMSET, CSMCOE, CSMFLT

Segment 2:	INIT, GEOMIN
3	BOUND
4	SHARP
5	FREEZ
6	NMESH
7	ENTRLA
8	SHMOVE
9	MMESH
10	OUTPUT
11	BLOUT
12	POINTS
13	COEF
14	NSHOCK
15	MSHOCK
16	MREGIO
17	CFL
18	SHRPIN, SHPEDG
19	ARCONT, AEROCF, KAREN
20	NREGIO, INTSEC
21	MSURFA, MTEST
22	NSURFA, NTEST

SUBROUTINE DESCRIPTIONS

SUBROUTINE DESCRIPTION FOR QUICK

BLGEOM assigns body line model values and derivatives to control point coordinates.

BLMCHK correlates and checks the input data deck and the indices for the generated body line math models.

BLMDEF defines body line models from the input data.

BLMSET controls the determination values and first and second derivatives for all body line models at a given x-station.

CSCALC computes radial position and derivatives for specified cross section model, arc, and θ'.

CSGEOM is the main subroutine in the SUB-QUICK (look-up or exercising) portion of the QUICK system. It is called to establish $r' = f(\theta',x)$. It calls appropriate subroutines to evaluate body line values and construct cross section geometry at a given x-station. It is used for all geometry model interrogation.

CSMCHK correlates and checks the input data deck and the indices for the cross sectional math model.

CSMCOE composes the equations which are to define the cross section geometry at a given station.

CSMDEF logically defines the cross section models from the input data.

CSMFLT creates control point definitions to permit the insertion of a smooth fillet between cross sectional arcs.

CSMINT locates user specified intersections between cross sectional arcs and adjusts their use-theta limits.

CSMSET sets up the control point coordinate arrays used to define the cross section geometry at a specified x-station.

CURVES calculates values and first and second derivatives for individual curve fits.

DLOKUP is a simple dictionary look-up routine. It assigns an index to match an input name to a codeword list, but is not capable of adding new items to that list.

DSETUP is an adapting dictionary look-up routine. New items are added to a codeword list, an index (counter) is returned for the codeword, and an indicator (INEW) is set equal to 1 when a new item is encountered.

GEMCHK exercises the mathematical model at user request via MODE1, MODE2, etc.

GEMOUT outputs the math model generated by the defining portions of QUICK (this is referred to as the QUICK intermediate data deck). Also ensures that all body lines required by a cross-sectional model are defined for the range of that model.

GEOMIN reads in the math model generated by the defining portion of QUICK and output by GEMOUT (the QUICK intermediate data deck).

KRVDEF calculates coefficients for the various curve fits associated with body line math models.

MDOTV performs matrix multiplication of a vector.

MODE1 is called by GEMCHK to trace body line model values.

MODE2 is called by GEMCHK to create cross sectional cuts.

MODE3 is called by GEMCHK to examine the cross sectional modeling in the region about control points. Mode -3 plotting is transferred to MODE1 (multiple body line traces to create plan and profile views).

MODE4 is called by GEMCHK to exercise subroutine SLOPE and examine the numerically formed derivatives at various x-stations along traces at a constant value of θ'.

MODE5 is called by GEMCHK to examine the surface unit normals.

MODE7 is called by GEMCHK to examine all defined arcs at a given x-station. This routine is used for plotting purposes only.

QUICK is the main routine. It sets the read and write units and controls the flow of the defining, checking, and exercising portions of the QUICK system.

SLOPE forms a numerical estimate of the first derivatives of a supplied set of points. It is used as an independent check on computed QUICK derivatives.

THELIM creates and controls use-theta arrays to establish continuity in the cross sectional model.

VDOTV computes a vector dot product.

SUBROUTINE DESCRIPTION FOR STEIN

AEROCF	performs the integration of pressure forces and moments on the body for aerodynamic coefficient calculations.
ARCONT	controls the integration of pressure forces and moments on the body for aerodynamic coefficient calculations.
BLGEOM	(This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)
BLMSET	(This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)
BLOUT	outputs the entire flow field on tape at every computational step, to be used by STRMBL and BOOM.
BODY	computes the position (B(M)) of the body in the mapped space and its derivatives (BH(M) and BZ(M)). The body is defined in the physical space, in the routine BODY an iterative procedure is used to find the position of the body in the mapped space, and then BH(M) and BZ(M) are computed analytically.
BOUND	computes the position and derivatives of all boundaries of the computational space (CC(M,L), CCY(M,L), CCZ(M,L), HCZ(N,I) and HCX(N,I)) from their positions in the mapped space.
CFL	computes the step size DZ that satisfies the Courant-Friedrichs-Lewy criterion for stability. It is called from the main routine once per step.
COEF	computes the coefficients used in the conformal mappings and their derivatives. The positions of the top, bottom, and wing tip are transferred to COEF through common. These geometry variables are used to compute the coefficients of the mapping which are then stored in common.
CSCALC	(This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)

CSGEOM (This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)

CSMCOE (This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)

CSMFLT (This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)

CSMINT (This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)

CSMSET (This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)

CURVES (This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)

ENTRLA is used to compute, detect, and collapse the entropy layer surface. It is called in each level of the MacCormack scheme (LOOP = 0 and LOOP = 1). If IENTE is input as zero, control will return from ENTRLA immediately but if IENTE $\neq$ 0 for the points on the entropy layer surface which have already been detected (IENT(M) = 1) the position and dependent variables will be computed. When ENTRLA is called with LOOP = 1, after the dependent variables are computed, additional entropy layer points are looked for and all entropy layer points are tested to see which are to be collapsed (IENT(M) = 2) at the current station.

EXPAN computes the flow through a 2-D centered expansion corner. Given the upstream Mach number (XM1), GAMLO(N,M) and the flow deflection (DELTA). EXPAN will compute the conditions after the expansion (pressure ratio P2QP1, temperature ratio T2QT1, Mach number XM2 and the slope (BETA) of the first expansion wave).

FREEZ is called at a station Z = ZFREEZ when the thermodynamics of the flow field is in equilibrium. In FREEZ an equivalent "frozen state" is computed at each mesh point, IGAS is set to 2 so that the thermodynamics of the flow is frozen from that station on. FREEZ is called, at most, once per vehicle.

GAS relates all the thermodynamic variables for ideal gas (IGAS = 0), equilibrium air (IGAS = 1) and frozen gas (IGAS = 2). If IN = 1, P ($\ell n\ p/p_\infty$) and S (entropy) are input; if IN = 2, P and H (enthalpy) are input; if IN = 3, S and H are input. GAS will compute GAMLO (N,M) and T(N,M) and then return if IOUT = 1. If IOUT $\neq$ 1, GAS will compute the temperature (THE) and the variable P,S or H that is not input in addition to GAMLO(N,M) and T(N,M).

GEOMIN (This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)

IMAP is the inverse mapping subroutine. It uses X and Y (physical Cartesian coordinates in the Z = constant plane) to compute R and THE (polar coordinates in the mapped space). The index I indicates which value of the coefficients (gotten in common) are to be used -- those at Z for I = 1, those at Z + DZ for I = 0.

INIT is used to initialize variables. In INIT all input data is read and then most variables are initialized. INIT is called only once per run.

INTSEC is called from NREGIO when two wing shock type shock points intersect. In INTSEC the conditions behind the resulting shock are computed.

KAREN computes the area of the discrete triangular facets and sets up the unit normals used to integrate pressure forces on the body.

MAP — is the mapping routine. It uses R and THE to compute X and Y (see description of IMAP) with the index I indicating at which value of Z the coefficients are to be used (as in IMAP). If ID = 0, X and Y are computed and control is returned. If ID = 1, the derivatives of the mapping, XR, YR, XZ, YZ, XH, YH (x_r, y_r, x_ζ, y_ζ, x_θ, y_θ) and RX, RY, RZ, HX, HY, HZ, (r_x, r_y, r_z, θ_x, θ_y, θ_z) are also computed and returned in the argument list. In POINTS, for the body calculation, the second derivatives of the mapping are also needed, so that for ID = 2, RXR, RYR, RZR, HXR, HYR, RXH, RYH, RZH, HXH, HYH, HZH, RXZ, RYZ, RZZ, HXZ, HZZ (r_{xr}, r_{yr}, r_{zr}, θ_{xr}, θ_{yr}, $r_{x\theta}$, $r_{y\theta}$, $r_{z\theta}$, $\theta_{x\theta}$, $r_{x\zeta}$, $r_{y\zeta}$, $r_{z\zeta}$, $\theta_{x\zeta}$, $\theta_{y\zeta}$, $\theta_{z\zeta}$) are computed and stored in common.

MDOTV — (This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)

MINTER — plays the same role as NINTER but for circumferential interpolation.

MMESH — is called at Z = ZMADD to add MDEL points in the circumferential direction. These points will be divided proportionately between all the regions in the circumferential direction.

MOLEH — uses curve fits of GAMLO(N,M), T(N,M), S(N,M) and the temperature as functions of P($\ell n\ p/p_\infty$) and H (enthalpy) for air in equilibrium.

MOLES — uses an iteration to compute GAMLO(N,M), T(N,M), H and temperature (THE) from P and S for air in equilibrium.

MREGIO — shifts mesh points in the circumferential direction. There are no provisions for crossflow shocks intersecting.

MSHOCK — serves the same purpose as NSHOCK but for crossflow shocks.

MSURFA — serves the same purpose as NSURFA but for crossflow shocks and surfaces.

MTEST serves the same purpose as NTEST but for crossflow shocks. Crossflow shock points started as infinitely weak shocks.

NINTER is a general purpose interpolation routine. At some value of M, NINTER interpolates from an old mesh with NC(L) mesh points in LC regions onto a new mesh with NCN(L) points in LCN regions. The positions of the old shocks are C(M,L) and those of the new shocks are CN(M,L).

NMESH is called at Z = ZNADD to add NDEL points in the radial direction. These points will be divided proportionately between all regions in the radial direction.

NREGIO shifts mesh points in the radial direction as wing type shocks approach each other. When two wing type shocks are close enough to each other at some value of Y, they are intersected at that point, the outer shock being considered the resulting shock and the inner shock becoming an "arbitrary surface" at this point. When all the points on one shock intersect another, this shock is eliminated as a boundary.

NSHOCK computes the high pressure side of the wing type shocks, including the bow shock. NSHOCK is called from the main routine in each level of the MacCormack scheme. After the interior points have been computed in level one of the MacCormack scheme NSHOCK uses the predicted values of the dependent variables on the low pressure side of the shock to integrate to a value of CZN(M,L). After level two of the MacCormack scheme the corrected values of the dependent variables on the low pressure side of the shock and CZN(M,L) compute in level one, are used to recompute the high pressure side of the shock. The bow shock is computed only in level one since the flow on its low pressure side is constant. The position and derivatives (CH(M, L) and CZ(M,L)) of the wing shock type surfaces are also computed in NSHOCK.

NSURFA is used to rearrange the mesh when wing type shocks and wing shock surfaces are first inserted in the flow field. This routine is called after a shock point has been detected; in it the arbitrary surface is initialized. A new grid is defined and the dependent variables are interpolated.

NTEST detects wing type shock points. If Z is not between Z1NSH(J) and Z2NSH(J), for some value of J, control is returned from NTEST. Once shock points are detected the initial jump conditions are gotten by extrapolating from either side and then CZN(M,L) and CHN(M,L) are computed.

OBSHK serves the same purpose as EXPAN but for a 2-D wedge compression. Both OBSHK and EXPAN are used in the sharp leading edge wing calculation.

OUTPUT outputs on unit IWRIT the dependent and independent variables at each output station. The user specifies ZWRIT1 (initial output station), DZWRIT (output interval) and ZWRIT2 (last output station). The user can also specify NSOUT and ZSOUT for additional output. The maximum number of steps between output stations is JA and this routine will be called if execution is terminated for any reason. When requested, aerodynamic coefficients are also output from this routine. OUTPUT also writes (on unit IPUNCH) the starting plane data for the next run at Z = ZEND or K = KA (only if IPUNCH > 0).

POINTS computes all the dependent variables at interior points, body points, and on the low pressure side of all shock waves. For the portion of the internal boundaries that are not shocks the dependent variables are set equal across them in POINTS. POINTS is called from the main routine for each level of the MacCormack integration scheme. In POINTS the body second derivatives BHH, BZZ, and BHZ are also computed.

PRAN computes the flow through a Prandtl-Meyer centered expansion for equilibrium or ideal gas. Given P($\ln \bar{p}/\bar{p}_\infty$) on either side of the expansion, the entropy (constant through the expansion) and the velocity in the plane of the fan (VN1) PRAN computes the change in flow direction DXNU.

RANK computes the flow through a shock. Given VN1 (velocity normal to the shock), GAM1 (the value of GAMLO(N,M)), P1 $\ln(p/p_\infty)$, S1 (entropy), T1(p/ρ), and H1 (enthalpy), all on the low pressure side of the shock, RANK computes these quantities on the high pressure side of the shock.

SHARP computes the exact solution for the flow over a sharp circular cone at zero angle of attack (with half cone angle CONE) (for attached shocks). It also give an approximate solution for sharp cones at small angle of attack. SHARP is called once per run only if ICASE is input as 1.

SHMOVE computes the positions and derivatives in the Z = constant plane of all shocks (crossflow and wing type). SHMOVE is called once per step from main. HN(N,M) is also computed here.

SHPEDG computes the body unit normal components at a given fuselage station (X) on counterclockwise first (ILOHI = 1) or last (ILOHI = 2) cross section arc ending or beginning with a control point at a specified angle (THE).

SHRPIN iterates to find the exact location of the start of a sharp edge. Then it sets up a call to SHPEDG to establish the body normals.

SHTEST is used in the initial setup for starting a sharp leading edge wing. In SHTEST the mesh is adjusted to accommodate a sharp leading edge shock in the wing plane or top or bottom symmetry plane.

SHTIP calculates the flow variables behind a sharp leading edge wing. In SHTIP, given the conditions in front of the sharp tip, the conditions behind the expansion or compression at the tip are computed.

STEIN is the main program of this code. It is used for control mainly. In STEIN the geometry test is generated, some initialization is performed, the marching loop is entered (i.e., ZN = Z + DZ) and finally, the routines that detect shocks or rearrange mesh points are called.

THELIM (This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)

TIPSUR computes the position and derivatives (HSN(N,I), HSRN(N,I), and HSZN(N,I)) of the wing tip crossflow surface.

UPDATE is called once in each level of the McCormack scheme to "update" the dependent and independent variables. In UPDATE the symmetry conditions (U(N,1) = U(N,MC(IC) + MREG(IC)) = 0 and CH(1,L) = CH(MC(IC) + MREG(IC),L) = 0) are also imposed.

VDOTV (This routine is used both in STEIN and QUICK, it is described in the section on QUICK routines.)

SUBROUTINE DESCRIPTION FOR STRMBL

BLDEL establishes the length of each line, in the direction of the body surface normal, which makes up the p-s-s. Currently this is an approximation for the boundary layer thickness on a flat plate $\delta = 5\sqrt{\frac{\nu z}{M\sqrt{\gamma}}} = 5 * z/\sqrt{Rez}$.

BLGEOM (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

BLMSET (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

BRCKT examines the distribution of mesh points in the current data plane to determine those points which will bracket a specified location.

BRCKTO examines the distribution of mesh points in the previous data plane to determine those points which will bracket a specified location.

CSCALC (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

CSGEOM (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

CSMCOE (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

CSMFLT (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

CSMINT (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

CSMSET (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

CURVES (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

DELTHE controls the determination of flow variables on a given streamline at the current station (Z), computes $d\theta'_S/dz$ for the given streamline, integrates to find θ'_S (circumferential location of the streamline) and S_η (arc length measured along the streamline), and determines r_S (radial position of the streamline). (θ'_S, S_η, and r_S at Z + DZ).

FLINE is a simple function used for a line (where y = f(x)), determined from two distinct points, to calculate y* at a specific x*.

GEOMIN (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

INOUT initializes all I/O units.

INTERH performs a simple, second order interpolation in M (circumferential direction) at a specified N.

INTERR performs a simple, second order interpolation in N (radial direction) at a specified M.

INTRH1 performs a simple, second order interpolation in M (circumferential direction) for variables only evaluated at the body (a function of M only).

INTR2D performs a two dimensional, second order interpolation for quantities at a specified location.

INTR3D performs a three dimensional interpolation for any variable. The z-location of the point of interest must lie between the previous and current data planes.

LOCATE determines the location (z′, r′, θ′) of a given point lying along the body surface normal taken at a specified z and θ′.

MAIN2 is a subroutine, but acts as a second main program once STRMBL has established the z and θ′ locations at which body surface normals are to be taken to establish the pseudo-stream-surfaces (p-s-s). The data tape is rewound just prior to entry into MAIN2, which then proceeds to search the flow field data, interpolating in three dimensions, and dynamically allocating storage to find, store, and output all quantities of interest in the p-s-s.

MDOTV (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

NOUT gives printed output of flow variables in the pseudo-stream-surfaces (p-s-s) and forms numerical derivatives of these variables in the p-s-s and outputs them.

SOUT gives printed output of location and flow variable values for a given streamline.

STRMBL is the main routine. It reads data from cards and tape, calls the integrating and output routines, and sets up the stations at which the cuts will be taken for body surface normals to establish the pseudo-stream-surfaces.

THELIM (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

VDOTV (This routine is used both in STRMBL and QUICK, it is described in the section on QUICK routines.)

SUBROUTINE DESCRIPTION FOR BOOM

BLGEOM (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

BLMSET (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

BOOM is the main routine. It reads data from cards and tape, calls the appropriate interpolation routines, and outputs the data cylinder computed quantities.

BRCKT1 examines the distribution of mesh points to determine those points which will bracket a specified location. An INDEX is returned to indicate that the point was found in the field (INDEX = 0), inside the body (INDEX = 1), or in the free stream outside the bow shock (INDEX = 2).

CSCALC (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

CSGEOM (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

CSMCOE (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

CSMFLT (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

CSMINT (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

CSMSET (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

CURVES (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

FLINE is a simple function used for a line (where $y = f(x)$), determined from two distinct points, to calculate y^* at a specific x^*.

GEOMIN (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

INOUT (This routine is used both in BOOM and STRMBL, it is described in the section on STRMBL routines.)

INTERH performs a simple, second order interpolation in M (circumferential direction) at a specified N.

INTERR performs a simple, second order interpolation in N (radial direction) at a specified M.

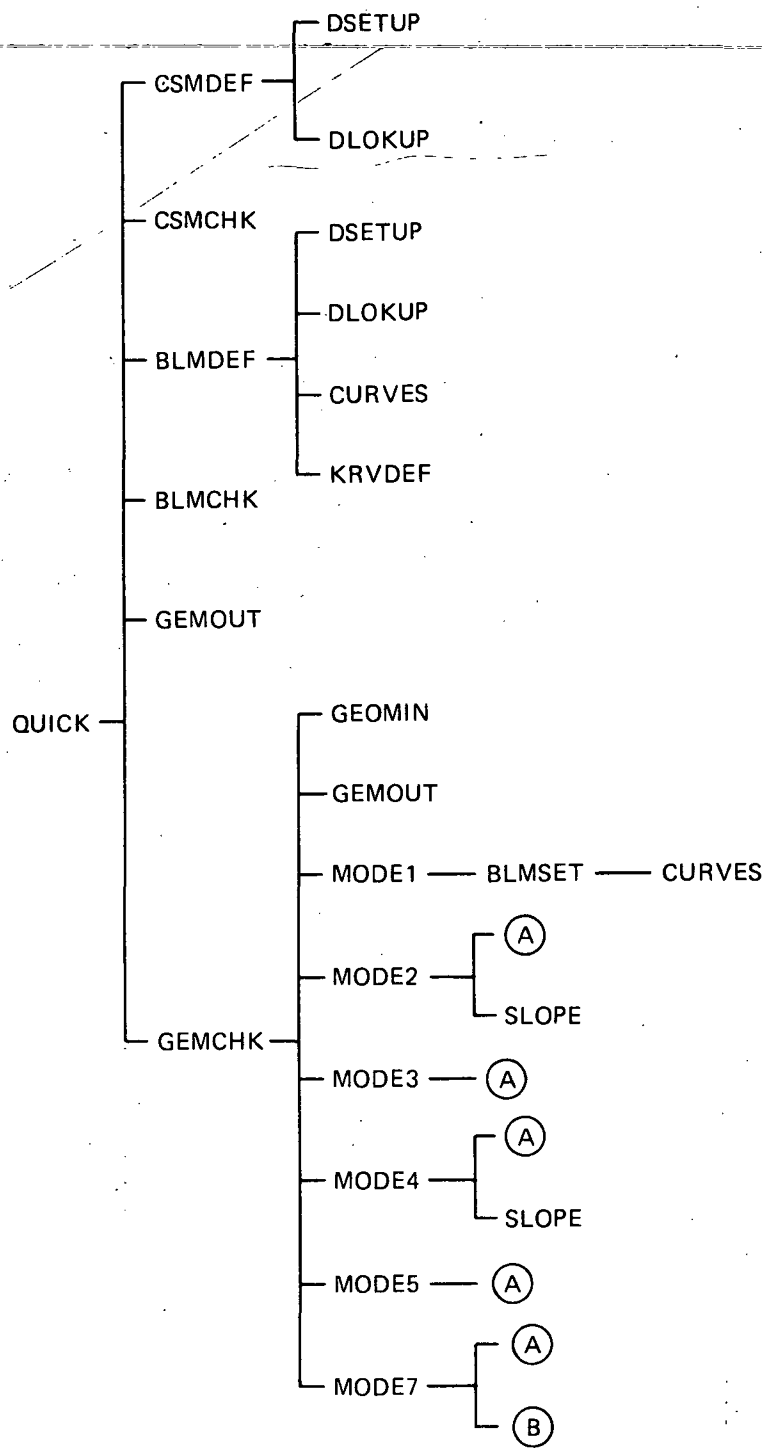
INTR2D performs a two dimensional, second order interpolation for quantities at a specified location.

MDOTV (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

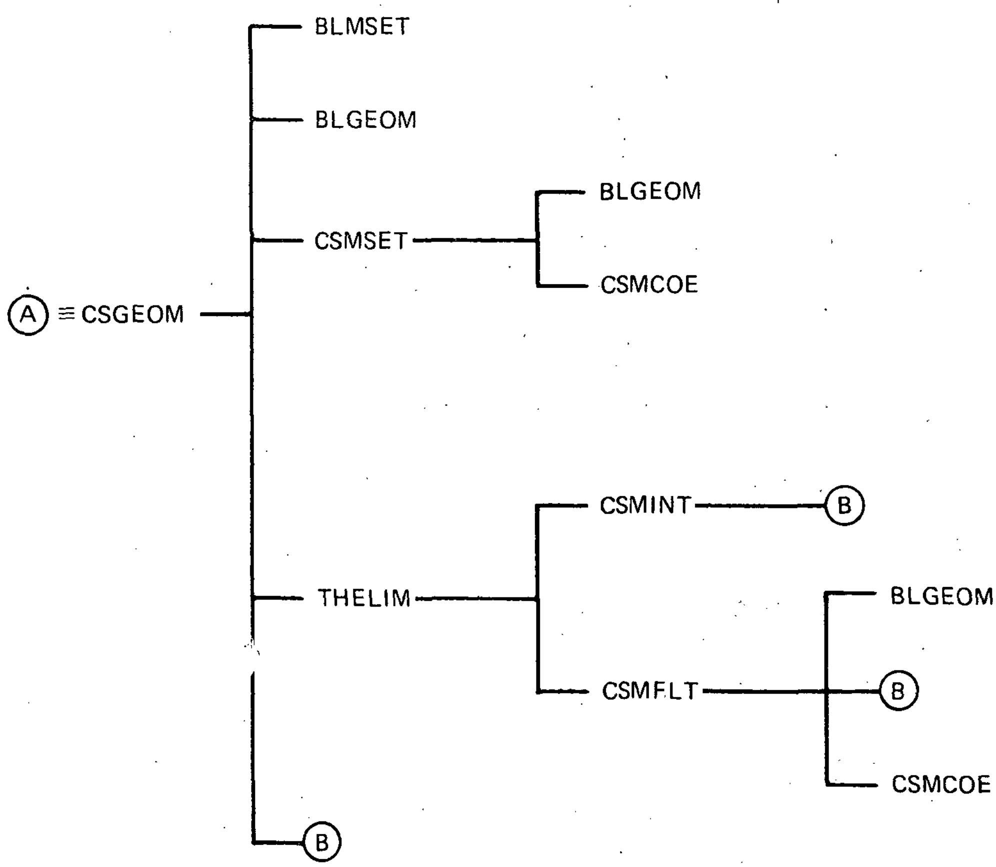
THELIM (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

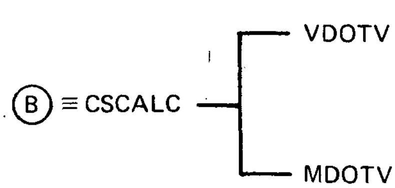
VDOTV (This routine is used both in BOOM and QUICK, it is described in the section on QUICK routines.)

QUICK TREE DIAGRAM

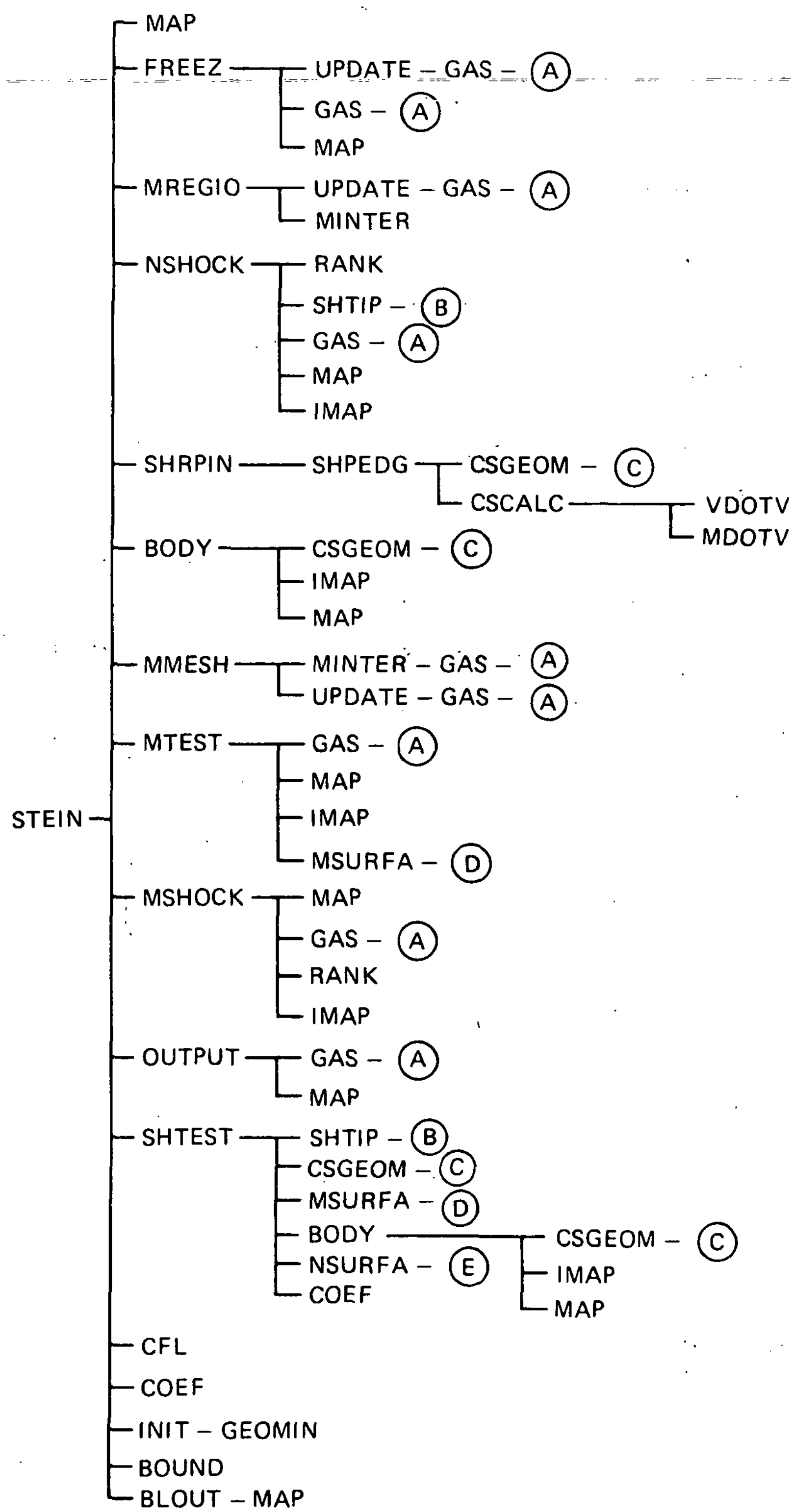


QUICK TREE DIAGRAM (CONT'D)

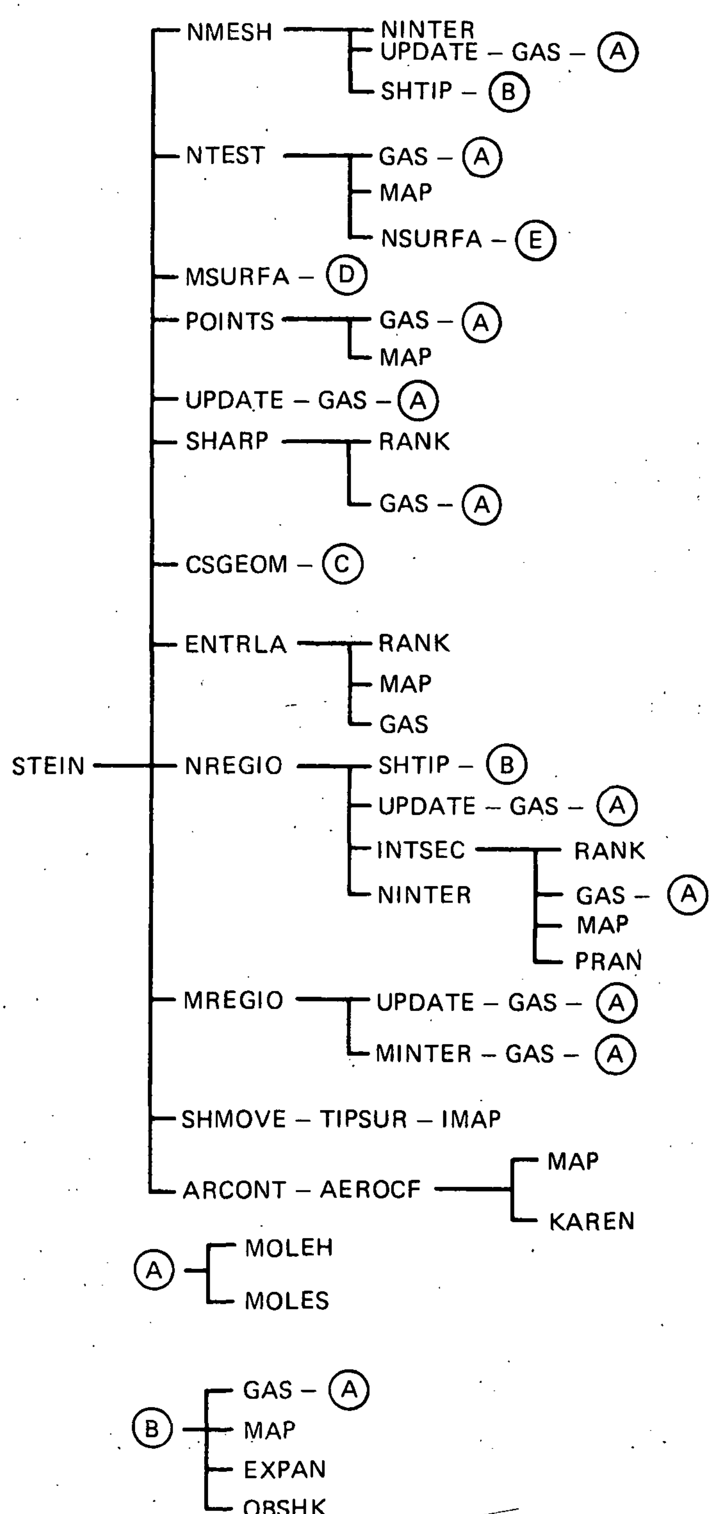




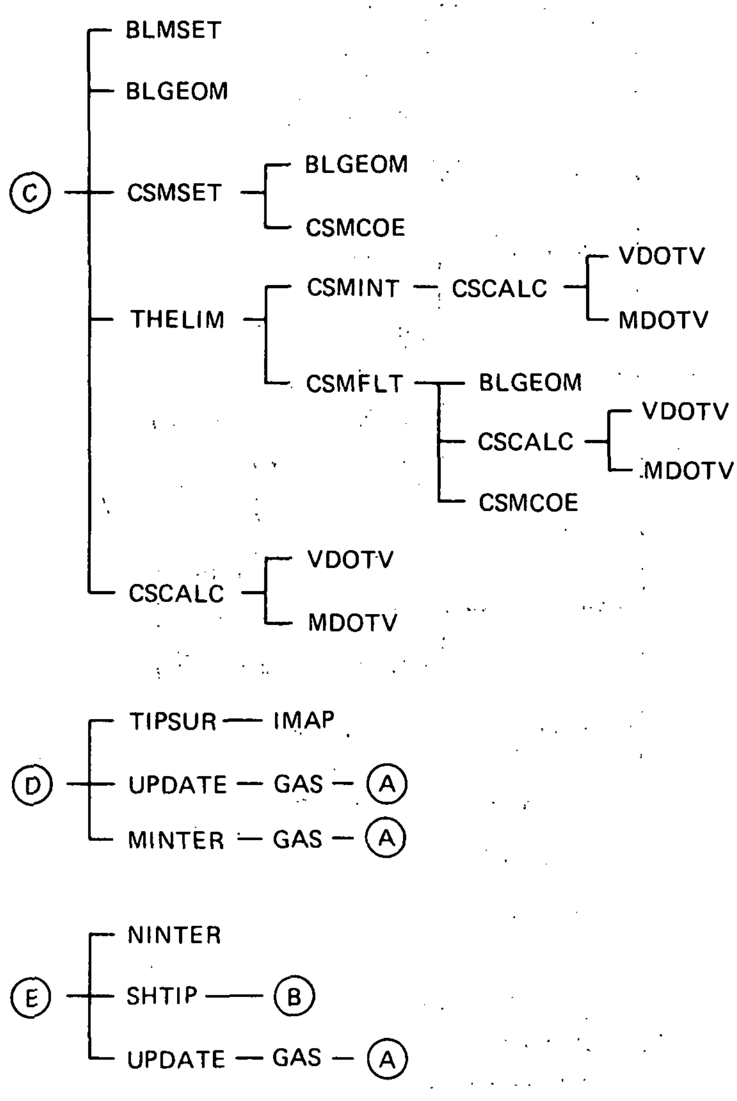
STEIN TREE DIAGRAM



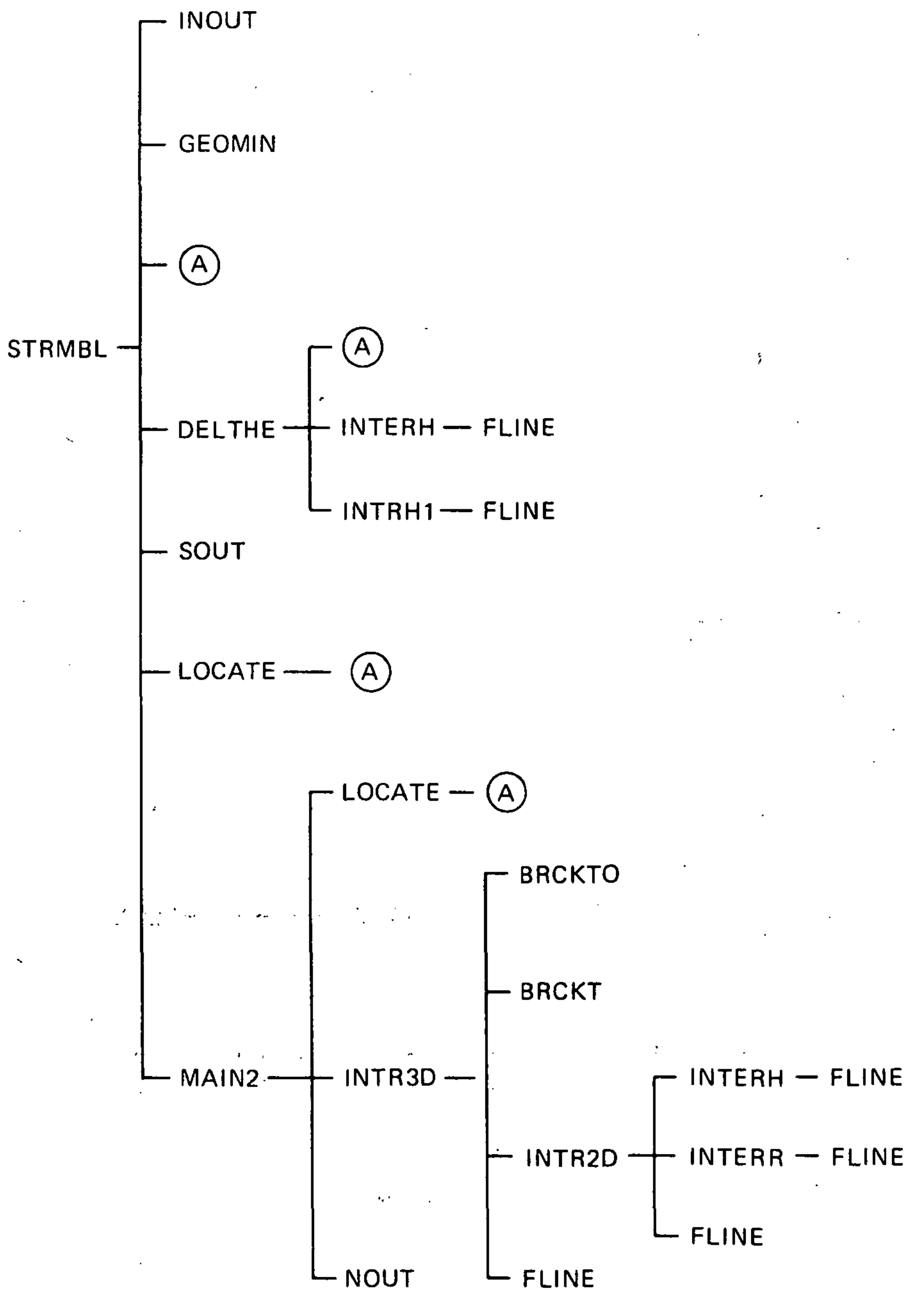
STEIN TREE DIAGRAM (CONT'D)



STEIN TREE DIAGRAM (CONT'D)

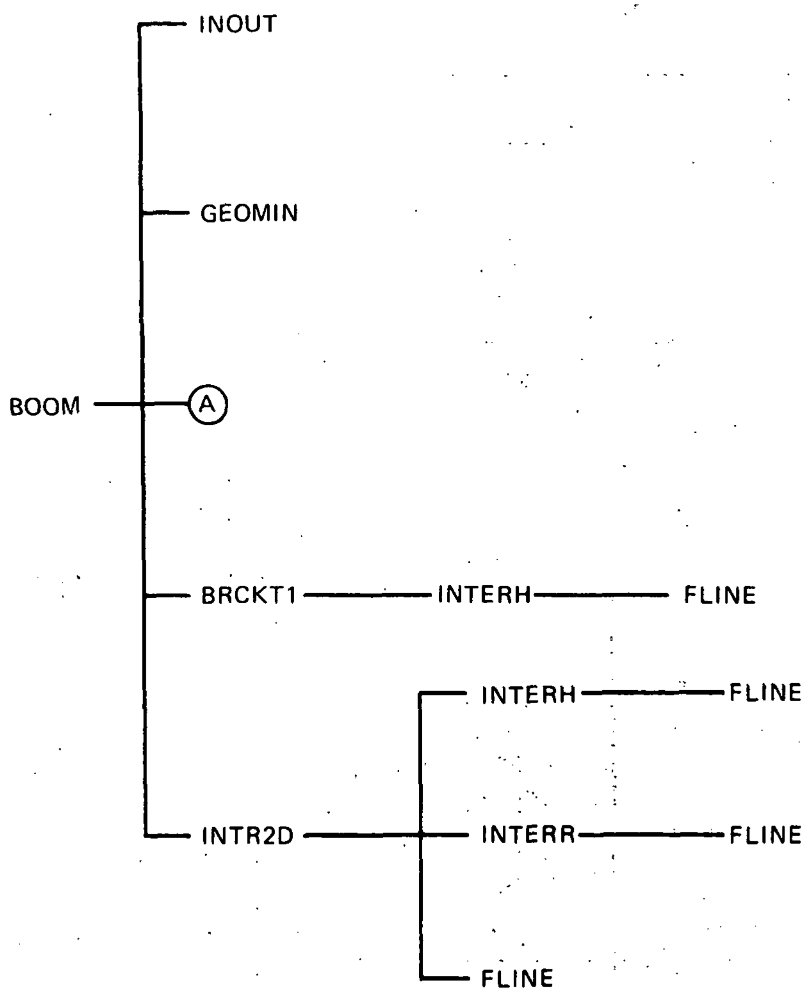


STRMBL TREE DIAGRAM



(A) ≡ AS DEFINED IN QUICK TREE DIAGRAM

BOOM TREE DIAGRAM



(A) ≡ AS DEFINED IN QUICK TREE DIAGRAM

REFERENCE

(1) G. Moretti and G. Bleich, Three-Dimensional Flow Around Blunt Bodies, AIAA J., 5, 1967.

APPENDIX A

A BRIEF CODE-ORIENTED USER'S GUIDE

FOR THE QUICK GEOMETRY SYSTEM

QUICK is a highly general geometry package based on library controlled mathematical modeling of cross sectional arcs and body lines. The mathematical models for the cross sections and the defining lines are taken together to provide a continuous analytic model of the surface geometry. Slopes, normals and all derivatives are therefore developed analytically. Of course, either discontinuous intersections or smooth fairings can be modeled and enforced in both the cross sections and the body lines.

QUICK generally works in two basic coordinate systems (x, y, z) and (x, r, θ); see Figure A1. Data for modeling is input in Cartesian coordinates, while interrogations for exercising the models are performed in Cylindrical coordinates. Both of the coordinate systems are further subject to a translation in z. This is due to the necessary presence of a mapaxis, located in the symmetry plane, usually corresponding to the position of maximum half-breadth (y_{max}); see Figure A2. The mapaxis is necessary to fulfill one of the basic constraints of the QUICK approach, which is: the radius (r) must be a single-valued function of the angle (θ). Figure A2 (b) obviously does not meet this constraint, while Figure A2 (c), with a properly defined mapaxis, does.

During the discussion of the use of QUICK, several terms will appear frequently, and as such, will be defined here:

(1) Cross section - standard definition; a planar cut through the vehicle normal to the FRL at a given x-station.

(2) Cross sectional model - mathematical abstraction of a cross section, using simple curves to represent arcs between specified control points.

(3) Control points - logically selected break or joining points between cross sectional arcs; initial and terminal points for defining each arc.

(4) Arc - a portion of one simple mathematical curve between two control points in cross section.

(5) Body lines - the defining lines of the vehicle geometry in plan and profile views; x-running control points given as $y_i = y_i(x)$ and/or $z_i = z_i(x)$.

(6) Body line model - mathematical abstraction of a body line, using simple curves to represent segments between specified match points.

(7) Match points - logically selected break or joining points between body line segments; initial and terminal points for defining each segment.

(8) Segment - a portion of one simple mathematical curve between two match points of a body line model.

(9) Component - same as an arc; usually considered to be a named portion of the vehicle geometry (e.g., a wing-upper-ellipse may be component WNGUPELL).

QUICK modeling is performed in terms of the basically independent logical cross section models and logical/mathematical body line models. The cross sections are defined purely in terms of the named component arcs and the named control points; see Figure A3 (a), which models the vehicle shown in Figure A2 (a). Body lines, corresponding to the named control points, are then defined mathematically for the length of the vehicle (or as long or short as is necessary); see Figure A3 (b). At a given x-station the body lines are interrogated to give values for the control points. These control point values are then used to create the required cross sectional arc models which are interrogated at a given value of θ.

In cross section, a component arc is defined in terms of its control points, its shape, and its type. The arcs are considered to be ordered counter-clockwise (looking up the x-axis, i.e., in the negative x direction) starting at the bottom of the vehicle ($\theta = -\pi/2$) and going to the top of the vehicle ($\theta = +\pi/2$); see Figure A3 (a). A full complement of these arcs will define a cross sectional model, which is then given a specific range, in x, over which the model is applicable. The only exception, or extension, to the ordering rule is used to allow intersections between cross sectional arcs to be computed internal to the code. Components which may be considered to start in the body and grow out (such as a canopy or wing; see Figure A3 (a)) make use of ARCNM, as defined later in Figure A4, to specify to the code the other arc sharing the intersection point. Such growing components are ordered as before except they appear after the last arc in the outer, basic skin. Fillets (see Table AII and Figure A4) are also ordered as before, but appear last as a group; i.e., all fillets follow both the basic skin and the growing adaptive pieces.

The arc shapes available in cross section, along with their key words and equations follow in Table AI.

TABLE AI - CROSS SECTION ARC SHAPES

SHAPE	KEYWORD	EQUATION
LINE	LINE	$Ay + Bz + C = 0$
ELLIPSE (Concave to Origin)	ELLI	$\frac{(y-y_o)^2}{A^2} + \frac{(z-z_o)^2}{B^2} = 1$
ELLIPSE (Convex to Origin)	ELLO	Same

The line is defined exclusively in terms of its end points (control points); the ellipses also require a slope control point.

The curve type controls the blending of the various arcs (or segments, since the cross sectional curves use the same group of curve types as the body lines). In cross section, fore and aft are determined from the component ordering as mentioned before. A list of the curve types available, their keywords, and their functions follow in Table AII.

TABLE AII - CROSS SECTION AND BODY LINE CURVE TYPES

(Blending Control)

TYPE	KEYWORD	FUNCTION
Piece	PIECE	Curve is defined as a unit, with end points and slope control point if necessary.
Aft-Link	ALINK	*Curve being defined begins at the end of the previous curve and is tangent to it.
Fore-Link	FLINK	*Curve being defined ends at the beginning of the following curve and is tangent to it.
Patch	PATCH	*Curve being defined begins at the end of the previous curve and ends at the beginning of the following curve and is tangent to both of the adjoining curves.
Fillet	FILET	End points and slopes of curve being defined are calculated from specified positions on the adjoining curves.
**Null	NULL	Deletes an already existing segment.

*In body line definition, "previous" and "following" are only relative, as the specific segments being linked or patched to are given as part of the data.

**Available only in the modeling of body lines.

Figure A4, which follows, gives a card-by-card description of the data input format for cross sectional modeling.

Consider, for an example, the simple forebody shown in Figure A5 (a). There are two basic cross sectional configurations corresponding to the initial purely conical section and the final section with flat sides.

One therefore selects the cross sections as shown in Figure A5 (b). The coding of the input data is shown in Figure A6. Note that in the first model both ellipses are PIECE's, while in the second model one ellipse is an FLINK and one is an ALINK. Also note the order in which the arcs are to be defined (JSEQ); for either of the ellipses to link to the line, the line must first exist. Of course, depending upon the definition of the two slope control points, either or both of the ellipses could have been PIECE's. In the current setup, note that in model two the slope control points establish a slope for the center line points only, the slopes of the tangent points being established by the line.

For a body line (a control point definition as a function of x), a segment is defined in terms of its match points, its shape, and its type, much the same as a cross sectional arc. The major difference between segment and arc definitions is that segment match points are numbers, establishing immediately the mathematical representation of the given curve, while, as shown before, arc control points are, at the input stage, logical definitions only. Body lines may also be aliased to other body lines, when duplicate definitions are desired. The segments are considered to be ordered in the increasing x-direction over a range of applicability established by the match points. Segments are input in the order in which they are to be defined and have an index to establish their x-direction ordering as opposed to the cross sectional arcs which are input in their order of appearance (bottom to top) and have an index to establish their order of definition. This will be better understood after looking at Figure A6 a little later and after having seen an example. A full complement of these segments (from one to the code's dimensional limits - these are presented later) will define a body line.

The segment shapes available are more numerous than are the arc shapes, and they follow, along with their key words and equations, in Table AIII.

TABLE AIII - BODY LINE SEGMENT SHAPES

SHAPE	KEYWORD	EQUATION
Line	LINE	$Ax+By = 0$
x-Parabola	XPAR	$Ax+By+y^2 = 0$
y-Parabola	YPAR	$Ax+By+x^2 = 0$
Rotated x-Parabola	RXPA	$Ax+By+Cxy+y^2 = 0$
Rotated y-Parabola	RYPA	$Ax+By+Cxy+x^2 = 0$
x-Ellipse	ELLX	$Ax+By+Cx^2+y^2 = 0$
y-Ellipse	ELLY	$Ax+By+Cy^2+x^2 = 0$
Cubic	CUBI(C)	$Ax+By+Cx^2+x^3 = 0$

The line is defined exclusively in terms of its endpoints; the x- and y-parabolas require, in addition, one slope to be specified and one to be left free; all other curves require two points, and two slopes (the slopes usually being established by means of a slope control point).

The curve type controls the blending of the various segments, as for the cross sectional arcs. The list of curve types available for body line segments, as well as arcs, along with their key words and functions, has already been tabulated in Table AII.

Following, in Figure A6, is a card-by-card description of the data input format for body line modeling. A given segment is defined from an initial point as (x_1, v_1) to a final point (x_2, v_2) with an initial slope, t_1, and a final slope, t_2. Where applicable, t_1 and t_2 are determined from a slope control point at (x_3, v_3). The letter "v" is used to represent y or z since either may currently be under definition. These cards follow the cross section data.

Consider, for example, the same simple forebody that was used to demonstrate cross sectional modeling; Figure A5 (a). Looking back to our cross sectional model, we see that we have defined a total of seven control points (BDYBCL, BDYLTN, BDYLSCP, BDYUTN, BDYTCL, BDYUSCP and MAPAXIS). Each of these must now have y and z defined as a function of x. (The mapaxis is constrained to the symmetry plane; i.e., y = 0.) Immediately following the cross section input data shown in Figure A7 one would input the body line data shown in Figure A8. Note that since tan (10°) = .176327, the definitions for YBDYLTN and YBDYUTN are equivalent, and therefore could have been aliased. Also note that in aliasing, only the model itself is important, and thus one may alias Z̲BDYTCL with Y̲BDYUTN. Observe that a negative reflection of a given body line requires a separate model.

After reading the previous sections, a general approach to modeling any given configuration should begin to be apparent. One must first look at the general shapes involved in the cross sections, and determine how many unique cross section models are necessary to completely define the vehicle. These cross sections must then be logically defined by choosing the appropriate control points and arcs as in Figure A3 (b) and Figure A5 (b), and by deciding upon each model's range of applicability, in x. The coding of the input data for these cross sections can then be commenced. After this, one must carefully go through and define y(x) and z(x) for each control point. This completely defines the vehicle geometry.

The code is currently dimensioned to allow 10 arcs per cross sectional model, 10 segments per body line model, 10 cross sectional models and 25 body line models. Of course, these may be adjusted if required.

To exercise the geometry model, there are several modes of interrogation built into QUICK. Following the blank card which terminates the body line modeling, one may insert a card of the format shown in Figure A9. A positive MODE produces printed output, a negative MODE produces a data file on unit IPLOT which may be used for plotting purposes. A blank card must follow these checkout requests to terminate the program.

In the main routine, there are five integer variables which control I/O operations. They are:

IREAD = read unit
IRITE = write unit
ICRITE = write unit for any error messages
ITAPE = write/read unit for intermediate data file
IPLOT = I/O unit for plotting data output from GEMCHK, MODE1, MODE2, etc.

In addition, a reference punch unit (IPUNCH) is set equal to seven (7) in a data statement. This variable is used simply to prevent improper I/O operations on the punch unit and is normally transparent to the user; however, if the punch unit is not seven (7), then IPUNCH must be redefined to the proper unit in QUICK and GEOMIN.

The intermediate data file is an interface between QUICK and SUB-QUICK. SUB-QUICK is a subset of QUICK's subroutines which may be used in conjunction with any other code. In exercising QUICK, the intermediate data file will be written on the unit corresponding to ITAPE. All necessary information is passed between the defining and checking subroutines and the interrogating subroutines of SUB-QUICK via common blocks when they are used together; however, the intermediate data deck is both necessary and sufficient for SUB-QUICK when exercising it alone. A list of the routines in QUICK/SUB-QUICK follows:

QUICK
DSETUP
DLOKUP
CSMDEF
CSMCHK
BLMDEF
BLMCHK
KRVDEF
GEMOUT
GEMCHK
MODE1
MODE2
MODE3

MODE4
MODE5
MODE7
SLOPE
GEOMIN
CSGEOM
BLMSET
CURVES
CSMSET
BLGEOM ——— SUB-QUICK
VDOTV
MDOTV
THELIM
CSCALC
CSMINT
CSMFLT

To make use of SUB-QUICK, one must call two subroutines, the first being GEOMIN to read in the intermediate data deck, the second being CSGEOM for each point of interest.

To read the data:

CALL GEOMIN (ITAPE, IRITE, ICRITE, IREAD)

Where: ITAPE = unit location of intermediate data deck for vehicle geometry

IRITE = write unit

ICRITE = write unit for any error messages

IREAD = read unit (not currently used in SUB-QUICK)

To interrogate at a point:

CALL CSGEOM (X, H, R, RX, RH, RXX, RXH, NDERV)

Where: X = x location

H = theta location ($-\pi/2 \leq \theta \leq +\pi/2$)

R = radial distance from mapaxis to point on body surface corresponding to X and H.

RX = dr/dx at this point

RH = $dr/d\theta$ at this point

RXX = d^2r/dx^2 at this point

RXH = $d^2r/dxd\theta$ at this point

NDERV = $\pm$ N, where N is the order of derivative to be calculated

+ N, previous call was to different location; must compute R and all temporary variables

- N, previous call was to same point, thus derivatives may be computed without recomputing R or certain temporary variables

The quantities X, H and NDERV are, of course, user specified, and the geometry code will return all other values.

Two additional and more complex geometry modeling examples are included in Appendix A-A for the potential user's reference.

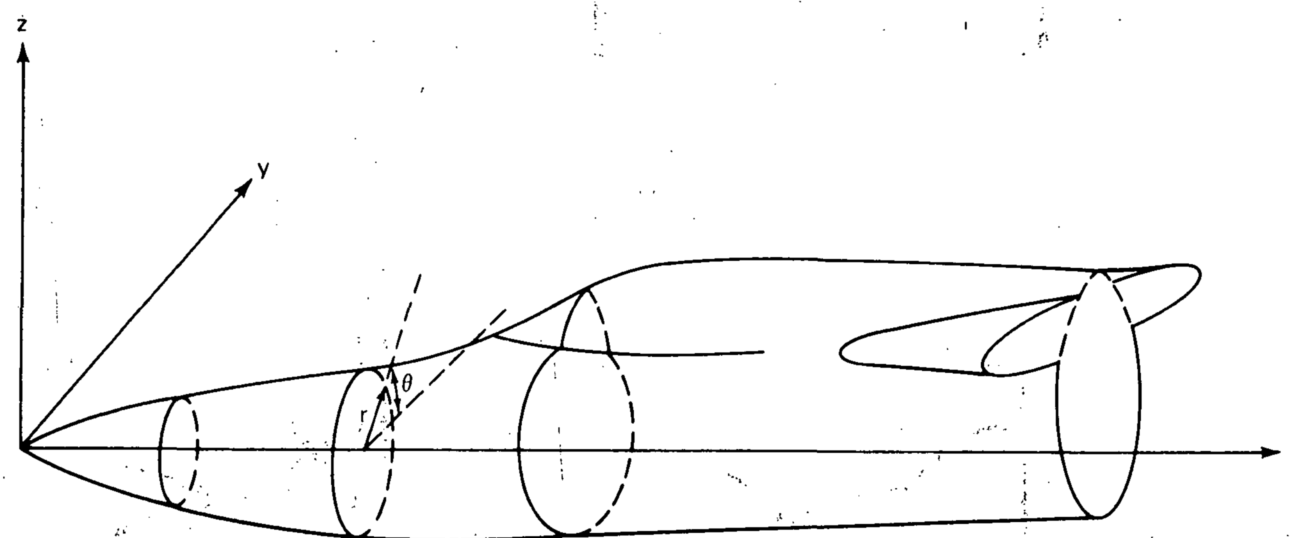


Figure A1 - DEFINITION OF COORDINATE SYSTEM

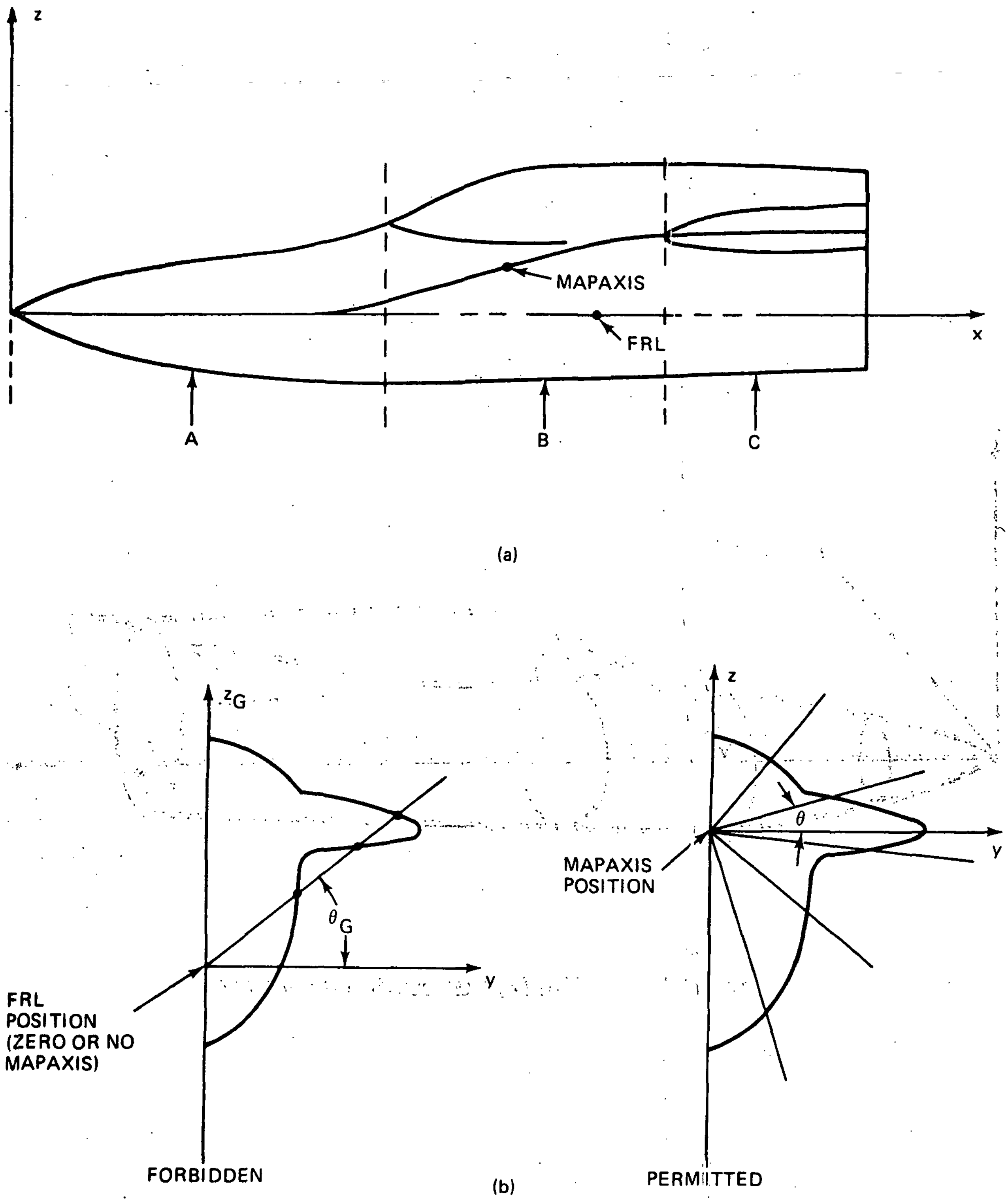


Figure A2 - AN EXAMPLE OF THE FUNCTION OF THE MAPAXIS

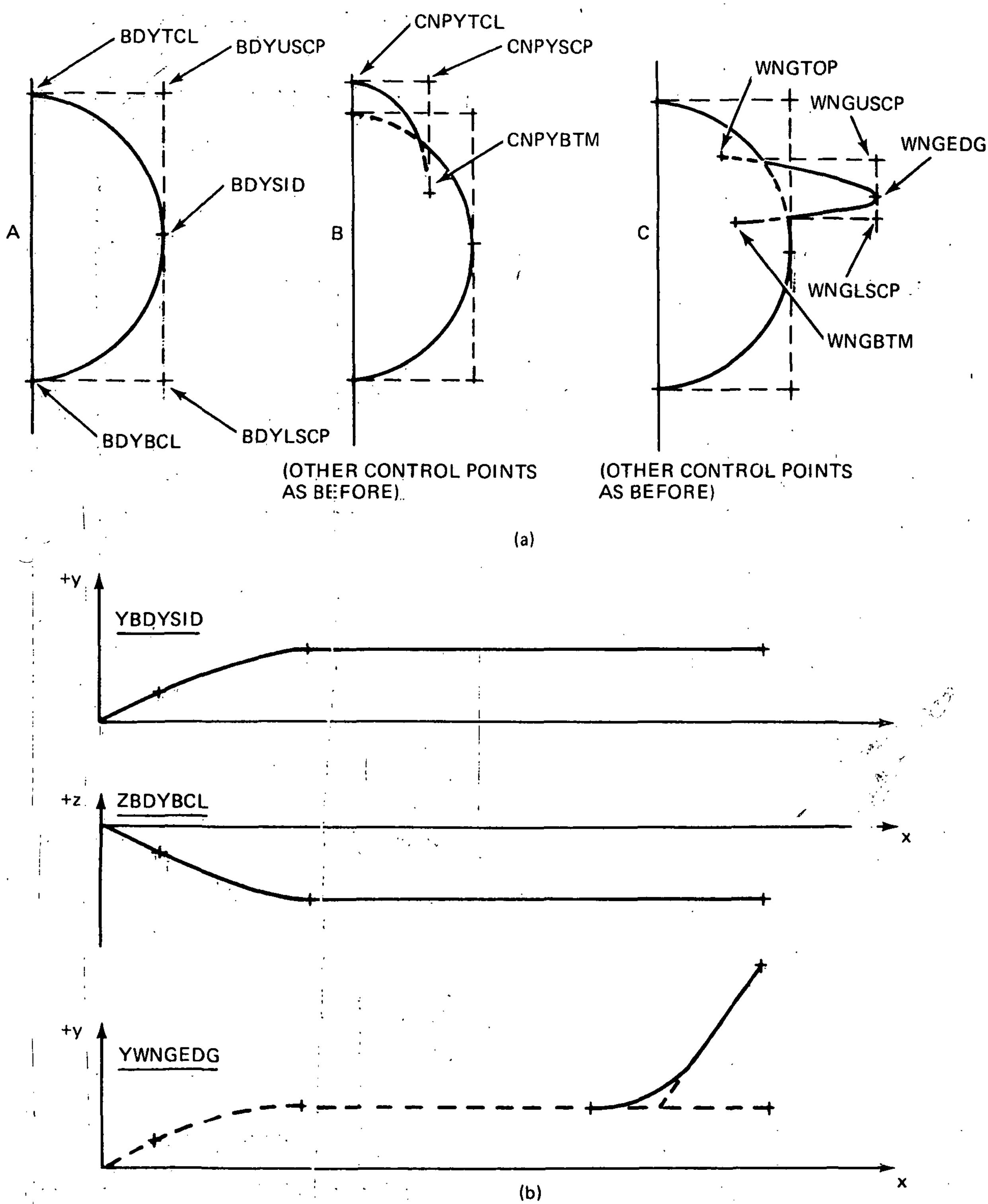


Figure A3 - EXAMPLE OF QUICK DIVISION INTO CROSS SECTIONS AND BODY LINES

	FORMAT	SYMBOL	DESCRIPTION
Card 1			
Col. 1-80	15A4	VTITLE	Vehicle or run title.
Card 2			
Col. 1-2	I2	NCSM	Number of distinct cross section models.
Card 3 (There will be exactly NCSM number of these cards appearing together with the appropriate cards of type 4.)			
Col. 1-2	I2	KDUM	Running count of the current cross section model (from 1 to NCSM).
Col. 3-4	I2	KARC	Number of arcs in current cross section model.
Col. 11-50	10A4	CTITLE	Title and/or descriptor of current cross section model.
Card 4 (There will be exactly KARC number of these cards per model; i.e. one for each arc, and they will be grouped together for a given model after a card of type 3.)			
Col. 1-8	A8	ARCNAM	Arc or component name.
Col. 9-10	I2	JSEQ	Definition sequence (order in which the arcs are to be defined).
Col. 11-14	A4	ASHAPE	Arc or component shape.
Col. 17-20	A4	ATYPE	Arc or component type.
Col. 25	A1	ASPEC(1)	= blank yields no effect. = Y when type is FILET, and only y - values are to be specified for that control point (z is computed on controlling component). = Z when type is FILET, and only z - values are to be specified for that control point (y is computed on controlling component). = B to indicate that this control point is the bottom center line of the vehicle for this model (optional). - T to indicate that this control point is the top center line of the vehicle for this model (optional).
Col. 26-33	A8	PNTNAM(1)	Control point name for the beginning of this arc.
Col. 35	A1	ASPEC(2)	Same as Col. 25, ASPEC (1).

Figure A4 - DATA INPUT FORMAT FOR CROSS SECTION MODELING

	FORMAT	SYMBOL	DESCRIPTION
Card 4 (Cont)			
Col. 36-43	A8	PNTNAM(2)	Control point name for the termination of this arc.
Col. 46-53	A8	PNTNAM(3)	Slope control point name for this arc when required, blank if not.
Col. 56-63	A8	ARCNM(1)	If type is FILET: the name of the most aft component arc to which the current arc's forward end is to be filleted.
			If type is other: the name of the most aft component arc which, in case of intersection with the current arc, is to update the forward end of the current arc and the aft end of the intersected arc.
Col. 66-73	A8	ARCNM(2)	If type is FILET: the name of the most forward component arc to which the current arc's aft end is to be filleted.
			If type is other: the name of the most forward component arc which, in case of intersection with the current arc, is to update the aft end of the current arc and the forward end of the intersected arc.
Card type 5 (Appears after NCSM blocks of one card type 3 and KARC cards of type 4.)			
Col. 1-2	I2	KNTCSM	Number of cross section models to define entire vehicle.
Col. 11-18	A8	ZMAPNM	Name of mapaxis.
Card type 6 (There will be exactly KNTCSM number of these cards.)			
Col. 1-2	I2	KDUM	Running count of the current cross section model (from 1 to KNTCSM).
Col. 3-4	I2	MODEL	Index corresponding to the already defined cross section models (between 1 and NCSM).
			NOTE: KNTCSM may be larger than NCSM if a given model is used more than once.
Col. 11-20	F10.5	XCSMS1	Starting x-station of the current cross section model.
Col. 21-30	F10.5	XCSMS2	Ending x-station of the current cross section model.

Figure A4 - DATA INPUT FORMAT FOR CROSS SECTION MODELING (Continued)

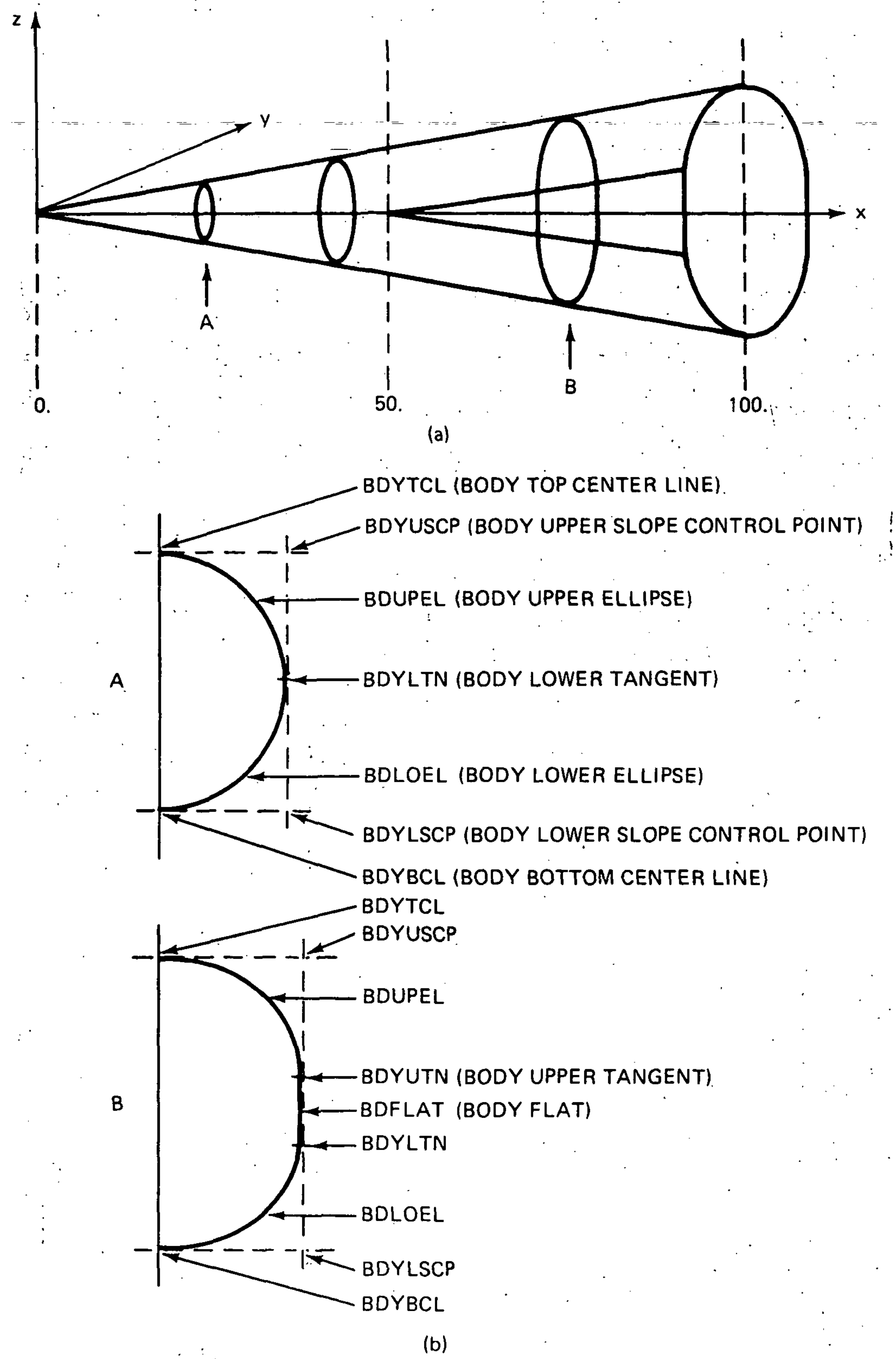


Figure A5 - SAMPLE FOREBODY WITH LOGICAL CROSS SECTION DEFINITION

Card 1	FORMAT	SYMBOL	DESCRIPTION

(Note: There will be as many Cards of type 1, followed by its Cards of type 2 and 3, as there are body line models, and as many cards of type 1, alone, as there are aliased control point coordinates, plus one blank card to terminate modeling input.)

Col. 1	A1	BYORZ	The letter Y or Z to indicate which data definition is to follow (a blank terminates all modeling input data).
Col. 2-9	A8	BNAME	Body Line/Control Point name which is to be defined.
Col. 11	A1	AYORZ	The letter Y or Z to indicate which definition is to be used when aliasing (blank when not).
Col. 12-19	A8	ANAME	Body Line/Control Point name to which BNAME is to be aliased, when applicable (blank when not).
Col. 31-70	10A4	TITLE	Any comments.

Card 2 (if not aliasing)

(Note: There will be as many Cards of type 2 and 3 as there are segments in a given body line, plus one Card type 2 with KSEG = -1.)

Col. 1-2	I2	KSEG	The order (in increasing x) in which this segment appears in this body line model. A KSEG = -1 (further arguments not required) terminates the data for a given body line (one Card 1).
Col. 4-7	A4	SSHAPE	Segment shape (including NULL, in which case this segment is essentially deleted, and no further parameters are required).
Col. 11-14	A4	STYPE	Segment type
Col. 17-18	A2	SDEF	Segment definition mode (currently, only two point, two slope/slope control point method is available - input "KV")
Col. 19	I1	IFREE	Index of the datum quantity which is to be "free", i.e., determined by the code. IFREE ranges from 1 to 6 corresponding to x_1, u_1, x_2, v_2, t_1, t_2, as ordered. A line must have any one of these free; an x- or y(v)- parabola must have either 5 or 6 free; other curves should have IFREE = 0.

Figure A6 - DATA INPUT FORMAT FOR BODY LINE MODELING (Sheet 1 of 2)

Card 3 (if not aliasing)
(see note for Card 2)

(Note: If SSHAPE is NULL, this card is deleted)

	FORMAT	SYMBOL	DESCRIPTION
Col. 1-10	F10.5	D (1)	If type is PIECE, FLINK, this is x_1. If type is ALINK, PATCH, or FILET, this is a floating point number equal to KSEG of the segment from which x_1 and/or v_1 are to be determined.
Col. 11-20	F10.5	D (2)	If type is PIECE or FLINK, this is v_1. If type is ALINK, PATCH or FILET, this is a floating point number equal to KSEG of the segment from which t_1 is to be determined.
Col. 21-30	F10.5	D (3)	If type is PIECE or ALINK, this is x_2. If type is FLINK, PATCH, or FILET, this is a floating point number equal to KSEG of the segment from which x_2 and/or v_2 are to be determined.
Col. 31-39	F9.4	D (4)	If type is PIECE or ALINK, this is v_2. If type is FLINK, PATCH, or FILET, this is a floating point number equal to KSEG of the segment from which t_2 is to be determined.
Col. 40	A1	SLP1	= blank yields no effect = S when following item, D (5), is to be explicit t_1. = A when following item, D (5), is to be arctan t_1 (in degrees).
Col. 41-49	F9.4	D (5)	If SLP1 is blank: If type is FILET, this is x_1 (v_1 and t_1 are to be determined from the segment specified by D (1) and D (2)) If type is other, this is x_3 If SLP1 is other than blank, see definition of SLP1, Col. 40.
Col. 50	A1	SLP2	= blank yields no effect = S when following item, D (6), is to be explicit t_2. = A when following item, D (6), is to be arctan t_2 (in degrees).
Col. 51-60	F10.5	D (6)	If SLP2 is blank: If type is FILET, this is x_2 (v_2 and t_2 are to be determined from the segment specified by D (3) and D (4)). If type is other, this is v_3. If SLP2 is other than blank, see definition of SLP2.

Figure A6 - DATA INPUT FORMAT FOR BODY LINE MODELING (Sheet 2 of 2)

FORTRAN CODING FORM

PROGRAM	PREPARED BY	GROUP (DEPT)	DATE	JOB NO.	PRI. NO.	SEC. NO.	PROGRAM NO.	PAGE
SAMPLE CROSS SECTION MODEL	L. S. YAEGER	423	4/23/74					1 OF 1

```
QUICK GEØMETRY FØR $IMPLE FØREBØDY
 2
 1 2      NØ$E TØ $TART ØF FLAT
BDLØEL   1ELLI  PIECE    BDYBCL    BDYLTN    BDYL$CP
BDUPEL   2ELLI  PIECE    BDYLTN    BDYTCL    BDYU$CP
 2 3      FLAT TØ END ØF FØREBØDY
BDLØEL   2ELLI  FLINK    BDYBCL    BDYLTN    BDYL$CP
BDFLAT   1LINE  PIECE    BDYLTN    BDYUTN
BDUPEL   3ELLI  ALINK    BDYUTN    BDYTCL    BDYU$CP
 2        MAPAXIS
 1 1      0.        50.
 2 2      50.       100.
```

CODING INSTRUCTIONS

1. ALPHABETICAL CHARACTERS ARE WRITTEN AS FOLLOWS A B C D E F G H I J K L M N Ø P Q R $ T U V W X Y Z
2. NUMERICAL CHARACTERS ARE WRITTEN AS FOLLOWS 1 2 3 4 5 6 7 8 9 0
3. CARDS WITH A C IN COLUMN 1 ARE NOT PROCESSED BY FORTRAN, AND COLUMNS 2-72 MAY BE USED FOR COMMENTS
4. COLUMNS 1-5 OF THE FIRST LINE OF A STATEMENT MAY CONTAIN A STATEMENT NUMBER THAT IS LESS THAN 32,768, TO IDENTIFY THE STATEMENT.
5. COLUMN 6 IS USED TO INDICATE THE CONTINUATION OF A LONG STATEMENT. USE NON-BLANK, NON-ZERO CHARACTERS IN THIS COLUMN TO INDICATE CONTINUATION CARDS. THE LIMIT OF CONTINUATION CARDS IS 9 FOR FORTRAN 11 AND 19 FOR FORTRAN IV
6. COLUMNS 73-80 ARE FOR IDENTIFICATION AND SEQUENCING.
7. START ALL STATEMENTS IN COLUMN 7 AND USE BLANKS ONLY WHERE NEEDED (IN HOLLERITH SECTIONS OF FORMAT STATEMENTS)

KEYPUNCH	VERIFIED

Figure A7

PROGRAM	PREPARED BY	GROUP (DEPT)	DATE	JOB NO.	PRI. NO.	SEC. NO.	PROGRAM NO.	PAGE
SAMPLE BODY LINE MODEL	L.S. YAEGER	423	4/23/74					1 OF 2

FORTRAN CODING FORM

```
YBDYBCL
 1 LINE   PIECE KV5
0.        0.        100.      0.
-1
ZBDYBCL
 1 LINE   PIECE KV5
0.        0.        100.      -17.6327
-1
YBDYLTN
 1 LINE   PIECE KV5
0.        0.        100.      17.6327
-1
ZBDYLTN
 1 LINE   PIECE KV5
0.        0.        60.       0.
 3 LINE   PIECE KV5
60.       0.        100.      -5.
 2 RYPA   FILET KV0
1.        1.        3.        3.        50.       70.
-1
YBDYL$CP  YBDYLTN
ZBDYL$CP  ZBDYBCL
YBDYUTN
 1 LINE   PIECE KV4
0.        0.        100.     A10.
-1
ZBDYUTN
 1 LINE   PIECE KV5
0.        0.        60.       0.
 3 LINE   PIECE KV5
```

CODING INSTRUCTIONS:

1. ALPHABETICAL CHARACTERS ARE WRITTEN AS FOLLOWS A B C D E F G H I J K L M N Ø P Q R $ T U V W X Y Z
2. NUMERICAL CHARACTERS ARE WRITTEN AS FOLLOWS 1 2 3 4 5 6 7 8 9 0
3. CARDS WITH A C IN COLUMN 1 ARE NOT PROCESSED BY FORTRAN, AND COLUMNS 2-72 MAY BE USED FOR COMMENTS.
4. COLUMNS 1-5 OF THE FIRST LINE OF A STATEMENT MAY CONTAIN A STATEMENT NUMBER THAT IS LESS THAN 32,768, TO IDENTIFY THE STATEMENT.
5. COLUMN 6 IS USED TO INDICATE THE CONTINUATION OF A LONG STATEMENT. USE NON-BLANK, NON-ZERO CHARACTERS IN THIS COLUMN TO INDICATE CONTINUATION CARDS. THE LIMIT OF CONTINUATION CARDS IS 9 FOR FORTRAN II AND 19 FOR FORTRAN-IV.
6. COLUMNS 73-80 ARE FOR IDENTIFICATION AND SEQUENCING.
7. START ALL STATEMENTS IN COLUMN 7 AND USE BLANKS ONLY WHERE NEEDED (IN HOLLERITH SECTIONS OF FORMAT STATEMENTS).

KEYPUNCH:	VERIFIED:

Figure A8

PROGRAM	PREPARED BY	GROUP (DEPT)	DATE	JOB NO.	PRI. NO.	SEC. NO.	PROGRAM NO.	PAGE
SAMPLE BODY LINE MODEL (CONT'D.)	L.S. YAEGER	423	4/23/74					2 OF 2

FORTRAN CODING FORM

```
60.       0.        100.      5.
 2 RYPA   FILET KVO
1.        1.        3.        3.        50.       70.
-1
YBDYTCL   YBDYBCL
ZBDYTCL   YBDYUTN
YBDYUSCP  YBDYUTN
ZBDYUSCP  ZBDYTCL
YMAPAXIS  YBDYBCL
ZMAPAXIS  YBDYBCL
bbb -- ← (blank card)
```

CODING INSTRUCTIONS

1. ALPHABETICAL CHARACTERS ARE WRITTEN AS FOLLOWS
   A B C D E F G H I J K L M N Ø P Q R $ T U V W X Y Z
2. NUMERICAL CHARACTERS ARE WRITTEN AS FOLLOWS
   1 2 3 4 5 6 7 8 9 0
3. CARDS WITH A C IN COLUMN 1 ARE NOT PROCESSED BY FORTRAN, AND COLUMNS 2-72 MAY BE USED FOR COMMENTS.
4. COLUMNS 1-5 OF THE FIRST LINE OF A STATEMENT MAY CONTAIN A STATEMENT NUMBER THAT IS LESS THAN 32,768, TO IDENTIFY THE STATEMENT.
5. COLUMN 6 IS USED TO INDICATE THE CONTINUATION OF A LONG STATEMENT. USE NON-BLANK, NON-ZERO CHARACTERS IN THIS COLUMN TO INDICATE CONTINUATION CARDS. THE LIMIT OF CONTINUATION CARDS IS 9 FOR FORTRAN II AND 19 FOR FORTRAN IV.
6. COLUMNS 73-80 ARE FOR IDENTIFICATION AND SEQUENCING.
7. START ALL STATEMENTS IN COLUMN 7 AND USE BLANKS ONLY WHERE NEEDED (IN HOLLERITH SECTIONS OF FORMAT STATEMENTS).

KEYPUNCH:	VERIFIED:

Figure A8 (Con't)

	FORMAT	SYMBOL	DESCRIPTION
Col. 1-2	I2	MODE	= 0 (or blank), terminates all input.
			= ± 1, creates body line traces.
			= ± 2, creates cross sectional cuts.
			= + 3, interrogates cross sections in neighborhood of control points.
			- 3, allows multiple body line traces to create plan and profile views.
			= + 4, comparison of analytic derivatives with numerically formed derivatives.
			= + 5, check of unit vectors normal to body surface.
			= + 6, exercises modes 1, 2, and 3 at the limits of each cross sectional model.
			- 6, exercises modes - 2 and - 7 at the limits of each cross sectional model.
			= - 7, (plotting mode only) creates cross sectional cuts, but includes all arcs in their entirety (including growing pieces still contained within the basic skin).
Col. 4-5	I2	NDERV	= ± N, where N is the order of derivative to be calculated (N=0, 1 or 2).
			+ N, should always be used for these interrogations (means each call to a given location is new, thus the radius and all temporary variables must be computed).
			- N, should not be used for these interrogations (requires previous call to same location (x and θ), radius and certain temporary variables are not recomputed).
Col. 11-20	F10.5	XGO	Initial x-station to be interrogated.
Col. 21-30	F10.5	XEND	Final x-station to be interrogated.
Col. 31-40	F10.5	XDEL	Increment size in x, to establish outputs stations between XGO and XEND.
Col. 41-50	F10.5	HGO	Initial value of theta (in degrees) to be interrogated; not required for modes 1, 3.
Col. 51-60	F10.5	HEND	Final value of theta (in degrees) to be interrogated; not required for modes 1, 3.
Col. 61-70	F10.5	HDEL	Increment size in degrees to establish interrogation points between HGO and HEND; not required for modes 1, 3.

Figure A9 - DATA INPUT FORMAT FOR EXERCISING THE GEOMETRIC MODEL

APPENDIX A-A

QUICK GEOMETRY MODELING PACKAGE

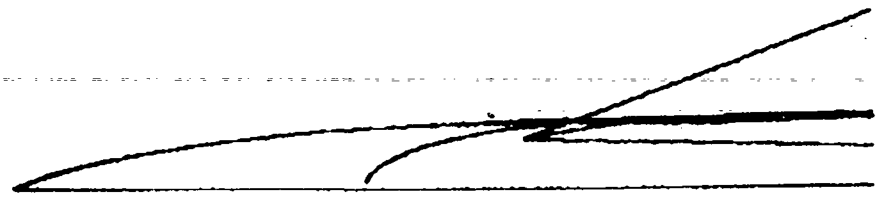
EXAMPLES

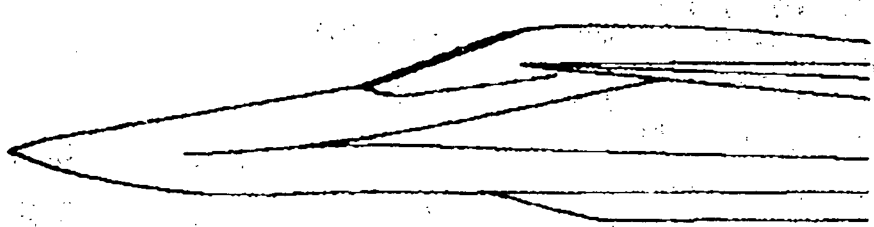
```
NCEF111A:  QUICK GEOMETRY FOR THE EF-111A (W/RADOME)
 5
 1 2       NOSE TO START OF FLATS
BDYLOELL 1ELLI  PIECE     BDBCL     BDLSDTN   BDLSCP
BDYUPELL 2ELLI  PIECE     BDLSDTN   BDTCL     BDUSCP
 2 4       FLATS TO START OF CANOPY
BDYLOFLT 1LINE  PIECE     BDBCL     BDBTMTN
BDYLOELL 3ELLI  PATCH     BDBTMTN   BDLSDTN
BDYSDFLT 2LINE  PIECE     BDLSDTN   BDUSDTN
BDYUPELL 4ELLI  ALINK     BDUSDTN   BDTCL     BDUSCP
 3 5       CANOPY TO START OF RADOME
BDYLOFLT 1LINE  PIECE     BDBCL     BDBTMTN
BDYLOELL 3ELLI  PATCH     BDBTMTN   BDLSDTN
BDYSDFLT 2LINE  PIECE     BDLSDTN   BDUSDTN
BDYUPELL 4ELLI  ALINK     BDUSDTN   BDTCL     BDUSCP
CANOPY   5ELLI  PIECE     CNBTM     CNTCL     CNSCP     BDYUPELL
 4 6       RADOME TO START OF WING
BDYLOFLT 1LINE  PIECE     BDBCL     BDBTMTN
BDYLOELL 3ELLI  PATCH     BDBTMTN   BDLSDTN
BDYSDFLT 2LINE  PIECE     BDLSDTN   BDUSDTN
BDYUPELL 4ELLI  ALINK     BDUSDTN   BDTCL     BDUSCP
RADOME   5ELLI  PIECE     RDMBCL    RDMEDG    RDMSCP              BDYLOFLT
CANOPY   6ELLI  PIECE     CNBTM     CNTCL     CNSCP     BDYUPELL
 5 8       WING TO INLET LIP
BDYLOFLT 1LINE  PIECE     BDBCL     BDBTMTN
BDYLOELL 3ELLI  PATCH     BDBTMTN   BDLSDTN
BDYSDFLT 2LINE  PIECE     BDLSDTN   BDUSDTN
BDYUPELL 4ELLI  ALINK     BDUSDTN   BDTCL     BDUSCP
BDYUPEL2 5ELLI  ALINK     BDUSDTN   BDTCL     BDUSCP
RADOME   6ELLI  PIECE     RDMBCL    RDMEDG    RDMSCP              BDYLOFLT
WNGLOELL 7ELLI  PIECE     WGBTM     WGEDG     WGLSCP    BDYUPELL
WNGUPELL 8ELLI  PIECE     WGEDG     WGTOP     WGUSCP              BDYUPEL2
 5         MAPAXIS
 1 1       0.        140.01
 2 2       140.01    180.
 3 3       180.      240.
 4 4       240.      274.
 5 5       274.      440.15
YBDBCL
 1 LINE    PIECE KU5
0.         0.        440.15    0.
-1
ZBDBCL
 1 LINE    PIECE KU4
0.         0.        12.                 A-18.5
 2 ELLX    ALINK KU0
1.         1.        120.      -20.75    50.       -20.75
 4 LINE    PIECE KU5
120.       -21.25    180.      -17.75
 3 ELLX    FILET KU0
2.         2.        4.        4.        120.      135.
 6 LINE    PIECE KU5
180.       -17.95    440.15    -14.97
 5 ELLX    FILET KU0
```

```
4.          4.          6.          6.          165.        180.
-1
ZBOLSDTN
 1 LINE     PIECE KV5
0.          0.          100.        0.
 3 LINE     PIECE KV5
100.        0.          140.        3.6
 2 ELLY     FILET KV0
1.          1.          3.          3.          90.         110.
 5 LINE     PIECE KV5
210.        6.75        440.15      2.
 4 ELLX     PATCH KV0
3.          3.          5.          5.
-1
YBOLSDTN
 1 LINE     PIECE KV4
0.          0.          12.                     A18.5
 2 ELLX     ALINK KV0
1.          1.          180.        30.16       160.        28.78
 3 RYPA     ALINK KV0
2.          2.          328.        32.8        300.        33.32
 4 RYPA     ALINK KV0
3.          3.          440.15      34.52       424.        34.
-1
YBOLSCP     YBOLSDTN
ZBOLSCP     ZBOBCL
YBOTCL      YBOBCL
ZBOTCL
 1 LINE     PIECE KV4
0.          0.          12.                     A18.5
 3 LINE     PIECE KV5
40.         10.15       180.        36.016
 2 ELLX     PATCH KV0
1.          1.          3.          3.
 5 LINE     PIECE KV5
184.        36.016      274.        70.
 4 RYPA     FILET KV0
3.          3.          5.          5.          180.        195.
 9 LINE     PIECE KV5
400.        65.12       440.15      61.64
 8 ELLX     FLINK KV0
316.236     69.5        9.          9.          400.        69.5
 7 ELLX     FLINK KV0
274.        68.16       8.          8.          300.        69.8
 6 ELLX     FILET KV0
5.          5.          7.          7.          260.        274.
-1
YBOUSCP     YBOLSDTN
ZBOUSCP     ZBOTCL
YBOBTMTN
 1 RYPA     PIECE KV0
140.        0.          170.        3.3         147.52      0.
 3 LINE     PIECE KV5
210.        7.15        440.15      18.14
```

```
 2 RYPA    PATCH KU0
1.         1.         3.         3.
-1
ZBDBTMTN   ZBDBCL
YBDUSDTN
 1 LINE    PIECE KU4
0.         0.         12.                   A18.5
 2 ELLX    ALINK KU0
1.         1.         180.       30.16      160.       28.78
 4 LINE    PIECE KU5
333.36     35.4       440.15     36.52
 3 RYPA    PATCH KU0
2.         2.         4.         4.
 1 NULL
-1
ZBDUSDTN
 1 LINE    PIECE KU5
100.       0.         140.       3.6
 3 LINE    PIECE KU5
144.       3.6        204.       14.0
 5 LINE    PIECE KU5
204.       14.0       333.36     41.74
 7 LINE    PIECE KU5
333.36     41.74      440.15     32.48
 2 ELLY    FILET KU0
1.         1.         3.         3.         140.       160.
 4 ELLX    FILET KU0
3.         3.         5.         5.         190.       210.
 6 RYPA    FILET KU0
5.         5.         7.         7.         328.       338.
 1 NULL
-1
YCNTCL     YBDTCL
ZCNTCL
 1 LINE    PIECE KU5
180.       36.016     230.       56.
 2 ELLX    ALINK KU0
1.         1.         274.       68.16      300.       69.8
-1
YCNBTM
 1 ELLX    PIECE KU0
180.       0.         280.       32.5       180.       25.7
-1
ZCNBTM
 1 ELLY    PIECE KU0
180.       36.016     200.       32.        190.       32.
 3 LINE    PIECE KU5
210.       33.        280.       43.
 2 ELLY    PATCH KU0
1.         1.         3.         3.
-1
YCNSCP
 1 ELLX    PIECE KU0
180.       0.         280.       20.        180.       16.
```

```
 1 NULL
-1
YRDMEDG
 1 LINE     PIECE KU5
240.        12.         440.15      12.
-1
ZRDMEDG
 1 LINE     PIECE KU5
180.        -5.95       440.15      -2.97
 3 LINE     PIECE KU5
245.52      -5.19       300.        -18.783
 5 LINE     PIECE KU5
180.        -17.9       433.        -15.
 2 RYPA     FILET KU0
1.          1.          3.          3.          240.        255.
 4 RYPA     FILET KU0
3.          3.          5.          5.          285.        305.
 6 RYPA     ALINK KU0
5.          5.          472.        -10.2       433.        -18.12
 1 NULL
-1
YRDMSCP     YRDMEDG
ZRDMSCP     ZRDMBCL
YMAPAXIS    YBDBCL
ZMAPAXIS
 1 LINE     PIECE KU5
0.          0.          80.         0.
 2 ELLY     PIECE KU0
80.         0.          180.        20.         140.        0.
 5 LINE     PIECE KU5
290.        47.96       440.15      43.28
 4 RYPA     FLINK KU0
260.        48.8        5.          5.          280.        48.8
 3 ELLY     PATCH KU0
2.          2.          4.          4.
-1
```





```
 FULSHTL:  QUICK GEOMETRY FOR SHUTTLE ORBITER
 7
 1 2        NOSE TO START OF BOTTOM FLAT
BDYLOELL 2ELLI  PIECE    BODYBCL   BDSDTUP   BDLOSCP
BDYUPELL 1ELLI  PIECE    BDSDTUP   BODYTOP   BDUPSCP
 2 3        BOTTOM FLAT TO START OF SIDE FLAT
FLATBTM  1LINE  PIECE    BODYBCL   BDBTMTN
BDYLOELL 3ELLI  PATCH    BDBTMTN   BDSDTUP
BDYUPELL 2ELLI  PIECE    BDSDTUP   BODYTOP   BDUPSCP
 3 4        FLATS TO START OF CANOPY
FLATBTM  1LINE  PIECE    BODYBCL   BDBTMTN
BDYLOELL 3ELLI  PATCH    BDBTMTN   BDSDTLO
FLATSIDE 2LINE  PIECE    BDSDTLO   BDSDTUP
BDYUPELL 4ELLI  PIECE    BDSDTUP   BODYTOP   BDUPSCP
 4 5        CANOPY TO START OF FAIRING
FLATBTM  1LINE  PIECE    BODYBCL   BDBTMTN
BDYLOELL 3ELLI  PATCH    BDBTMTN   BDSDTLO
FLATSIDE 2LINE  PIECE    BDSDTLO   BDSDTUP
BDYUPELL 4ELLI  PIECE    BDSDTUP   BODYTOP   BDUPSCP
CANOPY   5ELLI  PIECE    CNBDINT   CNPYTOP   CNPYSCP   BDYUPELL
 5 6        FAIRING TO START OF WING (AND END OF CANOPY)
FLATBTM  1LINE  PIECE    BODYBCL   BDBTMTN
BDYLOELL 3ELLI  PATCH    BDBTMTN   BDSDTLO
FLATSIDE 2LINE  PIECE    BDSDTLO   BDSDTUP
BDYUPELL 4ELLI  PIECE    BDSDTUP   BODYTOP   BDUPSCP
CANOPY   5ELLI  PIECE    CNBDINT   CNPYTOP   CNPYSCP   BDYUPELL
FAIRING  6ELLI  PIECE    FRBDINT   FRNGTOP   FRNGSCP   CANOPY
 6 6        WING TO START OF TAIL
FLATBTM  1LINE  PIECE    BODYBCL   BDBTMTN
WGBDLOEL 3ELLI  PATCH    BDBTMTN   WINGLE
WINGUPER 2ELLI  PIECE    WINGLE    BDSDTLO   WINGSCP
FLATSIDE 4LINE  PIECE    BDSDTLO   BDSDTUP
BDYUPELL 5ELLI  PIECE    BDSDTUP   BODYTOP   BDUPSCP
FAIRING  6ELLI  PIECE    FRBDINT   FRNGTOP   FRNGSCP   BDYUPELL
 7 7        TAIL TO END
FLATBTM  1LINE  PIECE    BODYBCL   BDBTMTN
WGBDLOEL 3ELLI  PATCH    BDBTMTN   WINGLE
WINGUPER 2ELLI  PIECE    WINGLE    BDSDTLO   WINGSCP
FLATSIDE 4LINE  PIECE    BDSDTLO   BDSDTUP
BDYUPELL 5ELLI  PIECE    BDSDTUP   BODYTOP   BDUPSCP
FAIRING  6ELLI  PIECE    FRBDINT   FRNGTOP   FRNGSCP   BDYUPELL
VERTTAIL 7ELLI  PIECE    TLFRINT   TAILTOP   TAILSCP   FAIRING
 7          MAPAXIS
 1 1        0.         20.
 2 2        20.        40.
 3 3        40.        180.01
 4 4        180.01     226.6
 5 5        226.6      360.
 6 6        360.       1077.8769
 7 7        1077.8769  1280.5206
VBODYBCL
 1 LINE     PIECE KV5
0.          0.         1280.5206 0.
-1
```

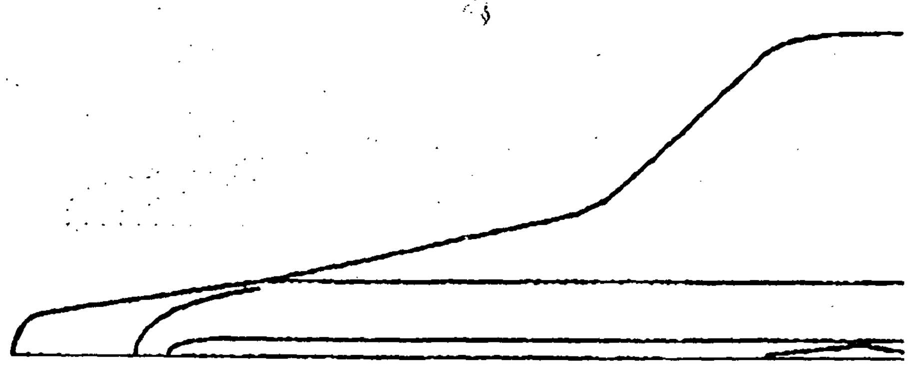
```
ZBODYBCL
 2 LINE    PIECE KV5
45.4119    -49.6848  929.7185  -74.1348
 1 ELLX    FLINK KV0
0.         0.        2.        2.        0.        -50.
 3 RYPA    ALINK KV0
2.         2.        1280.5206 -65.      1250.0    -67.0
-1
YBDSDTUP
 2 LINE    PIECE KV5
46.7026    59.       370.      108.
 1 ELLY    FLINK KV0
0.         0.        2.        2.        0.        10.
 4 LINE    PIECE KV5
370.       108.      1280.5206 108.
 3 ELLY    FILET KV0
2.         2.        4.        4.        360.      380.
-1
ZBDSDTUP
 1 LINE    PIECE KV5
0.         0.        46.7026   0.
 3 LINE    PIECE KV5
46.7026    0.        343.      58.
 2 ELLX    FILET KV0
1.         1.        3.        3.        40.       55.
 5 LINE    PIECE KV5
343.       58.       1107.     58.
 4 ELLX    FILET KV0
3.         3.        5.        5.        335.      360.
 7 LINE    PIECE KV5
1107.      58.       1328.3    66.25
 6 ELLX    FILET KV0
5.         5.        7.        7.        1100.     1120.
-1
YBDLOSCP   YBDSDTUP
ZBDLOSCP   ZBODYBCL
YBODYTOP   YBODYBCL
ZBODYTOP
 2 LINE    PIECE KV5
37.7036    53.6344   338.54    163.
 1 ELLX    FLINK KV0
0.         0.        2.        2.        0.        10.
 4 LINE    PIECE KV5
338.54     163.      1280.5206 163.
 3 ELLX    FILET KV0
2.         2.        4.        4.        330.      360.
-1
YBDUPSCP   YBDSDTUP
ZBDUPSCP   ZBODYTOP
YBDBTMTN
 1 LINE    PIECE KV5
0.         0.        30.       0.
 3 ELLX    PIECE KV0
30.        0.        360.      80.       120.      38.8
```

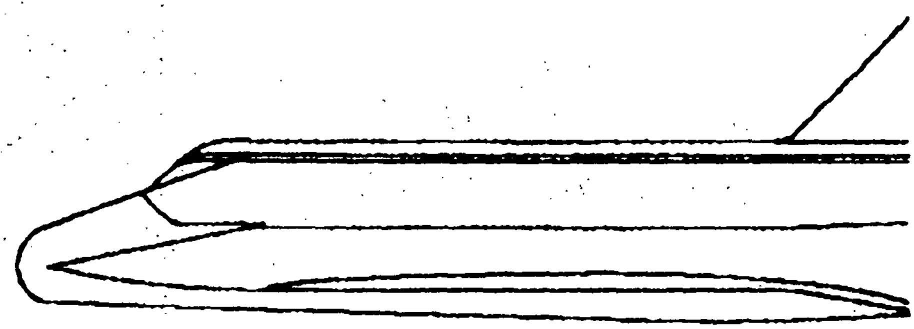
```
 2 ELLY     FILET KV0
1.          1.         3.         3.         20.        40.
 5 LINE     PIECE KV5
360.        80.        750.       60.
 4 ELLX     FILET KV0
3.          3.         5.         5.         350.       370.
 7 LINE     PIECE KV5
750.        60.        1100.      60.
 6 ELLY     FILET KV0
5.          5.         7.         7.         740.       760.
 9 LINE     PIECE KV5
1100.       60.        1280.5206  70.
 8 ELLY     FILET KV0
7.          7.         9.         9.         1080.      1120.
 1 NULL
-1
ZBDBTMTN    ZBODYBCL
YBDSDTLO    YBDSDTUP
ZBDSDTLO
 1 LINE     PIECE KV5
0.          0.         46.7026    0.
 3 ELLX     PIECE KV0
46.7026     0.         360.       -30.3826   180.       -30.3826
 2 ELLX     FILET KV0
1.          1.         3.         3.         40.        55.
 4 ELLX     PIECE KV0
360.        -30.3826   850.       -6.6       360.       -6.6
 5 ELLX     ALINK KV0
4.          4.         1280.5206  -50.       1300.      -57.
-1
YCNBDINT
 1 ELLX     PIECE KV0
180.        0.         360.       95.        180.       75.
-1
ZCNBDINT
 1 LINE     PIECE KV5
180.        105.365    226.6      63.
 3 LINE     PIECE KV5
226.6       63.        360.       63.
 2 ELLX     FILET KV0
1.          1.         3.         3.         220.       235.
-1
YCNPYTOP    YBODYBCL
ZCNPYTOP
 1 LINE     PIECE KV5
37.7036     53.6344    338.54     163.
 3 LINE     PIECE KV5
190.        109.       226.6      143.
 5 LINE     PIECE KV5
276.3705    163.       360.       163.
 2 ELLY     FILET KV0
1.          1.         3.         3.         180.       200.
 4 ELLX     PATCH KV0
3.          3.         5.         5.
```

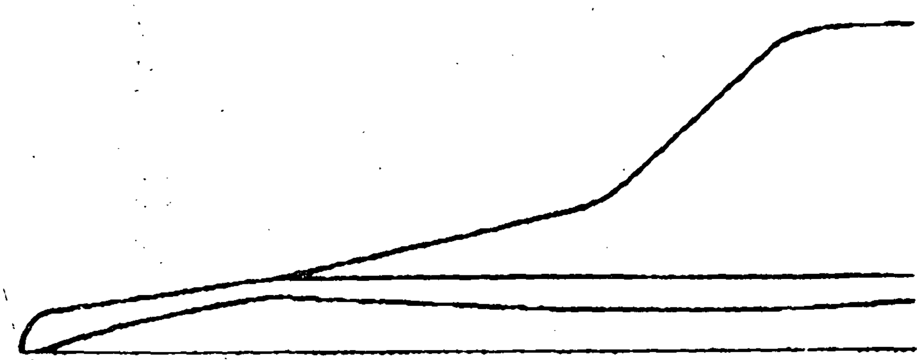
```
-1
ZCNSCP      ZCNTCL
YWGBTM
 2 LINE     PIECE KV5
333.36      34.         440.15      35.12
 1 RYPA     FLINK KV0
260.        25.         2.          2.          333.36      38.
-1
YWGTOP
 1 LINE     PIECE KV5
260.        25.         440.15      20.
-1
YWGEDG
 2 LINE     PIECE KV5
304.        40.2        440.15      89.2
 1 RYPA     FLINK KV0
260.        25.         2.          2.          285.        25.
-1
ZWGBTM
 2 LINE     PIECE KV5
333.36      41.74       440.15      32.48
 1 RXPA     FLINK KV0
260.        48.8        2.          2.          260.        40.
-1
ZWGTOP
 1 RXPA     PIECE KV0
260.        48.8        281.        49.38       260.        49.12
 2 ELLX     ALINK KV0
1.          1.          440.15      56.         400.        56.
-1
ZWGEDG
 2 LINE     PIECE KV5
290.        47.96       440.15      43.28
 1 RYPA     FLINK KV0
260.        48.8        2.          2.          280.        48.8
-1
YWGLSCP     YWGEDG
ZWGLSCP     ZWGBTM
YWGUSCP     YWGEDG
ZWGUSCP     ZWGTOP
YRDMBCL     YBDBCL
ZRDMBCL
 1 LINE     PIECE KV5
180.        -17.95      440.15      -14.97
 3 LINE     PIECE KV5
245.52      -17.19      300.        -30.783
 5 LINE     PIECE KV5
300.        -30.783     433.        -29.63
 2 RYPA     FILET KV0
1.          1.          3.          3.          240.        255.
 4 RYPA     FILET KV0
3.          3.          5.          5.          285.        305.
 6 RYPA     ALINK KV0
5.          5.          472.        -24.83      433.        -32.75
```

```
 1 NULL
-1
YCNPYSCP  YCNBDINT
ZCNPYSCP  ZCNPYTOP
YFRBDINT
 1 ELLX   PIECE KU0
226.6     0.        309.      24.359    226.6     24.359
 2 LINE   PIECE KU5
309.      24.359    1280.5206 24.359
-1
ZFRBDINT
 2 LINE   PIECE KU5
276.3705  155.      1280.5206 155.
 1 ELLY   FLINK KU0
226.      143.      2.        2.        190.      109.
-1
YFRNGTOP  YBODYBCL
ZFRNGTOP
 2 LINE   PIECE KU5
309.      183.      1280.5206 183.
 1 ELLY   FLINK KU0
226.6     143.      2.        2.        190.      109.
-1
YFRNGSCP
 1 ELLX   PIECE KU0
226.6     0.        309.      19.       226.6     19.
 2 LINE   PIECE KU5
309.      19.       1280.5206 19.
-1
ZFRNGSCP  ZFRNGTOP
YWINGLE
 1 LINE   PIECE KU5
46.7026   59.       370.      108.
 3 LINE   PIECE KU5
370.      108.      834.8     210.8
 5 LINE   PIECE KU5
834.8     210.8     1064.2    428.34
 7 LINE   PIECE KU5
1241.6    468.34    1280.5206 468.34
 6 RYPA   PATCH KU0
5.        5.        7.        7.
 4 ELLX   FILET KU0
3.        3.        5.        5.        790.      870.
 2 ELLY   FILET KU0
1.        1.        3.        3.        360.      380.
 1 NULL
-1
ZWINGLE
 1 LINE   PIECE KU5
360.      -30.3826  1105.     -30.3826
 3 LINE   PIECE KU5
1105.     -30.3826  1280.5206 -60.
 2 ELLY   FILET KU0
1.        1.        3.        3.        1090.     1120.
```

```
-1
YWINGSCP  YWINGLE
ZWINGSCP  ZBDSDTLO
YTLFRINT
 2 LINE   PIECE KV5
1082.     4.4       1215.4    18.4
 1 ELLX   FLINK KV0
1077.8769 0.        2.        2.        1077.8769 10.
 4 LINE   PIECE KV5
1215.4    18.4      1280.5206 9.2
 3 RYPA   FILET KV0
2.        2.        4.        4.        1210.4    1220.4
-1
ZTLFRINT
 1 LINE   PIECE KV5
1077.8769 163.      1280.5206 163.
-1
YTAILTOP  YBODYBCL
ZTAILTOP
 2 LINE   PIECE KV5
1107.     192.6     1366.8    453.
 1 ELLX   FLINK KV0
1077.8769 183.      2.        2.        1000.     183.
-1
YTAILSCP  YTLFRINT
ZTAILSCP  ZTAILTOP
YMAPAXIS  YBODYBCL
ZMAPAXIS
 1 LINE   PIECE KV5
0.        0.        300.      0.
 3 LINE   PIECE KV5
360.      -30.3826  1105.     -30.3826
 2 CUBIC  PATCH KV0
1.        1.        3.        3.
 5 LINE   PIECE KV5
1105.     -30.3826  1280.5206 -60.
 4 ELLY   FILET KV0
3.        3.        5.        5.        1090.     1120.
-1
```







APPENDIX B

A BRIEF USER'S GUIDE

TO THE

THREE-DIMENSIONAL BLUNT BODY CODE (BLUNT)

BLUNT is a simple to use code which will accept the QUICK intermediate data deck to define a blunt nose body and will supply a directly useable data deck for the starting plane of STEIN.

Here BLUNT's input data will be described. There are three input data cards for BLUNT in addition to the QUICK INTERMEDIATE DATA DECK. This intermediate data deck is output from QUICK and the user need not get involved in its details.

Input:

Card #1	NRUN, MONTH, MDAY, MYEAR, NA, MA, LA, KA, JA, LB, LE, IN, IGAS, IRESTRT
Card #2	ACH, GAMMA, STAB, THEMAX, ELL, XO, ANGLE, ALPHA
Card #3	PIN, TIN
Card #4	NCSU, MCSU, IPUNCH

QUICK INTERMEDIATE DATA DECK

Formats:

All quantities on Card #1 are read in I5 format
All quantities on Cards #2 and #3 are read in E10.4 format
All quantities on Card #4 are read in I5 format

Nomenclature:

NRUN	Run number
MONTH	Month
MDAY	Day
MYEAR	Year
NA	Number of intervals in the r direction (maximum of 10) (Fig. B1)
MA	Number of intervals in the θ direction (maximum of 10) (Fig. B1)

LA	Number of intervals in the ϕ direction (maximum of 8) (Fig. B1)
KA	The number of steps to be computed, after which the code will output initial data. Typically KA = 700 to reach steady state.
JA	The number of steps between outputs before the steady state.
LB	Indicator for output quantities indicating convergence at every step LB = 0 for no output.
LE	Geometry indicator: LE = 0 General geometry input (from "QUICK") LE $\neq$ 0 Circular cross sections, geometry is nondimensionalized with respect to the radius of curvature of the nose. LE = 1 Paraboloid cap LE = 2 Ellipsoid cap with a given axis ratio (ELL) and followed by a cone with half angle (ANGLE)
IN	Index not used
IGAS	Gas Indicator IGAS = 0 for perfect gas IGAS = 1 for air in equilibrium
IRESTRT	Restart indicator: = 0 Blunt body is started with code supplied guess and outputs data on unit 8 for restarting blunt body code. = 1 BLUNT reads starting data from unit 8 and continues.
ACH	Free stream Mach number
GAMMA	Ratio of specific heats (Cp/Cv) in free stream
STAB	Stability factor for C.F.L. condition (DT = DTmin(STAB)). Typically STAB = 1.2.
THEMAX	Limit on θ. Now computed in code but still in read statement.

ELL	Used only when LE = 2. Axis ratio of ellipsoid, ELL = 1. for spherically caped cone.
XO	Location of center of coordinate system (Fig. B1) (XO should be large enough so that initial data plane for supersonic flow calculation in supersonic)
ANGLE	Cone half angle for LE = 2
ALPHA	Angle of attack
PIN	Free stream pressure (p_∞/p_{SL}) use only when IGAS = 1.
TIN	Free stream temperature (T_∞/T_{SL}) used only when IGAS = 1.
NCSU	Number of mesh points in the initial data plane in the $\bar{r}$ direction (Fig. B2). NCSU can be different from NA + 1.
MCSU	Number of mesh points in the $\bar{\theta}$ direction (Fig. B2)
IPUNCH	Output unit for initial data plane results.

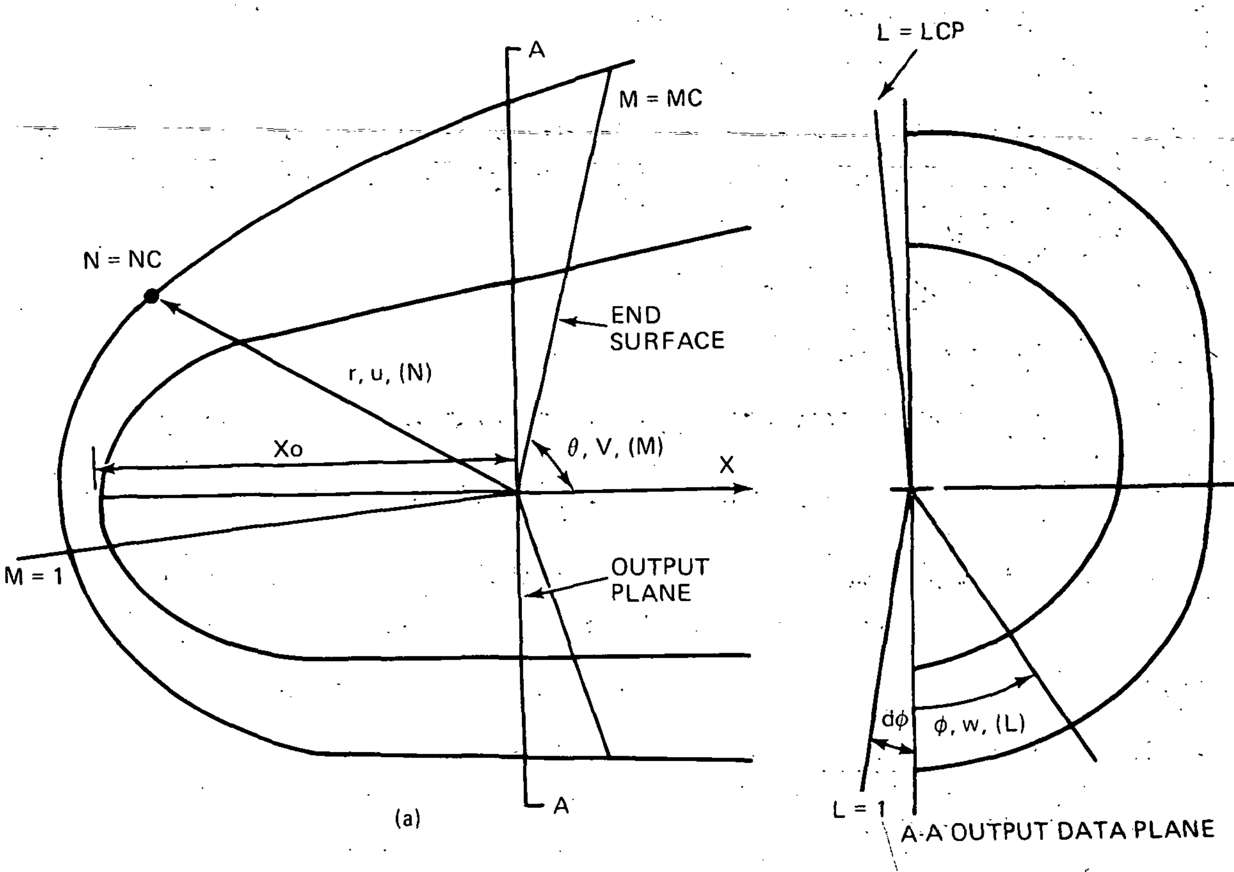


Figure B1 - COORDINATE SYSTEM

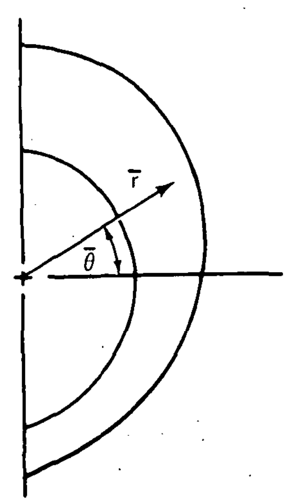


Figure B2 - INITIAL DATA PLANE

*U.S. GOVERNMENT PRINTING OFFICE: 1976 - 635-275/99