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
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 configurations are included to demonstrate the usefulness of this tool.
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

PART 1 USER-ORIENTED DOCUMENTATION

Description of Codes and Their Interaction ...................... 2
Nomenclature ..................................................................... 5
  QUICK ........................................................................ 5
  STEIN ........................................................................ 19
  STRMBL ...................................................................... 31
  BOOM .......................................................................... 36
Input Data Format ................................................................ 43
  QUICK ........................................................................ 43
  STEIN ........................................................................ 45
  STRMBL ...................................................................... 49
  BOOM .......................................................................... 50
Output Data Format ............................................................... 53
  QUICK ........................................................................ 53
  STEIN ........................................................................ 57
  STRMBL ...................................................................... 59
  BOOM .......................................................................... 59
Storage Requirements and Computer Time ......................... 76
  QUICK ........................................................................ 76
  STEIN ........................................................................ 77
  STRMBL ...................................................................... 78
  BOOM .......................................................................... 79

PART 2 PROGRAMMER-ORIENTED DOCUMENTATION

Overall Flow of Logic ....................................................... 80
  QUICK ........................................................................ 80
  STEIN ........................................................................ 80
  STRMBL ...................................................................... 80
  BOOM .......................................................................... 81
<table>
<thead>
<tr>
<th>Subroutine Descriptions</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUICK</td>
<td>85</td>
</tr>
<tr>
<td>STEIN</td>
<td>88</td>
</tr>
<tr>
<td>STRMBL</td>
<td>96</td>
</tr>
<tr>
<td>BOOM</td>
<td>99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subroutine Tree Diagrams (Call Sequence)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUICK</td>
<td>101</td>
</tr>
<tr>
<td>STEIN</td>
<td>103</td>
</tr>
<tr>
<td>STRMBL</td>
<td>106</td>
</tr>
<tr>
<td>BOOM</td>
<td>107</td>
</tr>
</tbody>
</table>

| REFERENCE                                       | 108  |

| APPENDIX A: A BRIEF CODE-ORIENTED USER'S GUIDE FOR THE QUICK-GEOMETRY SYSTEM | 109  |
| APPENDIX A-A: QUICK-GEOMETRY MODELING PACKAGE EXAMPLES                       | 132  |
| APPENDIX B: A BRIEF USER'S GUIDE TO THE THREE-DIMENSIONAL BLUNT-BODY CODE (BLUNT) | 145  |
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 aero-dynamic 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.
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
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 QUIC

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 \( \theta \) in degrees (used in various exercising routines; e.g., MODE1, MODE2, etc).

**HNOWR**  
Current value of \( \theta \) 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 Yl(I = 1), Zl(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 ith 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 Jth arc for growing pieces, or which is to supply filleting information if Jth 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

ICSASQ(J,K)

ICSATY(J,K)

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

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

I/0 unit for plot mode output from GEMCHK, MODE1, MODE2, etc.

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
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;

= +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 \( \theta \) at the initial control point location for each arc (at the current x-station)

THETA2(j)  Value of \( \theta \) 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

VTTITLE(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 (d²v/dx²) 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),  θ_o(I = 2),  A²(I = 3),  B²(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)  d²Y1(J)/dx²

Y2(J)  y of final point for each CS arc
Y2X(J) \quad dY2(J)/dx
Y2XX(J) \quad d^2Y2(J)/dx^2
Y3(J) \quad y of slope control point for forward (initial) end of each CS arc
Y3X(J) \quad dY3(J)/dx
Y3XX(J) \quad d^2Y3(J)/dx^2
Y4(J) \quad y of slope control point for aft (final) end of each CS arc
Y4X(J) \quad dY4(J)/dx
Y4XX(J) \quad d^2Y4(J)/dx^2

ZCL(I) \quad I = 1 to 3; current value \( z \) of bottom center line \((I = 1)\), top center line \((I = 2)\), and mapaxis \((I = 3)\)
ZCLX(I) \quad I = 1 to 3; current slope \(dz/dx\) of bottom center line \((I = 1)\), top centerline \((I = 2)\), and mapaxis \((I = 3)\)
ZCLXX(I) \quad 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 \quad Name of mapaxis
Z1(J) \quad z of initial point for each CS arc
Z1X(J) \quad dZ1(J)/dx
Z1XX(J) \quad d^2Z1(J)/dx^2
Z2(J) \quad z of final point for each CS arc
Z2X(J) \quad dZ2(J)/dx
Z2XX(J) \quad d^2Z2(J)/dx^2
Z3(J) \quad z of slope control point for forward (initial) end of each CS arc
Z3X(J) \quad dZ3(J)/dx
Z3XX(J) \quad d^2Z3(J)/dx^2
$z_4(J) \quad z \text{ of slope control point for aft (final) end of each CS arc}$

$z_4x(J) \quad \frac{dz_4(J)}{dx}$

$z_4xx(J) \quad \frac{d^2z_4(J)}{dx^2}$
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

BHM(M) Second derivative of body radius with respect to $\theta$

BHZ(M) Cross derivative of body radius with respect to $\phi$ and $\theta$

BN(M) Radial position of body in mapped plane at $Z + DZ$

BZ(M) Second derivative of body radius with respect to $\phi$

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) Radial position of shock $L$ in the mapped plane and its derivatives with respect to $\theta$ and $\phi$ (mapped coordinates)
Radial position of L\textsuperscript{th} wing type shock surface \((CC(M,L)) = B(M)\) and \(CC(M,L) = C(M,L-1)\) (\(L = 2 \ldots LC +1\)) and its
derivatives with respect to \(Y\) and \(Z\)

I = 1 to 5; alphanumeric request for aerodynamic coefficients (i.e., CL, CD, CM, CN, and CA).

I = 1 to \(KCOMP\), CMPTTL \((KCOMP + 1) = TOTL\) (total); alphanumeric title for each component (above)

Radial position of shock \(L\) in the mapped plane at \(Z + DZ\) and its derivatives with respect to \(\theta\) and \(\zeta\) (mapped coordinate at \(Z + DZ\))

Cone half angle (input in degrees); only used for sharp cone calculations

Mesh spacing in the radial direction, in region \(L\).

Mesh spacing in the circumferential direction, in region \(I\).

Step size

Factor multiplying \(DZ\) computed from CFL stability condition (usually \(DZFAC = .7\))

Interval for geometry test

Interval for printed output

\(J\)\textsuperscript{th} error generated in an iteration

Equivalent ratio of specific heats \((\gamma)\) for frozen flow

Free stream ratio of specific heats \((\gamma)\)

Local value of \(\frac{a^2}{(p/\rho)}\)

\((a = \text{speed of sound}, \ p = \text{pressure}, \ \rho = \text{density})\)

Mapped space polar angle

The \(X\) and \(Z\) derivatives of the \(I\)\textsuperscript{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 KC0MP, \( J = 1 \) to KPIECE(I); final value of \( \theta' \)
          defining each component and piece at previous station
HHL(M)   Value of \( \theta \) in the mapped plane of entropy layer surface
          point M (Fig. 3)
HIN(I,J)  Same as HI0(I,J) but at current station (see Fig. 8)
HI0(I,J)  \( I = 1 \) to KC0MP, \( J = 1 \) to KPIECE(I); initial value of \( \theta' \)
          defining each component and piece at previous station
HO(M)    Cylindrical \( \theta' \) at mesh points on the body at \( Z \) (see Fig. 8)
HS(N,I),  Circumferential position of cross flow surface \( I \) and its
HSR(N,I),  derivatives with respect to \( r \) and \( \varphi \) at \( Z \)
HSZ(N,I)  Circumferential position of cross flow surface \( I \) in the
          mapped plane and its derivative with respect to \( r \) and \( \varphi \)
          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.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IENTE</td>
<td>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.</td>
</tr>
<tr>
<td>IFCP(I,J)</td>
<td>I = 1 to KCOMP, J = 1 to KPIECE(I); final control point (in θ) for each component and piece (determined from QUICK modeling)</td>
</tr>
<tr>
<td>IGAS</td>
<td>Indicator; IGAS = 0: ideal gas; IGAS = 1: equilibrium; is set equal to 2 at Z = ZREEZ (freezing station)</td>
</tr>
<tr>
<td>IHS</td>
<td>Indicator; IHS = 0: metric factor h₁ not computed. IHS = 1: h₁ initial plane data read and computed. IHS = -1: h₁ initial plane data not read, but initialized by code to the body radius at each mesh point and computed.</td>
</tr>
<tr>
<td>III</td>
<td>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.</td>
</tr>
<tr>
<td>INCP(I,J)</td>
<td>I = 1 to KCOMP, J = 1 to KPIECE(I); initial control point (in θ) for each component and piece (determined from QUICK modeling)</td>
</tr>
<tr>
<td>IPUNCH</td>
<td>Output unit for starting plane data for next run</td>
</tr>
<tr>
<td>IREADO</td>
<td>Set to 5 in data statement in INIT-read unit for read #1</td>
</tr>
<tr>
<td>IREAD1</td>
<td>Read unit for control data 1</td>
</tr>
<tr>
<td>IREAD2</td>
<td>Read unit for control data 2</td>
</tr>
<tr>
<td>IREAD3</td>
<td>Read unit for starting plane data</td>
</tr>
<tr>
<td>IREAD4</td>
<td>Read unit for QUICK intermediate data</td>
</tr>
</tbody>
</table>
Indicators: \( J = 1 \) denotes the bottom symmetry plane, \( J = 2 \) wing plane, and \( J = 3 \) the top symmetry plane.

- \( \text{ISHBEG}(J) = 0 \) no sharp leading edge at the \( J^{\text{th}} \) plane.
- \( \text{ISHBEG}(J) = 1 \) there is a sharp leading edge at the \( J^{\text{th}} \) plane but the shock has not been detected yet. \( \text{ISHBEG}(J) \) is set equal to 2 when the shock has been detected.
- \( \text{ISHBEG}(J) \) is set equal to 3 when the shock is in.

Wing type shocks surface indication for shock \( L \) at \( M \):

- \( \text{ISHOK}(M,L) = 0 \): arbitrary surface
- \( \text{ISHOK}(M,L) = 1 \): shock point (detached)
- \( \text{ISHOK}(M,L) = 2 \): sharp leading edge shock point

Indicator: \( \text{ISHTIP} = 0 \) no sharp leading edges;
\( \text{ISHTIP} \neq 0 \) sharp leading edges exist on the geometry.

Output unit for printed flow field data

- \( \text{IZ}(I,J) = 0 \) to \( \text{KCOMP} \), \( J = 1 \) to \( \text{KPIECE}(I) \); Indicator:
  - \( = 0 \): Component piece is not present between this \( Z \) and \( Z + \text{DZ} \)
  - \( = 2 \): Component piece is present between this \( Z \) and \( Z + \text{DZ} \) and, thus, must be integrated over

Maximum number of steps between printed output

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

Maximum number of steps before punching output and stopping run

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

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

\( I = 1 \), \( \text{KCOMP} \); 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(lC) + MREG(lC) is the top symmetry plane

MC(l),  Correspond to NC(l), MSHK1(l), MSHK2(l). NREG(l) but for cross flow type surfaces

NSHK1(l),

NSHK2(l),

MREG(l)

MCIR  Minimum number of points in the "M" direction in any region I (usually MCIR = 5)

MCl  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),  Values of M at end shock points of shock L (Fig. 4)

MSHK2(l)

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 ≤ 10)

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

The three $\text{UNOR}(1,J)$, $\text{UNOR}(2,J)$ and $\text{UNOR}(3,J)$ Cartesian components of the unit normal to the body at the $J^{th}$ sharp leading edge (Fig. 6)

Free stream velocity

$I = 2, 3$; $y$ and $z$ positions of line about which moments are computed

Computational plane coordinate ($X(1,L) = 0$ and $X(NC(L),L) = 1$) (Fig. 2)

Cartesian $x'$ at mesh points on the body at $Z$ (see Fig. 7)

$x$ position (in the physical plane) of the wing tip and its derivative with respect to $z$ (Fig. 5)

Computational plane coordinate ($Y(1, I) = 0$ and $Y(MC(I),I) = 1$) (Fig. 2)

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)

Cartesian $y'$ at mesh points on the body at $Z$ (see Fig. 7)

Axial station

$z$ station immediately prior to start of sharp leading edge

Last axial station to be computed before punching output and stopping
ZFINL(I,J)  I = 1 to KCOMP, 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 < ZMAP1 and they are fully developed for Z ≥ 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 (∼ ZCOMP)

ZSOUT(I)  Specific values at z at which there is to be printed output I = 1 → NSOUT (if NSOUT ≤ 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 ≤ 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 ≥ ZSTART usually)

ZWRIT2 Last axial station at which output is printed (ZWRIT2 ≤ 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 \( 1 \times 10^6 \) and \( Z2NSH(J) \) is set to \(-1 \times 10^6\) 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, \( \Delta z \)
DZO  previous step size
FNU  Nondimensional kinematic viscosity
\[ FNU = \frac{\nu}{\nu_{\text{ref}}} \]  where \((\cdot) = \text{dimensional quantity and} \)
\[ \nu_{\text{ref}} = \sqrt{\frac{\gamma}{\gamma} \frac{P_\infty}{\rho_\infty}} \]
GAMMA  Free stream ratio of specific heats, read from tape
HCUT(IS,ICUT)  \( \theta' \) - location of each streamline at each cut for body normals
HP(N,M)  Angle from \( x' \) -axis (see Fig. 8) to mesh points (\( \theta' \) 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

31
**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(I) = O*

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) = O*

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(p/p_\infty) at mesh points (where \( \tilde{p} \) is pressure)

PI \pi

*As in STEIN
PNORM(INPT, IS, IICUT). \( \tilde{p}/\rho_\infty \) (where \( \tilde{p} \) 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 \( \ln(\tilde{p}/\rho_\infty) \) for the current streamline and data plane (where \( \tilde{p} \) 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) \( \theta \) 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) \( \theta' \) for each streamline.

U(N,M) \( x \)-velocity component at mesh points.

UNORM(INPT, IS, IICUT) \( x \)-component of velocity 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 B2 line (J = 3)
YCLZ(J) \( \frac{dYCL(J)}{dz} \)
YCLZZ(J) \( \frac{d^2YCL(J)}{dz^2} \)
Z Current z
ZCUT(IICUT) 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
Number of circumferential points on the data cylinder at which values of the flow variables are to be determined

\[ N_{REG}(L) = N_{C}(L-l) + N_{REG}(L-l), \quad N_{REG}(1) = 0^* \]

\[ \ln(p/p_\infty) \] at mesh points (where \( p \) is pressure)

Radius of data cylinder

Radial distance from mapaxis (\( B_2 \) line) to mesh points (\( r' \) in Fig. 8)

entropy at mesh points

Angle of attack, in degrees, read from cards

Free stream ratio of specific heats, read from cards

Free stream Mach number, read from cards

Initial value of \( z \), read from cards

\( x \)-velocity component at mesh points

\( y \)-velocity component at mesh points

\( z \)-velocity component at mesh points

\( J = 1 \) to \( 3 \); \( y \)-position of body bottom center line (\( J = 1 \)), body top centerline (\( J = 2 \)), mapaxis or \( B_2 \) line (\( J = 3 \))

\[ \frac{dYCL(J)}{dz} \]

\[ \frac{d^2YCL(J)}{dz^2} \]

Current \( z \)

Initial value of \( z \), read from tape

*As in STEIN*
Figure 4 - CROSS SECTIONS IN THE PHYSICAL AND COMPUTATIONAL SPACES
Figure 5 - MAPPING PARAMETERS

Figure 6 - SHARP LEADING EDGES
NC - 89B $M_{\infty} = 26.1, \alpha = 30^\circ$
$\gamma = 1.12$

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

<table>
<thead>
<tr>
<th>Card Type</th>
<th>Format</th>
<th>Variable Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15A4</td>
<td>VTITLE(I) (I = 1, 15)</td>
</tr>
<tr>
<td>2</td>
<td>I2</td>
<td>NCSM</td>
</tr>
<tr>
<td>3</td>
<td>2I2,6X,10A4</td>
<td>KDUM, KARC, CTITLE(I) (I = 1, 10)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Card Type</th>
<th>Format</th>
<th>Variable Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>A8,I2,A4,2X, A4,4X,A1,A8, 1X,A1,4A8</td>
<td>ARCNAM, JSEQ, ASHAPE, ATYPE, ASPEC(1), PNTNAM(1), ASPEC(2), PNTNAM(2), PNTNAM(3), ARCNM(1), ARCNM(2)</td>
</tr>
</tbody>
</table>

(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.)

<table>
<thead>
<tr>
<th>Card Type</th>
<th>Format</th>
<th>Variable Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>I2,8X,A8</td>
<td>KNTCSM, ZMAPNM</td>
</tr>
</tbody>
</table>
Card Type | Format | Variable Names
---|---|---
6 | 2I2,6X, 2F10.5 | KDUM, MODEL, XCSM1(KDUM), XCSM2(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.)

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.)

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° 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

<table>
<thead>
<tr>
<th>Card No.</th>
<th>Format</th>
<th>Variable Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16I5</td>
<td>IREAD1, IREAD2, IREAD3, IREAD4, IWRIT, IPUNCH, ICASE, IBUG, MCIR, NRUN, KA, JA, NLOOK, NSOUT, IBLOUT, IAERO</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Card No.</th>
<th>Format</th>
<th>Variable Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5F10.5</td>
<td>ZEND, ZWRIT1, ZWRIT2, DZWRIT, DZFAC</td>
</tr>
<tr>
<td>3</td>
<td>6F10.5</td>
<td>ZGEOM1, ZGEOM2, DZGEOM, ZWING, ZTIPS, ZFREEZ</td>
</tr>
<tr>
<td>4</td>
<td>2(F10.5, I5)</td>
<td>ZNADD, NDEL, ZMADD, MDEL</td>
</tr>
<tr>
<td>5 &amp; 5-a</td>
<td>8F10.5</td>
<td>ZSOUT(I) (I = 1, NSOUT) (if NSOUT &lt; 0 these cards are not read)</td>
</tr>
</tbody>
</table>
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 IFUNCH) 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:

<table>
<thead>
<tr>
<th>Card No.</th>
<th>Format</th>
<th>Variable Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5E15.5</td>
<td>Z1NSH(I) (I = 1, 5)</td>
</tr>
<tr>
<td>7</td>
<td>5E15.5</td>
<td>Z2NSH(I) (I = 1, 5)</td>
</tr>
<tr>
<td>8</td>
<td>5E15.5</td>
<td>Z1MSH(I) (I = 1, 5)</td>
</tr>
<tr>
<td>9</td>
<td>5E15.5</td>
<td>Z2MSH(I) (I = 1, 5)</td>
</tr>
<tr>
<td>10</td>
<td>2E15.5</td>
<td>ZMAP1, ZMAP2</td>
</tr>
<tr>
<td>11</td>
<td>7I5</td>
<td>IENTE, IGAS, ISHTIP, ISHBEG(I) (I = 1, 3)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Format</th>
<th>Variable Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>8X,11,1X, 5(A2, 3X)</td>
<td>IAERD, CFTITL(I) (I = 1, 5)</td>
</tr>
<tr>
<td>5E15.6</td>
<td>VMO(2), VMO(3), APINF, ARINF, AREF</td>
</tr>
<tr>
<td>12</td>
<td>KCOMP</td>
</tr>
<tr>
<td>12,3X,A4</td>
<td>NP, CMPTTL(I) (Note: NP is stored in KPIECE(I))</td>
</tr>
<tr>
<td>12,1X,12, 2F10.4</td>
<td>INCP(I,J), IFCP(I,J), ZINIT(I,J)</td>
</tr>
<tr>
<td></td>
<td>ZFINL(I,J) (I = 1, KCOMP; J = 1, NP = KPIECE(I))</td>
</tr>
</tbody>
</table>
The following data are read if and only if IAERD ≠ 0 (set and used by code for continuation runs).

<table>
<thead>
<tr>
<th>Format</th>
<th>Variables Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>6E13.7</td>
<td>PFT(I,J,K), PMT(I,J,K), AR(I,J) (I = 1, KCOMP; J = 1, NP = KPIECE(I); K = 1, 3)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Card No.</th>
<th>Format</th>
<th>Variable Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>4I5</td>
<td>LC, IC, NCL, MCL</td>
</tr>
<tr>
<td>13 &amp; 14</td>
<td>5E15.5</td>
<td>ZSTART, ACH, GAMIN, ATTACK, CONE, PIN, TIN</td>
</tr>
<tr>
<td>15</td>
<td>3E15.5</td>
<td>GAMFR, RQRI, SFR</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Format</th>
<th>Variable Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>80I1</td>
<td>IENT(M)</td>
</tr>
<tr>
<td></td>
<td>(M = 1, MC (IC) + MREG (IC))</td>
</tr>
<tr>
<td>6E13.5</td>
<td>RHLN(M), PHLN(M), UHLN(M), VHLN(M), WHLN(M) SHLN(M)</td>
</tr>
<tr>
<td></td>
<td>(M = 1, MC (IC) + MREG(IC))</td>
</tr>
</tbody>
</table>
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).

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

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 IREAD. 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 INCUT.

<table>
<thead>
<tr>
<th>Card No.</th>
<th>Format</th>
<th>Variable Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4F10.5</td>
<td>TESTM, TESTA, TESTG, TESTZ</td>
</tr>
<tr>
<td>2</td>
<td>E13.6</td>
<td>FNU</td>
</tr>
<tr>
<td>3</td>
<td>3I5</td>
<td>NS, JS, JCUT</td>
</tr>
</tbody>
</table>

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.
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.

<table>
<thead>
<tr>
<th>Card No.</th>
<th>Format</th>
<th>Variable Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4F10.5</td>
<td>TESTM, TESTA, TESTG, TESTZ</td>
</tr>
<tr>
<td>2</td>
<td>F10.4, 2I5</td>
<td>RCYL, NHPTS, JA</td>
</tr>
</tbody>
</table>

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.
| Figure 9 - SAMPLE INPUT DATA FOR QUICK |

**SCONP10: TEN DEGREE SHARP CONE**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDYLOWER</td>
<td>1ELL Piece BDYBOT EYYSID BDYLSCP</td>
</tr>
<tr>
<td>BDYUPPER</td>
<td>2ELL Piece BDYSID BDYTOP BDYUSCP</td>
</tr>
</tbody>
</table>

1. **MAP AXIS**
   0. 0. 20.

2. **YBDYBOT**
   1 LINE PIECE KV5
   0. 0. 20. 0.

3. **ZBDYBOT**
   1 LINE PIECE KV4
   0. 0. 15. A-10.

4. **3 LINE PIECE KV5**

5. **2 ELLX FILET KV0**
   1. 1. 3. 3. 10. 15.

6. **YBDYSID**
   1 LINE PIECE KV4
   0. 0. 15. A10.

7. **3 LINE PIECE KV5**
   10. 2. 20. 2.

8. **2 ELLX FILET KV0**
   1. 1. 3. 3. 10. 15.

9. **ZBDYSID YBDYBOT**
   **YBDYBOT**

10. **ZBDYTOP YBDYSID**
    **YBDYLSCP YBDYSID**

11. **ZBDYLSCP ZBDYBOT**
    **YBDYUSCP YBDYSID**

12. **ZBDYUSCP ZBDYTCP**
    **YMAPAXIS YBDYBOT**

13. **ZMAPAXIS YBDYBOT**
    1 2 5. 20. 5. -90. 90. 10.

14. 2 2 5. 20. 5. -90. 90. 30.

15. 3 2 5. 20. 5. -90. 90. 30.

16. 4 2 5. 20. 5. -90. 90. 30.

17. 5 1 5. 20. 5. -90. 90. 30.
Figure 10 - STEIN INPUT
OUTPUT FORMATS

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. 1lc 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. 1la. Each control point has two coordinates which must be defined \((y = f(x), z = g(x))\), and in Fig. 1lc, the index for a particular control point's \((n \text{ in Fig. 1la})\) \(z\) 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. 1la.

The output shown in Figs. 1ld 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. 1le 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. 1le 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 1lf 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 \(z\) \((VX = dV/dx, \text{ etc.})\). If MODE = - 1, no printed output will be generated, but the following output will be written on unit IPLOT:
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:

<table>
<thead>
<tr>
<th>Line</th>
<th>Variables</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IAM, IANDV</td>
<td>2I5</td>
</tr>
<tr>
<td>2</td>
<td>NXPTS, KNTBLM</td>
<td>2I5</td>
</tr>
<tr>
<td>3</td>
<td>XNOW</td>
<td>F10.4</td>
</tr>
<tr>
<td>4</td>
<td>V(I), VX(I), VXX(I) (I = 1, KNTBLM)</td>
<td>3F10.4</td>
</tr>
</tbody>
</table>

*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_θ, r_xx, and r_xθ.

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, UTHETA1(J) and UTHETA2(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_0, θ'_0, A_0^2, and B_0^2. 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, KNTHLM (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 I PLOT, 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:

<table>
<thead>
<tr>
<th>Line</th>
<th>Variables</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IAMD, IANDV</td>
<td>2I5</td>
</tr>
<tr>
<td>2</td>
<td>NXPTS, NHPTS</td>
<td>2I5</td>
</tr>
<tr>
<td>3</td>
<td>KARC, KNTARC</td>
<td>2I5</td>
</tr>
<tr>
<td>4</td>
<td>NOW</td>
<td>F10.5</td>
</tr>
<tr>
<td>5</td>
<td>Y, ZG, HNOWR*, RX*, RH*, RXX**, RXH**</td>
<td>7F10.5</td>
</tr>
</tbody>
</table>

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, \( \rho \) is the pressure \((\rho/\rho_m)\), 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 h₁ 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 = Q • n or Q • ζ) 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 radius 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.
Figure 11a - CROSS SECTION MODEL CHECK

Figure 11b - CHECK LIST MENU FOR BODY LINE MODELS
SCONEC: 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

<table>
<thead>
<tr>
<th>BODY LINE MODEL NUMBER</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 LINE (1)</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BODY LINE MODEL NUMBER</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 LINE (-1)</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2 ELLX (-3)</td>
<td>10.00000</td>
<td>-1.76327</td>
</tr>
<tr>
<td>3 LINE (1)</td>
<td>10.00000</td>
<td>-2.00000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BODY LINE MODEL NUMBER</th>
<th>3</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 LINE (1)</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2 ELLX (3)</td>
<td>10.00000</td>
<td>1.76327</td>
</tr>
<tr>
<td>3 LINE (1)</td>
<td>10.00000</td>
<td>2.00000</td>
</tr>
</tbody>
</table>

Figure 11d - BODY LINE MODEL CHECK TABLE 1
<table>
<thead>
<tr>
<th>Model ID</th>
<th>Body Line Model</th>
<th>Model Number</th>
<th>Number of Segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>YDBYBOT</td>
<td>BODY LINE MODEL</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><strong>BOUNDARY CONDITIONS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SEG SHAPE CONN DEF FREE GAP</td>
<td>X-ORIGIN</td>
<td>Y-ORIGIN</td>
</tr>
<tr>
<td>1 LINE</td>
<td>PIEC KV 5</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>SEG SHAPE EQUATION</td>
<td>A-COEFFICIENT</td>
<td>B-COEFFICIENT</td>
</tr>
<tr>
<td>1 LINE</td>
<td>0 = AX+BY</td>
<td>0.0</td>
<td>-0.20000000E 02</td>
</tr>
<tr>
<td></td>
<td>ALIAS LIST</td>
<td>ZBDYSTYD YBDYTOP YMAPAXIS ZMAPAXIS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model ID</th>
<th>Body Line Model</th>
<th>Model Number</th>
<th>Number of Segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>YDBYBOT</td>
<td>BODY LINE MODEL</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td><strong>BOUNDARY CONDITIONS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SEG SHAPE CONN DEF FREE GAP</td>
<td>X-ORIGIN</td>
<td>Y-ORIGIN</td>
</tr>
<tr>
<td>1 LINE</td>
<td>PIEC KV 4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2 ELXX FILL KV 0</td>
<td>10.00000</td>
<td>-1.76327</td>
<td>15.00000</td>
</tr>
<tr>
<td>3 LINE</td>
<td>PIEC KV 5</td>
<td>15.00000</td>
<td>-2.00000</td>
</tr>
<tr>
<td></td>
<td>SEG SHAPE EQUATION</td>
<td>A-COEFFICIENT</td>
<td>B-COEFFICIENT</td>
</tr>
<tr>
<td>1 LINE</td>
<td>0 = AX+BY</td>
<td>-0.26449032E 01</td>
<td>-0.15000000E 02</td>
</tr>
<tr>
<td>2 ELXX</td>
<td>0.48419382E+01</td>
<td>-0.27460033E 00</td>
<td>0.48419386E-02</td>
</tr>
<tr>
<td>3 LINE</td>
<td>0 = AX+BY</td>
<td>0.0</td>
<td>-0.10000000E 02</td>
</tr>
<tr>
<td></td>
<td>ALIAS LIST</td>
<td>ZBDYLYSCP</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model ID</th>
<th>Body Line Model</th>
<th>Model Number</th>
<th>Number of Segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZBDYSCP</td>
<td>BODY LINE MODEL</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td><strong>BOUNDARY CONDITIONS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SEG SHAPE CONN DEF FREE GAP</td>
<td>X-ORIGIN</td>
<td>Y-ORIGIN</td>
</tr>
<tr>
<td>1 LINE</td>
<td>PIEC KV 4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2 ELXX FILL KV 0</td>
<td>10.00000</td>
<td>1.76327</td>
<td>15.00000</td>
</tr>
<tr>
<td>3 LINE</td>
<td>PIEC KV 5</td>
<td>15.00000</td>
<td>2.00000</td>
</tr>
<tr>
<td></td>
<td>SEG SHAPE EQUATION</td>
<td>A-COEFFICIENT</td>
<td>B-COEFFICIENT</td>
</tr>
<tr>
<td>1 LINE</td>
<td>0 = AX+BY</td>
<td>-0.26449032E 01</td>
<td>-0.15000000E 02</td>
</tr>
<tr>
<td>2 ELXX</td>
<td>0.48419382E+01</td>
<td>-0.27460033E 00</td>
<td>0.48419386E-02</td>
</tr>
<tr>
<td>3 LINE</td>
<td>0 = AX+BY</td>
<td>0.0</td>
<td>-0.10000000E 02</td>
</tr>
<tr>
<td></td>
<td>ALIAS LIST</td>
<td>ZBDYLYSCP ZBDYTCP ZBDYUSCP ZBDYUSCF</td>
<td></td>
</tr>
</tbody>
</table>

---

Cross Section Check Against Body Lines
Cross Section Definition Check is Finished

Figure 11e - BODY LINE MODEL CHECK TABLE 2
Figure 1lf - SAMPLE QUICK INTERMEDIATE DATA DECK (MATH MODEL)
<table>
<thead>
<tr>
<th>INXBLM</th>
<th>INXBLS</th>
<th>V</th>
<th>VX</th>
<th>VXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-0.88163</td>
<td>-0.17633</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.88163</td>
<td>0.17633</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**XSTATION = 5.00000**

<table>
<thead>
<tr>
<th>INXBLM</th>
<th>INXBLS</th>
<th>V</th>
<th>VX</th>
<th>VXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-1.76327</td>
<td>-0.17633</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1.76327</td>
<td>0.17633</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**XSTATION = 10.00000**

<table>
<thead>
<tr>
<th>INXBLM</th>
<th>INXBLS</th>
<th>V</th>
<th>VX</th>
<th>VXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-2.00000</td>
<td>-0.00000</td>
<td>0.01295</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2.00000</td>
<td>0.00000</td>
<td>-0.01295</td>
</tr>
</tbody>
</table>

**XSTATION = 15.00000**

<table>
<thead>
<tr>
<th>INXBLM</th>
<th>INXBLS</th>
<th>V</th>
<th>VX</th>
<th>VXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>-2.00000</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2.00000</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**XSTATION = 20.00000**

---

Figure 12 - QUICK OUTPUT FOR MODE = 1
### Table 1: Ten Degree Sharp Cone

#### Geometry Check

**Station:** 5.00000

<table>
<thead>
<tr>
<th>Theta</th>
<th>Zmap</th>
<th>Zxmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.20</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.40</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.60</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.80</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

#### Derivatives Check

**Station:** 5.00000

<table>
<thead>
<tr>
<th>Theta</th>
<th>Zmap</th>
<th>Zxmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.20</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.40</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.60</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.80</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

---

**Figure 13 - Quick Output for Mode = 2**

---

**Note:** The table contains numerical data related to geometric and derivative checks for a ten-degree sharp cone, with specific calculations and values for different angles and coordinates.
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

0.5000000E01 0.881634E00 0.881634E00 0.0 0.0 -0.881634E00 0.0 0.0 0.0 0.0
9.100000E02 0.176327E01 0.176327E01 0.0 0.0 -0.176327E01 0.0 0.0 0.0 0.0
9.150000E02 0.200000E02 0.200000E02 0.0 0.0 -0.200000E02 0.0 0.0 0.0 0.0
9.200000E02 0.200000E02 0.200000E02 0.0 0.0 -0.200000E02 0.0 0.0 0.0 0.0

Figure 15 - QUICK OUTPUT FOR MODE = 4
Figure 15 (continued)

Figure 16 - QUICK OUTPUT FOR MODE = 5
**Figure 17 - INITIAL PLANE OUTPUT**

### INITIAL DATA

<table>
<thead>
<tr>
<th>V</th>
<th>E</th>
<th>W</th>
<th>L</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>W</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>-4.4181</td>
<td>1.4960</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>3</td>
<td>15.2151</td>
<td>-4.3202</td>
<td>0.1028</td>
<td>0.0338</td>
<td>0.0338</td>
<td>0.0338</td>
<td>0.0338</td>
<td>0.0338</td>
</tr>
<tr>
<td>4</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>5</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

### BOTTOM SYMMETRY PLANE

<table>
<thead>
<tr>
<th>V</th>
<th>E</th>
<th>W</th>
<th>L</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>W</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>-4.0047</td>
<td>1.4914</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>3</td>
<td>15.2151</td>
<td>-4.3202</td>
<td>0.1028</td>
<td>0.0338</td>
<td>0.0338</td>
<td>0.0338</td>
<td>0.0338</td>
<td>0.0338</td>
</tr>
<tr>
<td>4</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>5</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

### TOP SYMMETRY PLANE

<table>
<thead>
<tr>
<th>V</th>
<th>E</th>
<th>W</th>
<th>L</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>W</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>-4.0047</td>
<td>1.4914</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>3</td>
<td>15.2151</td>
<td>-4.3202</td>
<td>0.1028</td>
<td>0.0338</td>
<td>0.0338</td>
<td>0.0338</td>
<td>0.0338</td>
<td>0.0338</td>
</tr>
<tr>
<td>4</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>5</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
**Three-Dimensional Supersonic Flow Program**

**Z2 Start**: 47,000000
**Z2 End**: 66,70000

**KA**: 1000
**J**: 100
**CASL**: 2
**V**: 1
**JET**: 0
**ICAS**: 0

**A**: 0.00000
**ATTACK**: 0.0
**GAMMA**: 1.20000
**WING**:
- **Z2**: 0.10000 E 01
- **T**: 0.10000 E 01

**Z2 Shape**: 0.312000 E 02
**PIPS**: 0.312000 E 02
**S** = 0.312000 E 02

**Z2 Shape**: 0.70000 E 02
**WING**: 0.70000 E 02

**Figure 18 - Geometry Test Output**

### Geometry Test

**Z2 Geometry**: 0.0
**Z2 Geometry**: 0.667000 E 02
**Z2 Geometry**: 1.000000 E 01

**Mesh**: Quick Geometry for Model MSHA Configuration

### Wing Tip Surface

![Wing Tip Surface](image-url)

---

**Run Number**: 1

---

**Figure 18 - Geometry Test Output**

---

69
Figure 19 - BOW SHOCK AND ENTROPY LAYER SURFACE OUTPUT
AERODYNAMIC COEFFICIENTS USING

\[ P_{\text{INF}} = 1.0000 \]
\[ R_{\text{MGE}} = 0.10000 \times 10^1 \]
\[ V_{\text{IN}} = 9.4066 \]
\[ Q_{\text{IN}} = 44.2417 \]

Moments are taken about a line through

\[ Y_0 = 0.0 \]
\[ Z_0 = 10.0000 \]

Cone parameters

<table>
<thead>
<tr>
<th>Piece(s)</th>
<th>In Z-Range</th>
<th>Between Cont. Pts.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0000 - 20.0000</td>
<td>2.0 - 5.0</td>
</tr>
</tbody>
</table>

\[ C_L = 0.0098 \]
\[ C_D = 0.0365 \]
\[ C_M = -0.0002 \]
\[ C_N = 0.0110 \]
\[ C_A = 0.0362 \]

\[ \text{Area} = 398.941 \text{ sq. units} \]

Total parameters

\[ C_L = 0.0098 \]
\[ C_D = 0.0365 \]
\[ C_M = -0.0002 \]
\[ C_N = 0.0110 \]
\[ C_A = 0.0362 \]

\[ \text{Area} = 398.941 \text{ sq. units} \]

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
ZD = 10.0000

CONE PARAMETERS

FOR PIECE(S)  IN Z-RANGE BETWEEN CONT. PTS.

<table>
<thead>
<tr>
<th></th>
<th>1.0000</th>
<th>20.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>CL</td>
<td>0.3104</td>
<td></td>
</tr>
<tr>
<td>CD</td>
<td>1.1598</td>
<td></td>
</tr>
<tr>
<td>CM</td>
<td>-0.0355</td>
<td></td>
</tr>
<tr>
<td>CN</td>
<td>0.3507</td>
<td></td>
</tr>
<tr>
<td>CA</td>
<td>1.1483</td>
<td></td>
</tr>
</tbody>
</table>

TOTAL PARAMETERS

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CL</td>
<td>0.3104</td>
<td></td>
</tr>
<tr>
<td>CD</td>
<td>1.1598</td>
<td></td>
</tr>
<tr>
<td>CM</td>
<td>-0.0355</td>
<td></td>
</tr>
<tr>
<td>CN</td>
<td>0.3507</td>
<td></td>
</tr>
<tr>
<td>CA</td>
<td>1.1483</td>
<td></td>
</tr>
</tbody>
</table>

Figure 20b - AERODYNAMIC COEFFICIENTS OUTPUT 2
Figure 21a - STREAMLINE OUTPUT FROM STREMBL
### Pseudo Stream Surface Data

#### From Cut 1

<table>
<thead>
<tr>
<th>Point</th>
<th>Z</th>
<th>R</th>
<th>THTFA</th>
<th>X</th>
<th>Y</th>
<th>P</th>
<th>S</th>
<th>U</th>
<th>V</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95.4640</td>
<td>51.0686</td>
<td>-89.9999</td>
<td>0.0001</td>
<td>-51.0686</td>
<td>116509</td>
<td>26774</td>
<td>-0.0000</td>
<td>-0.2946</td>
<td>10.6624</td>
</tr>
<tr>
<td>2</td>
<td>95.4676</td>
<td>51.1121</td>
<td>-89.9999</td>
<td>0.0001</td>
<td>-51.1121</td>
<td>116455</td>
<td>26766</td>
<td>-0.0000</td>
<td>-0.2936</td>
<td>10.6660</td>
</tr>
<tr>
<td>3</td>
<td>95.4616</td>
<td>51.1555</td>
<td>-89.9999</td>
<td>0.0001</td>
<td>-51.1555</td>
<td>116401</td>
<td>26763</td>
<td>-0.0000</td>
<td>-0.2946</td>
<td>10.6678</td>
</tr>
<tr>
<td>4</td>
<td>95.4664</td>
<td>51.1989</td>
<td>-89.9999</td>
<td>0.0001</td>
<td>-51.1989</td>
<td>116348</td>
<td>26757</td>
<td>-0.0000</td>
<td>-0.2976</td>
<td>10.6693</td>
</tr>
<tr>
<td>5</td>
<td>95.4566</td>
<td>51.2424</td>
<td>-89.9999</td>
<td>0.0001</td>
<td>-51.2424</td>
<td>116295</td>
<td>26752</td>
<td>-0.0000</td>
<td>-0.2996</td>
<td>10.6762</td>
</tr>
<tr>
<td>6</td>
<td>95.4566</td>
<td>51.2858</td>
<td>-89.9999</td>
<td>0.0001</td>
<td>-51.2858</td>
<td>116242</td>
<td>26746</td>
<td>-0.0000</td>
<td>-0.3009</td>
<td>10.6931</td>
</tr>
<tr>
<td>7</td>
<td>95.4566</td>
<td>51.3293</td>
<td>-89.9999</td>
<td>0.0001</td>
<td>-51.3293</td>
<td>116189</td>
<td>26741</td>
<td>-0.0000</td>
<td>-0.3010</td>
<td>10.6921</td>
</tr>
<tr>
<td>8</td>
<td>95.4566</td>
<td>51.3727</td>
<td>-89.9999</td>
<td>0.0001</td>
<td>-51.3727</td>
<td>116137</td>
<td>26735</td>
<td>-0.0000</td>
<td>-0.3020</td>
<td>10.7051</td>
</tr>
<tr>
<td>9</td>
<td>95.4566</td>
<td>51.4161</td>
<td>-89.9999</td>
<td>0.0001</td>
<td>-51.4161</td>
<td>116085</td>
<td>26730</td>
<td>-0.0000</td>
<td>-0.3031</td>
<td>10.7110</td>
</tr>
<tr>
<td>10</td>
<td>95.4566</td>
<td>51.4596</td>
<td>-89.9999</td>
<td>0.0001</td>
<td>-51.4596</td>
<td>116034</td>
<td>26724</td>
<td>-0.0000</td>
<td>-0.3042</td>
<td>10.7170</td>
</tr>
</tbody>
</table>

#### Values Along Normal to Body...

<table>
<thead>
<tr>
<th>Point</th>
<th>N LENGTH</th>
<th>DPDN</th>
<th>DSDN</th>
<th>DUPDN</th>
<th>DVDN</th>
<th>DWDN</th>
<th>DVELDN</th>
<th>VFLOTN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>-0.1297</td>
<td>-0.0127</td>
<td>0.0000</td>
<td>-0.0227</td>
<td>0.1430</td>
<td>0.1435</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0.0436</td>
<td>-0.1242</td>
<td>-0.0127</td>
<td>0.0000</td>
<td>-0.0230</td>
<td>0.1423</td>
<td>0.1426</td>
<td>0.0008</td>
</tr>
<tr>
<td>3</td>
<td>0.0469</td>
<td>-0.1233</td>
<td>-0.0127</td>
<td>0.0000</td>
<td>-0.0233</td>
<td>0.1415</td>
<td>0.1421</td>
<td>0.0017</td>
</tr>
<tr>
<td>4</td>
<td>0.1364</td>
<td>-0.1226</td>
<td>-0.0128</td>
<td>0.0000</td>
<td>-0.0235</td>
<td>0.1406</td>
<td>0.1412</td>
<td>0.0025</td>
</tr>
<tr>
<td>5</td>
<td>0.1736</td>
<td>-0.1219</td>
<td>-0.0128</td>
<td>0.0000</td>
<td>-0.0238</td>
<td>0.1399</td>
<td>0.1405</td>
<td>0.0034</td>
</tr>
<tr>
<td>6</td>
<td>0.2173</td>
<td>-0.1210</td>
<td>-0.0128</td>
<td>0.0000</td>
<td>-0.0241</td>
<td>0.1391</td>
<td>0.1397</td>
<td>0.0042</td>
</tr>
<tr>
<td>7</td>
<td>0.2607</td>
<td>-0.1197</td>
<td>-0.0128</td>
<td>0.0000</td>
<td>-0.0244</td>
<td>0.1382</td>
<td>0.1389</td>
<td>0.0051</td>
</tr>
<tr>
<td>8</td>
<td>0.3042</td>
<td>-0.1197</td>
<td>-0.0128</td>
<td>0.0000</td>
<td>-0.0247</td>
<td>0.1375</td>
<td>0.1381</td>
<td>0.0060</td>
</tr>
<tr>
<td>9</td>
<td>0.3449</td>
<td>-0.1188</td>
<td>-0.0128</td>
<td>0.0000</td>
<td>-0.0250</td>
<td>0.1367</td>
<td>0.1374</td>
<td>0.0069</td>
</tr>
<tr>
<td>10</td>
<td>0.3911</td>
<td>-0.1173</td>
<td>-0.0129</td>
<td>0.0000</td>
<td>-0.0253</td>
<td>0.1361</td>
<td>0.1366</td>
<td>0.0079</td>
</tr>
</tbody>
</table>

### From Streamline 2

<table>
<thead>
<tr>
<th>Point</th>
<th>Z</th>
<th>R</th>
<th>THTFA</th>
<th>X</th>
<th>Y</th>
<th>P</th>
<th>S</th>
<th>U</th>
<th>V</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95.4640</td>
<td>51.2321</td>
<td>-85.4270</td>
<td>4.0090</td>
<td>-51.0686</td>
<td>117583</td>
<td>26772</td>
<td>-0.0163</td>
<td>-0.2945</td>
<td>10.6518</td>
</tr>
<tr>
<td>2</td>
<td>95.4676</td>
<td>51.2753</td>
<td>-85.4259</td>
<td>4.0090</td>
<td>-51.1120</td>
<td>117530</td>
<td>26767</td>
<td>-0.0157</td>
<td>-0.2946</td>
<td>10.6579</td>
</tr>
<tr>
<td>3</td>
<td>95.4616</td>
<td>51.3186</td>
<td>-85.4296</td>
<td>4.0090</td>
<td>-51.1555</td>
<td>117479</td>
<td>26762</td>
<td>-0.0151</td>
<td>-0.2967</td>
<td>10.6639</td>
</tr>
<tr>
<td>4</td>
<td>95.4566</td>
<td>51.3620</td>
<td>-85.4337</td>
<td>4.0090</td>
<td>-51.1989</td>
<td>117425</td>
<td>26756</td>
<td>-0.0146</td>
<td>-0.2979</td>
<td>10.6700</td>
</tr>
<tr>
<td>5</td>
<td>95.4592</td>
<td>51.4052</td>
<td>-85.4375</td>
<td>4.0090</td>
<td>-51.2424</td>
<td>117374</td>
<td>26751</td>
<td>-0.0140</td>
<td>-0.2990</td>
<td>10.6760</td>
</tr>
<tr>
<td>6</td>
<td>95.4564</td>
<td>51.4918</td>
<td>-85.4452</td>
<td>4.0090</td>
<td>-51.3292</td>
<td>117271</td>
<td>26740</td>
<td>-0.0129</td>
<td>-0.3014</td>
<td>10.6879</td>
</tr>
<tr>
<td>7</td>
<td>95.4566</td>
<td>51.5351</td>
<td>-85.4520</td>
<td>4.0090</td>
<td>-51.3727</td>
<td>117220</td>
<td>26734</td>
<td>-0.0123</td>
<td>-0.3025</td>
<td>10.6936</td>
</tr>
<tr>
<td>8</td>
<td>95.4556</td>
<td>51.5784</td>
<td>-85.4592</td>
<td>4.0090</td>
<td>-51.4161</td>
<td>117169</td>
<td>26729</td>
<td>-0.0118</td>
<td>-0.3038</td>
<td>10.6997</td>
</tr>
<tr>
<td>9</td>
<td>95.4532</td>
<td>51.6717</td>
<td>-85.4567</td>
<td>4.0090</td>
<td>-51.4595</td>
<td>117119</td>
<td>26723</td>
<td>-0.0112</td>
<td>-0.3050</td>
<td>10.7055</td>
</tr>
</tbody>
</table>

#### Normal Derivatives at Same Locations...

<table>
<thead>
<tr>
<th>Point</th>
<th>N LENGTH</th>
<th>DPDN</th>
<th>DSDN</th>
<th>DUPDN</th>
<th>DVDN</th>
<th>DWDN</th>
<th>DVELDN</th>
<th>VFLOTN</th>
</tr>
</thead>
</table>

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)

<table>
<thead>
<tr>
<th>INDEX</th>
<th>HC</th>
<th>P</th>
<th>S</th>
<th>U</th>
<th>V</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1.5708</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>2</td>
<td>-1.4902</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>3</td>
<td>-1.4097</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>4</td>
<td>-1.3291</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>5</td>
<td>-1.2486</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>6</td>
<td>-1.1680</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>7</td>
<td>-1.0875</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>8</td>
<td>-1.0069</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>9</td>
<td>-0.9264</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>10</td>
<td>-0.8458</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>11</td>
<td>-0.7653</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>12</td>
<td>-0.6847</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>13</td>
<td>-0.6042</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>14</td>
<td>-0.5236</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>15</td>
<td>-0.4430</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>16</td>
<td>-0.3625</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>17</td>
<td>-0.2819</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>18</td>
<td>-0.2014</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>19</td>
<td>-0.1208</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>20</td>
<td>-0.0403</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>21</td>
<td>0.0403</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>22</td>
<td>0.1208</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>23</td>
<td>0.2014</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>24</td>
<td>0.2819</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>25</td>
<td>0.3625</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>26</td>
<td>0.4430</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>27</td>
<td>0.5236</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>28</td>
<td>0.6042</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>29</td>
<td>0.6847</td>
<td>1.0000</td>
<td>0.0</td>
<td>0.0</td>
<td>13.8108</td>
<td>23.9210</td>
</tr>
<tr>
<td>30</td>
<td>0.7653</td>
<td>1.6333</td>
<td>0.8962</td>
<td>2.7111</td>
<td>12.4647</td>
<td>24.3062</td>
</tr>
<tr>
<td>31</td>
<td>0.8458</td>
<td>4.6277</td>
<td>1.6488</td>
<td>1.2488</td>
<td>10.7615</td>
<td>23.4038</td>
</tr>
<tr>
<td>32</td>
<td>0.9264</td>
<td>2.6368</td>
<td>2.0147</td>
<td>-0.0347</td>
<td>10.1836</td>
<td>23.0840</td>
</tr>
<tr>
<td>33</td>
<td>1.0069</td>
<td>1.6773</td>
<td>2.2607</td>
<td>-1.1147</td>
<td>9.3094</td>
<td>22.3604</td>
</tr>
<tr>
<td>35</td>
<td>1.1680</td>
<td>0.9108</td>
<td>2.5899</td>
<td>-3.2451</td>
<td>9.0350</td>
<td>20.5599</td>
</tr>
<tr>
<td>36</td>
<td>1.2486</td>
<td>0.7451</td>
<td>2.7254</td>
<td>-4.1283</td>
<td>8.3530</td>
<td>19.6477</td>
</tr>
<tr>
<td>37</td>
<td>1.3291</td>
<td>0.7149</td>
<td>2.8538</td>
<td>-4.6716</td>
<td>7.3395</td>
<td>18.0988</td>
</tr>
<tr>
<td>38</td>
<td>1.4097</td>
<td>1.9142</td>
<td>3.0574</td>
<td>-1.3095</td>
<td>4.5444</td>
<td>17.7060</td>
</tr>
<tr>
<td>39</td>
<td>1.4902</td>
<td>2.7432</td>
<td>3.2551</td>
<td>-1.3244</td>
<td>3.9880</td>
<td>18.2131</td>
</tr>
<tr>
<td>40</td>
<td>1.5708</td>
<td>2.5597</td>
<td>3.3075</td>
<td>-0.0001</td>
<td>3.7832</td>
<td>18.4587</td>
</tr>
</tbody>
</table>

Figure 22 - SONIC BOOM DATA CYLINDER OUTPUT

75
STORAGE REQUIREMENTS AND COMPUTER TIME FOR QUICK

Using the IBM G-compiler, QUICK requires approximately 128K\textsubscript{10} bytes of core to compile (\approx 40K\textsubscript{8} words), and 176K\textsubscript{10} bytes to execute (\approx 54K\textsubscript{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\textsubscript{10} bytes (on the IBM 370/168). The core required for logic without overlay is 400K\textsubscript{10}. So that 580K\textsubscript{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\textsubscript{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\textsubscript{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\textsubscript{\infty} = 10 and an angle of attack of 30\degree. 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 downstream 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\textsubscript{10} bytes to compile (\approx 37K\textsubscript{8} words), and 190K\textsubscript{10} bytes to execute (\approx 60K\textsubscript{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.
BOOM 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
INITIALIZATION OF VARIABLES K = 0

K = K + 1

CHANGE CHEMISTRY (FREEZE) AND OR CHANGE MESH POINTS, INSERT TIP SURFACE

COMPUTE DZ

OUTPUT DATA PLANE ON TAPE FOR STRMBL & BOOM

COMPUTE BODY & SHOCK POSITION AT Z + DZ

LOOP = 0

COMPUTE INTERIOR POINTS

COMPUTE WING SHOCK POINTS

COMPUTE CROSS FLOW SHOCK POINTS

COMPUTE ENTROPY LAYER POINTS

LOOP = LOOP + 1

INTEGRATE AERO COEFFICIENTS

Z = Z + DZ

OUTPUT

END OF RUN

YES

STOP

NO

Figure 24 - STEIN OVERALL LOGIC
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, ELgeom, 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, AEOCF, 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 \( \theta' \).

CSGEO 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 (INNEW) 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 $\theta$.

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.)

ENTRIA 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 ≠ 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.

EXSPAN computes the flow through a 2-D centered expansion corner. Given the upstream Mach number (XM1), GAMLO(N,M) and the flow deflection (DELTA). EXSPAN will compute the conditions after the expansion (pressure ratio P2/Q1, temperature ratio T2/T1, Mach number XM2 and the slope (BETA) of the first expansion wave).
FREEZ is called at a station $Z = Z_{\text{FREEZ}}$ 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, $P (\ln p/p_\infty)$ 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 $ICUT = 1$. 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 $Z = \text{constant}$ 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, (r_x, r_y, r_z, \theta_x, \theta_y, \theta_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, HYH, RZH, RXZ, RYZ, RZZ, HXZ, HZZ 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(ln 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 execu-
tion is terminated for any reason. When requested, aero-
dynamic coefficients are also output from this routine. OUTPUT
also writes (on unit IFUNCH) 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 $P(\ln \frac{\rho}{\rho_0})$ on either side of the expansion, the entropy (constant through the expansion) and the velocity in the plane of the fan ($V_{N1}$) PRAN computes the change in flow direction $DXNU$.

RANK computes the flow through a shock. Given $V_{N1}$ (velocity normal to the shock), $\Gamma_1$ (the value of $\Gamma_{NLO(N,M)}$), $P_1 \ln(\frac{\rho}{\rho_0})$, $S_1$ (entropy), $T_1(\frac{\rho}{\rho})$, and $H_1$ (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 ($U(N,I) = U(N,MC(IC) + MREG(IC)) = 0$ 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 $\delta = 5 \sqrt{\frac{v^2}{M\sqrt{\gamma}}} = 5 \times 2/\sqrt{Re_z}$.

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.)

CSCGOM (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 \( \frac{d\theta_S}{dz} \) for the given streamline, integrates to find \( \theta_S \) (circumferential location of the streamline) and \( S_\eta \) (arc length measured along the streamline), and determines \( r_S \) (radial position of the streamline). \((\theta_S', S_\eta', \text{and } r_S \text{ at } Z + DZ)\).

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 \((z', r', \theta')\) of a given point lying along the body surface normal taken at a specified \( z \) and \( \theta' \).

MAIN2 is a subroutine, but acts as a second main program once STRMBL has established the \( z \) and \( \theta' \) 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 (CONT'D)

A ≡ CSGEOM

BLMSET

BLGEOM

BLGEOM

CSMSET

CSMINT

CSMCOE

CSMFLT

THELIM

BLGEOM

B ≡ CSCALC

VDOTV

MDOTV
STEIN TREE DIAGRAM (CONT'D)

BLMSET
  BLGEOM
  CSMSET
    CSMCOE
    CSMINT
    CSMFLT
      BLGEOM
      CSCALC
      VDOTV
      MDOTV
    CSCALC
      VDOTV
      MDOTV
  CSCALC
    VDOTV
    MDOTV
  TIPSUR
    IMAP
  UPDATE
  MINTER

NINTER

SHTIP
  B
  UPDATE
    GAS
      A
STRMBL TREE DIAGRAM

STRMBL

- INOUT
  - GEOMIN
    - A
      - DELTHE
      - INTERH — FLINE
      - INTRH1 — FLINE
    - SOUT
    - LOCATE — A
  - LOCATE — A
    - BRCKTO
    - BRCKT
  - MAIN2
    - INTR3D
    - NOUT
      - FLINE
      - INTERH — FLINE
      - INTR2D
      - INTER — FLINE
      - FLINE

A ≡ AS DEFINED IN QUICK TREE DIAGRAM
(A) = AS DEFINED IN QUICK TREE DIAGRAM
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 \((x, r, \theta)\); 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 \(y_{\text{max}}\); 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 \((\theta)\). 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 $\theta$.

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**

<table>
<thead>
<tr>
<th>SHAPE</th>
<th>KEYWORD</th>
<th>EQUATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE</td>
<td>LINE</td>
<td>(Ay + Bz + C = 0)</td>
</tr>
<tr>
<td>ELLIPSE (Concave to Origin)</td>
<td>ELLI</td>
<td>(\frac{(y - y_o)^2}{A^2} + \frac{(z - z_o)^2}{B^2} = 1)</td>
</tr>
<tr>
<td>ELLIPSE (Convex to Origin)</td>
<td>ELLO</td>
<td>Same</td>
</tr>
</tbody>
</table>

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)

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

*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

<table>
<thead>
<tr>
<th>Shape</th>
<th>Keyword</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>LINE</td>
<td>Ax+By = 0</td>
</tr>
<tr>
<td>x-Parabola</td>
<td>XPAR</td>
<td>Ax+By+y² = 0</td>
</tr>
<tr>
<td>y-Parabola</td>
<td>XPAR</td>
<td>Ax+By+x² = 0</td>
</tr>
<tr>
<td>Rotated x-Parabola</td>
<td>RXPA</td>
<td>Ax+By+Cxy+y² = 0</td>
</tr>
<tr>
<td>Rotated y-Parabola</td>
<td>RYPA</td>
<td>Ax+By+Cxy+x² = 0</td>
</tr>
<tr>
<td>x-Ellipse</td>
<td>ELLX</td>
<td>Ax+By+Cx²+y² = 0</td>
</tr>
<tr>
<td>y-Ellipse</td>
<td>ELLY</td>
<td>Ax+By+Cx²+x² = 0</td>
</tr>
<tr>
<td>Cubic</td>
<td>CUBI(C)</td>
<td>Ax+By+Cx²+x³ = 0</td>
</tr>
</tbody>
</table>

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 \( \tan(10^\circ) = 0.176327 \), 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 ZBDYTCL with YBDYUTN. 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
I PLOT = 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
CS.MDEF
CSMCHK
BLMDEF
BLMCHK
KRVDEF
GEMOUT
GEMCHK
MODE1
MODE2
MODE3

117
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 (-π/2 ≤ θ ≤ +π/2)

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

RX = dr/dx at this point

RH = dr/dθ at this point
RXX = \frac{d^2r}{dx^2} at this point

RXH = d^2r/dx_d\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 Al - 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
<table>
<thead>
<tr>
<th>Card 1</th>
<th>FORMAT</th>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 1-80</td>
<td>15A4</td>
<td>VTITLE</td>
<td>Vehicle or run title.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Card 2</th>
<th>FORMAT</th>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 1-2</td>
<td>I2</td>
<td>NCSM</td>
<td>Number of distinct cross section models.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Card 3</th>
<th>(There will be exactly NCSM number of these cards appearing together with the appropriate cards of type 4.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 1-2</td>
<td>I2</td>
</tr>
<tr>
<td>Col. 3-4</td>
<td>I2</td>
</tr>
<tr>
<td>Col. 11-50</td>
<td>10A4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Card 4</th>
<th>(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.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 1-8</td>
<td>A8</td>
</tr>
<tr>
<td>Col. 9-10</td>
<td>I2</td>
</tr>
<tr>
<td>Col. 11-14</td>
<td>A4</td>
</tr>
<tr>
<td>Col. 17-20</td>
<td>A4</td>
</tr>
<tr>
<td>Col. 25</td>
<td>A1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Col. 26-33</td>
<td>A8</td>
</tr>
<tr>
<td>Col. 35</td>
<td>A1</td>
</tr>
</tbody>
</table>

Figure A4 - DATA INPUT FORMAT FOR CROSS SECTION MODELING
Card type 4 (Cont)

<table>
<thead>
<tr>
<th>FORMAT</th>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 36-43</td>
<td>A8</td>
<td>PNTNAM(2)</td>
</tr>
<tr>
<td>Col. 46-53</td>
<td>A8</td>
<td>PNTNAM(3)</td>
</tr>
<tr>
<td>Col. 56-63</td>
<td>A8</td>
<td>ARCNM(1)</td>
</tr>
<tr>
<td>Col. 66-73</td>
<td>A8</td>
<td>ARCNM(2)</td>
</tr>
</tbody>
</table>

Card type 5 (Appears after NCSM blocks of one card type 3 and KARC cards of type 4.)

<table>
<thead>
<tr>
<th>FORMAT</th>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 1-2</td>
<td>I2</td>
<td>KNTCSM</td>
</tr>
<tr>
<td>Col. 11-18</td>
<td>A8</td>
<td>ZMAPNM</td>
</tr>
</tbody>
</table>

Card type 6 (There will be exactly KNTCSM number of these cards.)

<table>
<thead>
<tr>
<th>FORMAT</th>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 1-2</td>
<td>I2</td>
<td>KDUM</td>
</tr>
<tr>
<td>Col. 3-4</td>
<td>I2</td>
<td>MODEL</td>
</tr>
<tr>
<td>Col. 11-20</td>
<td>F10.5</td>
<td>XCSMS1</td>
</tr>
<tr>
<td>Col. 21-30</td>
<td>F10.5</td>
<td>XCSMS2</td>
</tr>
</tbody>
</table>

Figure A4 - DATA INPUT FORMAT FOR CROSS SECTION MODELING (Continued)
Figure A5 - SAMPLE FOREBODY WITH LOGICAL CROSS SECTION DEFINITION
<table>
<thead>
<tr>
<th>Card 1</th>
<th>FORMAT</th>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 1</td>
<td>A1</td>
<td>BYORZ</td>
<td>The letter Y or Z to indicate which data definition is to follow (a blank terminates all modeling input data).</td>
</tr>
<tr>
<td>Col. 2-9</td>
<td>A8</td>
<td>BNAME</td>
<td>Body Line/Control Point name which is to be defined.</td>
</tr>
<tr>
<td>Col. 11</td>
<td>A1</td>
<td>AYORZ</td>
<td>The letter Y or Z to indicate which definition is to be used when aliasing (blank when not).</td>
</tr>
<tr>
<td>Col. 12-19</td>
<td>A8</td>
<td>ANAME</td>
<td>Body Line/Control Point name to which BNAME is to be aliased, when applicable (blank when not).</td>
</tr>
<tr>
<td>Col. 31-70</td>
<td>10A4</td>
<td>TITLE</td>
<td>Any comments.</td>
</tr>
</tbody>
</table>

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)
<table>
<thead>
<tr>
<th>FORMAT</th>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card 3 (if not aliasing) (see note for Card 2) (Note: If SSHAPE is NULL, this card is deleted)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Col. 1-10</td>
<td>F10.5</td>
<td>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.</td>
</tr>
<tr>
<td>Col. 11-20</td>
<td>F10.5</td>
<td>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.</td>
</tr>
<tr>
<td>Col. 21-30</td>
<td>F10.5</td>
<td>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.</td>
</tr>
<tr>
<td>Col. 31-39</td>
<td>F9.4</td>
<td>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.</td>
</tr>
<tr>
<td>Col. 40</td>
<td>A1</td>
<td>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).</td>
</tr>
<tr>
<td>Col. 41-49</td>
<td>F9.4</td>
<td>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.</td>
</tr>
<tr>
<td>Col. 50</td>
<td>A1</td>
<td>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).</td>
</tr>
<tr>
<td>Col. 51-60</td>
<td>F10.5</td>
<td>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.</td>
</tr>
</tbody>
</table>

Figure A6 - DATA INPUT FORMAT FOR BODY LINE MODELING (Sheet 2 of 2)
**Figure A7**

### Fortran Coding Form

<table>
<thead>
<tr>
<th>COLUMN 1</th>
<th>COLUMN 2</th>
<th>COLUMN 3</th>
<th>COLUMN 4</th>
<th>COLUMN 5</th>
<th>COLUMN 6</th>
<th>COLUMN 7</th>
<th>COLUMN 8</th>
<th>COLUMN 9</th>
<th>COLUMN 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>02</td>
<td>03</td>
<td>04</td>
<td>05</td>
<td>06</td>
<td>07</td>
<td>08</td>
<td>09</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
</tr>
<tr>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
</tr>
<tr>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
</tr>
<tr>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
</tr>
<tr>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
</tr>
</tbody>
</table>

**Coding Instructions**

1. Alphabetic characters are written as follows: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
2. Numerical characters are written as follows: 0 1 2 3 4 5 6 7 8 9
3. Cards with a C in Column 1 are not processed by Fortran and Column 2 is may be used for comments
4. Column 2 of the first line of a statement may contain a data item.
5. Column 3 is used to indicate the type of statement by inserting a digit in this column. The coding of this column is subject to limitations set forth in Section 6.2.
6. Column 4 is used for continuation cards. The digit in this column is subject to limitations set forth in Section 6.2.
7. Column 5 is used for continuation cards. The digit in this column is subject to limitations set forth in Section 6.2.
8. Column 6 is used for continuation cards. The digit in this column is subject to limitations set forth in Section 6.2.
9. Column 7 is used for continuation cards. The digit in this column is subject to limitations set forth in Section 6.2.
10. Column 8 is used for continuation cards. The digit in this column is subject to limitations set forth in Section 6.2.
11. Column 9 is used for continuation cards. The digit in this column is subject to limitations set forth in Section 6.2.
12. Column 10 is used for continuation cards. The digit in this column is subject to limitations set forth in Section 6.2.

**KEYPUNCH VERIFIED**
**Figure A8**
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>AbcDefGhijklmnoprstuvwxyz</td>
</tr>
<tr>
<td>4-6</td>
<td>Numerical characters are written as follows</td>
</tr>
</tbody>
</table>
| 7-9    | Continuation Card 1
| 10-12  | Continuation Card 2
| 13-15  | Continuation Card 3
| 16-18  | Continuation Card 4 |

**Figure A8 (Con't)**
<table>
<thead>
<tr>
<th>FORMAT</th>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 1-2</td>
<td>I2</td>
<td>MODE</td>
</tr>
<tr>
<td>- 0 (or blank), terminates all input.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ± 1, creates body line traces.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ± 2, creates cross sectional cuts.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ± 3, interrogates cross sections in neighborhood of control points.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- - 3, allows multiple body line traces to create plan and profile views.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ± 4, comparison of analytic derivatives with numerically formed derivatives.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ± 5, check of unit vectors normal to body surface.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ± 6, exercises modes 1, 2, and 3 at the limits of each cross sectional model.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- - 6, exercises modes - 2 and - 7 at the limits of each cross sectional model.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- - 7, (plotting mode only) creates cross sectional cuts, but includes all arcs in their entirety (including growing pieces still contained within the basic skin).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Col. 4-5</td>
<td>I2</td>
<td>NDERV</td>
</tr>
<tr>
<td>- ± N, where N is the order of derivative to be calculated (N=0, 1 or 2).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ 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).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- N, should not be used for these interrogations (requires previous call to same location (x and 8), radius and certain temporary variables are not recomputed).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Col. 11-20</td>
<td>F10.5</td>
<td>XGO</td>
</tr>
<tr>
<td>Initial x-station to be interrogated.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Col. 21-30</td>
<td>F10.5</td>
<td>XEND</td>
</tr>
<tr>
<td>Final x-station to be interrogated.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Col. 31-40</td>
<td>F10.5</td>
<td>XDEL</td>
</tr>
<tr>
<td>Increment size in x, to establish outputs stations between XGO and XEND.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Col. 41-50</td>
<td>F10.5</td>
<td>HGO</td>
</tr>
<tr>
<td>Initial value of theta (in degrees) to be interrogated; not required for modes 1, 3.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Col. 51-60</td>
<td>F10.5</td>
<td>HEND</td>
</tr>
<tr>
<td>Final value of theta (in degrees) to be interrogated; not required for modes 1, 3.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Col. 61-70</td>
<td>F10.5</td>
<td>HDEL</td>
</tr>
<tr>
<td>Increment size in degrees, to establish interrogation points between HGO and HEND; not required for modes 1, 3.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure A9 - DATA INPUT FORMAT FOR EXERCISING THE GEOMETRIC MODEL
APPENDIX A-A

QUICK GEOMETRY MODELING PACKAGE

EXAMPLES
NCEF111A: QUICK GEOMETRY FOR THE EF-111R (N-PAFONE)

1 2  NOSE TO START OF FLATS
   BDYLOELL 1ELLI PIECE   BDUSDTN BDUSCP
   BDYUPELL 2ELLI PIECE   BDUSDTN BDUSCP

2 4  FLATS TO START OF CANOPY
   BDYLOFLT 1LINE PIECE   BDUSDTN BDUSCP
   BDYLOELL 3ELLI PATCH   BDUSDTN BDUSCP
   BDYSOFTL 2LINE PIECE   BDUSDTN BDUSCP
   BDYUPELL 4ELLI ALINK   BDUSDTN BDUSCP

3 5  CANOPY TO START OF RADOME
   BDYLOFLT 1LINE PIECE   BDUSDTN BDUSCP
   BDYLOELL 3ELLI PATCH   BDUSDTN BDUSCP
   BDYSOFTL 2LINE PIECE   BDUSDTN BDUSCP
   BDYUPELL 4ELLI ALINK   BDUSDTN BDUSCP
   CANOPY 5ELLI PIECE     CNSTCN CNSCP BDYUPELL

4 6  RADOME TO START OF WING
   BDYLOFLT 1LINE PIECE   BDUSDTN BDUSCP
   BDYLOELL 3ELLI PATCH   BDUSDTN BDUSCP
   BDYSOFTL 2LINE PIECE   BDUSDTN BDUSCP
   BDYUPPELL 4ELLI ALINK   BDUSDTN BDUSCP
   RADOME 6ELLI PIECE     RDBLCL RDMEDG RDMSCP BDYLOFLT
   WINGLOELL 7ELLI PIECE   WGTBM WGEDG WGLSCP BDYUPELL
   WINGUPELL 8ELLI PIECE   WGTOP WGLSCP BDYUPELL

5 8  WING TO INLET LIP
   BDYLOFLT 1LINE PIECE   BDUSDTN BDUSCP
   BDYLOELL 3ELLI PATCH   BDUSDTN BDUSCP
   BDYSOFTL 2LINE PIECE   BDUSDTN BDUSCP
   BDYUPELL 4ELLI ALINK   BDUSDTN BDUSCP
   BDYUPELL2 5ELLI ALINK   BDUSDTN BDUSCP
   RADOME 6ELLI PIECE     RDBLCL RDMEDG RDMSCP BDYLOFLT
   WINGLOELL 7ELLI PIECE   WGTBM WGEDG WGLSCP BDYUPELL
   WINGUPELL 8ELLI PIECE   WGTOP WGLSCP BDYUPELL

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

VBDUCL
   1 LINE PIECE KU5
   0.  0.  440.15 0.
   -1

ZBDUCL
   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.  -17.25
   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
<table>
<thead>
<tr>
<th>PIECE</th>
<th>KU5</th>
<th>KU6</th>
<th>KU7</th>
<th>KU8</th>
<th>KU9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>1 LINE</td>
<td>PIECE KU5</td>
<td>100.</td>
<td>3.6</td>
<td>3.</td>
<td>90.</td>
</tr>
<tr>
<td>2 ELIX</td>
<td>FILET KU0</td>
<td>140.</td>
<td>3.</td>
<td>3.</td>
<td>90.</td>
</tr>
<tr>
<td>3 LINE</td>
<td>PIECE KU5</td>
<td>100.</td>
<td>3.6</td>
<td>3.</td>
<td>90.</td>
</tr>
<tr>
<td>4 ELIX</td>
<td>PATCH KU0</td>
<td>6.75</td>
<td>440.15</td>
<td>2.</td>
<td>90.</td>
</tr>
<tr>
<td>5 LINE</td>
<td>PIECE KU5</td>
<td>100.</td>
<td>3.6</td>
<td>3.</td>
<td>90.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PIECE</th>
<th>KU4</th>
<th>KU5</th>
<th>KU6</th>
<th>KU7</th>
<th>KU8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>1 LINE</td>
<td>PIECE KU4</td>
<td>100.</td>
<td>3.6</td>
<td>3.</td>
<td>90.</td>
</tr>
<tr>
<td>2 ELIX</td>
<td>ALINK KU0</td>
<td>180.</td>
<td>30.16</td>
<td>160.</td>
<td>28.78</td>
</tr>
<tr>
<td>3 RYPA</td>
<td>ALINK KU0</td>
<td>328.</td>
<td>32.8</td>
<td>300.</td>
<td>33.32</td>
</tr>
<tr>
<td>4 RYPA</td>
<td>ALINK KU0</td>
<td>440.15</td>
<td>34.52</td>
<td>424.</td>
<td>34.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PIECE</th>
<th>KU4</th>
<th>KU5</th>
<th>KU6</th>
<th>KU7</th>
<th>KU8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>1 LINE</td>
<td>PIECE KU4</td>
<td>100.</td>
<td>3.6</td>
<td>3.</td>
<td>90.</td>
</tr>
<tr>
<td>2 ELIX</td>
<td>PATCH KU0</td>
<td>180.</td>
<td>30.16</td>
<td>160.</td>
<td>28.78</td>
</tr>
<tr>
<td>3 LINE</td>
<td>PIECE KU5</td>
<td>36.016</td>
<td>274.</td>
<td>70.</td>
<td>90.</td>
</tr>
<tr>
<td>4 RYPA</td>
<td>FILET KU0</td>
<td>440.15</td>
<td>34.52</td>
<td>424.</td>
<td>34.</td>
</tr>
<tr>
<td>5 LINE</td>
<td>PIECE KU5</td>
<td>180.</td>
<td>30.16</td>
<td>160.</td>
<td>28.78</td>
</tr>
<tr>
<td>6 ELIX</td>
<td>FILET KU0</td>
<td>65.12</td>
<td>440.15</td>
<td>61.64</td>
<td>90.</td>
</tr>
<tr>
<td>7 ELIX</td>
<td>FLINK KU0</td>
<td>69.5</td>
<td>9.</td>
<td>9.</td>
<td>400.</td>
</tr>
<tr>
<td>8 ELIX</td>
<td>FLINK KU0</td>
<td>68.16</td>
<td>8.</td>
<td>8.</td>
<td>300.</td>
</tr>
<tr>
<td>9 LINE</td>
<td>PIECE KU5</td>
<td>5.</td>
<td>7.</td>
<td>7.</td>
<td>260.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PIECE</th>
<th>KU4</th>
<th>KU5</th>
<th>KU6</th>
<th>KU7</th>
<th>KU8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>1 LINE</td>
<td>PIECE KU4</td>
<td>100.</td>
<td>3.6</td>
<td>3.</td>
<td>90.</td>
</tr>
<tr>
<td>2 ELIX</td>
<td>ALINK KU0</td>
<td>180.</td>
<td>30.16</td>
<td>160.</td>
<td>28.78</td>
</tr>
<tr>
<td>3 RYPA</td>
<td>ALINK KU0</td>
<td>328.</td>
<td>32.8</td>
<td>300.</td>
<td>33.32</td>
</tr>
<tr>
<td>4 RYPA</td>
<td>ALINK KU0</td>
<td>440.15</td>
<td>34.52</td>
<td>424.</td>
<td>34.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PIECE</th>
<th>KU4</th>
<th>KU5</th>
<th>KU6</th>
<th>KU7</th>
<th>KU8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>1 LINE</td>
<td>PIECE KU4</td>
<td>100.</td>
<td>3.6</td>
<td>3.</td>
<td>90.</td>
</tr>
<tr>
<td>2 ELIX</td>
<td>PATCH KU0</td>
<td>180.</td>
<td>30.16</td>
<td>160.</td>
<td>28.78</td>
</tr>
<tr>
<td>3 LINE</td>
<td>PIECE KU5</td>
<td>36.016</td>
<td>274.</td>
<td>70.</td>
<td>90.</td>
</tr>
<tr>
<td>4 RYPA</td>
<td>FILET KU0</td>
<td>440.15</td>
<td>34.52</td>
<td>424.</td>
<td>34.</td>
</tr>
<tr>
<td>5 LINE</td>
<td>PIECE KU5</td>
<td>180.</td>
<td>30.16</td>
<td>160.</td>
<td>28.78</td>
</tr>
<tr>
<td>6 ELIX</td>
<td>FILET KU0</td>
<td>65.12</td>
<td>440.15</td>
<td>61.64</td>
<td>90.</td>
</tr>
<tr>
<td>7 ELIX</td>
<td>FLINK KU0</td>
<td>69.5</td>
<td>9.</td>
<td>9.</td>
<td>400.</td>
</tr>
<tr>
<td>8 ELIX</td>
<td>FLINK KU0</td>
<td>68.16</td>
<td>8.</td>
<td>8.</td>
<td>300.</td>
</tr>
<tr>
<td>9 LINE</td>
<td>PIECE KU5</td>
<td>5.</td>
<td>7.</td>
<td>7.</td>
<td>260.</td>
</tr>
<tr>
<td>2 RVPA</td>
<td>PATCH KU0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>1. 3. 3.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

-1

<table>
<thead>
<tr>
<th>2BDBTMTN 2DBBCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 LINE PIECE KU4</td>
</tr>
<tr>
<td>0. 12. 12. 12.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2 ELLX ALINK KU0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 180. 30.16 160. 28.78</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4 LINE PIECE KU5</th>
</tr>
</thead>
<tbody>
<tr>
<td>333.36 35.4 440.15 36.52</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3 RYP PATCH KU0</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. 4. 4. 4.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1 NULL</th>
</tr>
</thead>
</table>

-1

<table>
<thead>
<tr>
<th>2BDBSOTN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 LINE PIECE KU5</td>
</tr>
<tr>
<td>160. 140. 3.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3 LINE PIECE KU5</th>
</tr>
</thead>
<tbody>
<tr>
<td>144. 3.6 204. 14.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5 LINE PIECE KU5</th>
</tr>
</thead>
<tbody>
<tr>
<td>204. 14.0 333.36 41.74</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7 LINE PIECE KU5</th>
</tr>
</thead>
<tbody>
<tr>
<td>333.36 41.74 440.15 32.48</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2 ELLV FILET KU0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 1. 3. 3. 140. 160.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4 ELLX FILET KU0</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. 3. 5. 5. 190. 210.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6 RYP FILET KU0</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. 5. 7. 7. 328. 338.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1 NULL</th>
</tr>
</thead>
</table>

-1

<table>
<thead>
<tr>
<th>YCNDTCL YBDOCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 LINE PIECE KU5</td>
</tr>
<tr>
<td>180. 36.016 230. 56.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2 ELLX ALINK KU0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 274. 68.16 300. 69.8</td>
</tr>
</tbody>
</table>

-1

<table>
<thead>
<tr>
<th>YCNBTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ELLX PIECE KU0</td>
</tr>
<tr>
<td>180. 0. 280. 32.5 180. 25.7</td>
</tr>
</tbody>
</table>

-1

<table>
<thead>
<tr>
<th>ZCNBTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ELLY PIECE KU0</td>
</tr>
<tr>
<td>180. 36.016 200. 32. 190. 32.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3 LINE PIECE KU5</th>
</tr>
</thead>
<tbody>
<tr>
<td>210. 33. 280. 43.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2 ELLY PATCH KU0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 1. 3. 3.</td>
</tr>
</tbody>
</table>

-1

<table>
<thead>
<tr>
<th>YCNSCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ELLX PIECE KU0</td>
</tr>
<tr>
<td>180. 0. 280. 20. 180. 16.</td>
</tr>
</tbody>
</table>
1 NULL
-1
VRRMIG
  1 LINE PIECE KU5
  290. 12. 440. 15. 12.
-1
VRRMIG
  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 RVPA FILET KU0
  1.  1.  3.  3. 240. 255.
  4 RVPA FILET KU0
  3.  3.  5.  5. 285. 305.
  6 RVPA ALINK KU0
  5.  5.  472. -10.2 433. -18.12
1 NULL
-1
VRRMIG VRRMIG
VRRMIG VRRMIG
VRRMIG VRRMIG
VRRMIG VRRMIG
  1 LINE PIECE KU5
  1.  0.  80.  0. 0.
  2 ELLY PIECE KU0
  80. 0.  180.  20. 140. 0.
  5 LINE PIECE KU5
  290. 47.96 440. 15 43.28
  4 RVPA 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

1 2 NOSE TO START OF BOTTOM FLAT
BDVLOELL 2ELLI PIECE BOVDVBL BOVDTUP BDLOSCP
BDVUPPELL 1ELLI PIECE BDSDTUP BDVUPSCP

2 3 BOTTOM FLAT TO START OF SIDE FLAT
FLATBTM 1LINE PIECE BOVDVBL BDVTMTN
BDVLOELL 3ELLI PATCH BDVTMTN BDSDTUP
BDVUPPELL 2ELLI PIECE BDSDTUP BDVUPSCP

3 4 FLATS TO START OF CANOPY
FLATBTM 1LINE PIECE BOVDVBL BDVTMTN
BDVLOELL 3ELLI PATCH BDVTMTN BDSDTLO
FLATSIDE 2LINE PIECE BDSDTLO BDSDTUP
BDVUPPELL 4ELLI PIECE BDSDTUP BDVUPSCP

4 5 CANOPY TO START OF FAIRING
FLATBTM 1LINE PIECE BOVDVBL BDVTMTN
BDVLOELL 3ELLI PATCH BDVTMTN BDSDTLO
FLATSIDE 2LINE PIECE BDSDTLO BDSDTUP
BDVUPPELL 4ELLI PIECE BDSDTUP BDVUPSCP
CANOPY 5ELLI PIECE CNPVTOP CNPVSCP BDVUPPELL

5 6 FAIRING TO START OF WING (AND END OF CANOPY)
FLATBTM 1LINE PIECE BOVDVBL BDVTMTN
BDVLOELL 3ELLI PATCH BDVTMTN BDSDTLO
FLATSIDE 2LINE PIECE BDSDTLO BDSDTUP
BDVUPPELL 4ELLI PIECE BDSDTUP BDVUPSCP
CANOPY 5ELLI PIECE CNPVTOP CNPVSCP BDVUPPELL
FAIRING 6ELLI PIECE FRNGTOP FRNGSCP CANOPY

6 6 WING TO START OF TAIL
FLATBTM 1LINE PIECE BOVDVBL BDVTMTN
WGVLOELL 3ELLI PATCH BDVTMTN WINGE
WINGUPPER 2ELLI PIECE WINGE BDSDTLO WINGSCP
FLATSIDE 4LINE PIECE BDSDTLO BDSDTUP
BDVUPPELL 5ELLI PIECE BDSDTUP BDVUPSCP
FAIRING 6ELLI PIECE FRNGTOP FRNGSCP BDVUPPELL

7 7 TAIL TO END
FLATBTM 1LINE PIECE BOVDVBL BDVTMTN
WGVLOELL 3ELLI PATCH BDVTMTN WINGE
WINGUPPER 2ELLI PIECE WINGE BDSDTLO WINGSCP
FLATSIDE 4LINE PIECE BDSDTLO BDSDTUP
BDVUPPELL 5ELLI PIECE BDSDTUP BDVUPSCP
FAIRING 6ELLI PIECE FRNGTOP FRNGSCP BDVUPPELL
VERTTATE 7ELLI PIECE TLFRINT TAILTOP TAILSCP FAIRING

7 MAPAXIS
1 1 0. 20.
2 2 30. 40.
3 3 40. 130.01
4 4 130.01 226.6
5 5 226.6 360.
6 6 360. 1077.8769
7 7 1077.8769 1280.5206

VBOVDVBL
1 LINE PIECE KUS
0. 0. 1280.5206 0.
-1
<table>
<thead>
<tr>
<th>Piece KU5</th>
<th>1.</th>
<th>2.</th>
<th>3.</th>
<th>4.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILET KU0</td>
<td>1.</td>
<td>2.</td>
<td>3.</td>
<td>4.</td>
</tr>
<tr>
<td>LINE KU5</td>
<td>5.</td>
<td>6.</td>
<td>7.</td>
<td>8.</td>
</tr>
<tr>
<td>PIECE KU5</td>
<td>9.</td>
<td>10.</td>
<td>11.</td>
<td></td>
</tr>
</tbody>
</table>

Note: The table represents various parameters and measurements for different pieces, with columns indicating different specifications and values.
<table>
<thead>
<tr>
<th>Line</th>
<th>PIECE KU5</th>
<th>FLINK KU0</th>
<th>FILET KU0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34.</td>
<td>440.15</td>
<td>35.12</td>
</tr>
<tr>
<td>2</td>
<td>40.2</td>
<td>440.15</td>
<td>89.2</td>
</tr>
<tr>
<td>3</td>
<td>41.24</td>
<td>440.15</td>
<td>32.48</td>
</tr>
<tr>
<td>4</td>
<td>47.96</td>
<td>440.15</td>
<td>43.28</td>
</tr>
<tr>
<td>5</td>
<td>48.8</td>
<td>440.15</td>
<td>48.8</td>
</tr>
<tr>
<td>6</td>
<td>48.8</td>
<td>440.15</td>
<td>56.</td>
</tr>
<tr>
<td>7</td>
<td>48.8</td>
<td>440.15</td>
<td>56.</td>
</tr>
</tbody>
</table>

**Note:** The table contains measurements and values related to different components and lines.
<table>
<thead>
<tr>
<th>Piece</th>
<th>Width</th>
<th>Height</th>
<th>Offset</th>
<th>Thickness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piec1</td>
<td>226.6</td>
<td>309.0</td>
<td>24.359</td>
<td>226.6</td>
</tr>
<tr>
<td>Piec2</td>
<td>309.0</td>
<td>1280.5206</td>
<td>24.359</td>
<td></td>
</tr>
<tr>
<td>Piec3</td>
<td>276.3705</td>
<td>155.0</td>
<td>1280.5206</td>
<td>155.0</td>
</tr>
<tr>
<td>Piec4</td>
<td>226.6</td>
<td>190.0</td>
<td>109.0</td>
<td></td>
</tr>
<tr>
<td>Piec5</td>
<td>226.6</td>
<td>309.0</td>
<td>19.0</td>
<td>226.6</td>
</tr>
<tr>
<td>Piec6</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec7</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec8</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec9</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec10</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec11</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec12</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec13</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec14</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec15</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec16</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec17</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec18</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec19</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec20</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec21</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec22</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec23</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec24</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec25</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec26</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec27</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec28</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec29</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec30</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec31</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec32</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec33</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec34</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec35</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec36</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec37</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec38</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec39</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec40</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec41</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec42</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec43</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec44</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec45</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec46</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec47</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec48</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec49</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec50</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec51</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec52</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec53</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec54</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec55</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec56</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec57</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec58</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>Piec59</td>
<td>226.6</td>
<td>19.0</td>
<td>226.6</td>
<td></td>
</tr>
<tr>
<td>Piec60</td>
<td>309.0</td>
<td>1280.5206</td>
<td>19.0</td>
<td></td>
</tr>
</tbody>
</table>
VWINGSCP  VYINGLE
ZWINGSCP  ZBDGDTLO
VTFLPRINT
 2 LINE PIECE KUS  
 1082.  4.4  1215.4  18.4
 1 ELLX FLINK KUO  
1077.8769  0.  2.  2.  1077.8769  10.
 4 LINE PIECE KUS  
1215.4  18.4  1280.5206  9.2
 3 RYPA FILET KUO  
 2.  2.  4.  4.  1210.4  1220.4
-1
VTFLPRINT
 1 LINE PIECE KUS  
1077.8769 163.  1280.5206 163.
-1
VTAILTOP  VBODYBCL
2TAILTOP
 2 LINE PIECE KUS  
1107.  192.6  1366.8  453.
 1 ELLX FLINK KUO  
1077.8769 183.  2.  2.  1000.  183.
-1
VTAILSCP  VTFLPRINT
2TAILSCP  ZTAILTOP
ZMAPAXIS  VBODYBCL
2MAPAXIS
 1 LINE PIECE KUS  
 0.  0.  300.  0.
 3 LINE PIECE KUS  
360.  -30.3826  1105.  -30.3826
 2 CUBIC PATCH KUO  
 1.  1.  3.  3.
 5 LINE PIECE KUS  
1105.  -30.3826  1280.5206  -60.
 4 ELLY FILET KUO  
 3.  3.  5.  5.  1090.  1120.
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 $\phi$ direction (maximum of 8)  
(Fig. Bl)

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 ($C_p/C_v$) in free stream

STAB  Stability factor for C.F.L. condition ($DT = DT_{min}(STAB)$).  
Typically STAB = 1.2.

THEMAX  Limit on $\theta$. 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.

X0  Location of center of coordinate system (Fig. B1) (X0 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 (\(p_\infty/p_{SL}\)) use only when IGAS = 1.

TIN  Free stream temperature (\(T_\infty/T_{SL}\)) 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)

IFUNCH  Output unit for initial data plane results.
Figure B1 - COORDINATE SYSTEM

Figure B2 - INITIAL DATA PLANE

149

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