



**SIMPLIFIED INPUT FOR CERTAIN
AERODYNAMIC NOSE CONFIGURATIONS TO THE
GRUMMAN QUICK-GEOMETRY SYSTEM
(A KWIKNOSE USER'S MANUAL)**

**Frederick L. Shope
ARO, Inc., a Sverdrup Corporation Company**

**VON KÁRMÁN GAS DYNAMICS FACILITY
ARNOLD ENGINEERING DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
ARNOLD AIR FORCE STATION, TENNESSEE 37389**

February 1978

Final Report for Period January to September 1977

Approved for public release; distribution unlimited.

Prepared for

**ARNOLD ENGINEERING DEVELOPMENT CENTER/DOTR
ARNOLD AIR FORCE STATION, TENNESSEE 37389**

NOTICES

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

Qualified users may obtain copies of this report from the Defense Documentation Center.

References to named commercial products in this report are not to be considered in any sense as an indorsement of the product by the United States Air Force or the Government.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

APPROVAL STATEMENT

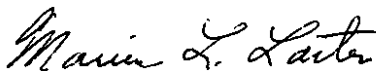
This report has been reviewed and approved.



ELTON R. THOMPSON
Project Manager, Research Division
Directorate of Test Engineering

Approved for publication:

FOR THE COMMANDER



MARION L. LASTER
Director of Test Engineering
Deputy for Operations

UNCLASSIFIED

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AEDC-TR-77-89	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SIMPLIFIED INPUT FOR CERTAIN AERODYNAMIC NOSE CONFIGURATIONS TO THE GRUMMAN QUICK-GEOMETRY SYSTEM (A KWIKNOSE USER'S MANUAL)		5. TYPE OF REPORT & PERIOD COVERED Final Report - January to September 1977
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Frederick L. Shope, ARO, Inc.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Arnold Engineering Development Center/DOT Air Force Systems Command Arnold Air Force Station, Tennessee 37389		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element 65807F
11. CONTROLLING OFFICE NAME AND ADDRESS Arnold Engineering Development Center/DOS Arnold Air Force Station, Tennessee 37389		12. REPORT DATE February 1978
		13. NUMBER OF PAGES 93
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Available in DDC		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) handbook user needs input Fortran computer programs geometric forms mathematical model flow fields aerodynamic configurations		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report is a user's manual for a Fortran computer program KWIKNOSE which, for certain axisymmetric and nonaxisymmetric nose configurations, provides simplified geometric input to the Grumman QUICK-geometry system, which in turn provides geometric information to various numerical flow codes. For a wide variety in choice of input parameters, KWIKNOSE sets up the QUICK input for an arbitrary sequence of conical and ogival sections. In this process, KWIKNOSE		

UNCLASSIFIED

UNCLASSIFIED

20. ABSTRACT (Continued)

performs the tedious computations necessary to locate the intersection points of successive arcs and to insert optional fillets or rounds over nontangent intersections. In addition, the code is capable of inserting arbitrary multiple slicing planes into the top, bottom, and side of the vehicle. Slicing plane intersections may be filleted or rounded. Thus, for a minimum of input and manual calculation by the user, KWIKNOSE is tailored to modeling the geometry of a sliced multiconic vehicle capped with an asymmetrically ablated nose. This manual provides check cases for the various geometry options, a description of input and output, and a listing of the source deck.

UNCLASSIFIED

PREFACE

The work reported herein was conducted by the Arnold Engineering Development Center (AEDC), Air Force Systems Command (AFSC), under program element 65807F. The results were obtained by ARO, Inc., AEDC Division (a Sverdrup Corporation Company), operating contractor for the AEDC, AFSC, Arnold Air Force Station, Tennessee. The research was done under ARO Project No. V33A-A8A. Elton R. Thompson was the Air Force project manager. The manuscript was submitted for publication on September 2, 1977.

CONTENTS

	<u>Page</u>
1.0 INTRODUCTION	5
2.0 PROGRAMMED GEOMETRIES	8
3.0 PROGRAM DESCRIPTION	
3.1 Input	11
3.2 Output	16
3.3 Processing Information	17
3.4 Programming Considerations	18
4.0 SUMMARY	21
REFERENCES	21

ILLUSTRATIONS

Figure

1. Typical Geometries Available via KWIKNOSE	23
2. Relative Regions of Applicability of Programs	
KWIKNOSE and PREKWIK	24
3. Oblique View of Asymmetrically Ablated Nose	25
4. Oblique View of Asymmetrical Nose from Sample Case	26
5. Oblique View of Asymmetrically Ablated, Bent, and	
Sliced Cone (Sample Case)	27
6. Nose-Afterbody Sample Case with Notation for	
Input Variables	28
7. Body Line Notation	29
8. Cross Section and Body Line Models	30

TABLES

1. Input to KWIKNOSE via Namelist/INPUT/	31
2. KWIKNOSE Input Options for Three Main Body Lines	
(ZUCL, ZLCL, YSID)	34

<u>TABLES</u> (continued)	<u>Page</u>
3. Input Description for Variable DIR(I)	35
4. Nomenclature for Input Variable DIR(I)	36
5. Jobstream for Sample Case	37
6. KWIKNOSE Printout for Sample Case	38
7. Definitions of Logical Names for QUICK Cross-Section and Body Line Models	47
8. KWIKNOSE Source Listing	48
9. Subroutine Description	85
10. Variable Description	87

1.0 INTRODUCTION

This document describes a computer program which allows greatly simplified input, for certain common axisymmetric and nonaxisymmetric nose-vehicle geometries of general interest, to the more general Grumman QUICK-geometry system. QUICK, a Fortran computer program developed by Vachris and Yaeger (Refs. 1 and 2), provides the user with a simple and well organized method of constructing and interrogating a mathematical model of an arbitrary, three-dimensional flight vehicle. This geometry system has primary application to the numerical computation of the flow fields over arbitrary aerodynamic configurations. QUICK was developed in two phases, the QUICKDEF phase, which provides facilities to build and check the geometry model, and the QUICKLOK phase, which provides surface coordinates and derivatives to user programs without requiring of the user a detailed knowledge of how the model was constructed. Current applications of the QUICKLOK phase include the time-dependent blunt body program BLUNT (Ref. 3); the supersonic, three-dimensional, external, inviscid flow-field code STEIN (Refs. 3 and 4); and the multishocked, three-dimensional, supersonic finite difference method of Kutler, Reinhardt, and Warming (Ref. 5). The QUICKDEF phase, of primary concern here, is a highly user-oriented program which greatly simplifies the geometric input by providing general curve fitting services to define arbitrary bodies from a minimum of logical and numerical input. In essence, QUICK requires the user to define each unique cross section of the vehicle in a plane normal to the coordinate axis of the vehicle by merely identifying the types of curves (lines, ellipses, parabolas, ...) which form the segmented cross section. The cross sections then become numerically specific when the user further defines, logically and numerically, certain key control points which determine the intersections and slopes of the curves comprising the cross sections. The only two limitations imposed by QUICK upon the vehicle geometry are (1) that the vehicle must have a vertical plane of symmetry and (2) that each cross section must be definable in terms of a single-valued radius

vector swung about a single point within the cross section. While these limitations prevent complete treatment of certain vehicle geometries (such as engine inlets, the oblique wing supersonic transport, or the double vertical stabilizer of the F-15), QUICK is an important advance in the description of vehicle geometry and makes practical the solution of flows over such complicated bodies as the space shuttle orbiter.

In contrast, QUICK is also well suited to the treatment of relatively simple bodies such as sphere-cones, biconics, and slab deltas, which are of basic interest to aerodynamicists. It was therefore deemed logical to develop yet another sequential computer program to set up input to QUICK for certain such simple geometries, thus reducing user-prepared input to a matter of two or three cards for the programmed geometries. This fourth program PREKWIK is documented in a previous user's manual.

During the development of PREKWIK, the need was recognized to study more complicated nose shapes than the simple spherical cap treated by PREKWIK. Interest centered around the effects of ablated nose shapes (Fig. 1) upon the aerodynamics of otherwise simple bodies such as cones and biconics with slices on the conic portions of the vehicle. A feature which most of the symmetrical noses have in common is that they can all be modeled (with varying degrees of accuracy) as a stack of conical and ogival sections which intersect in different fashions. Thus, it appeared practical to program the logic needed to model these geometries and set up the input to QUICK. Initially, the nose code was envisioned as a modification to PREKWIK, but the need for a fundamentally different programming approach soon forced the nose geometry problem into a separate computer program. The essential goal of PREKWIK was to receive a minimum of geometric input data via very few input options for a given geometry, perform a few relatively trivial intersection computations, and then generate the card image input required by QUICK. For the programmed geometries, PREKWIK eliminated the tedious chore of preparing

a new QUICK deck for every minor quantitative change in geometry (such as a different bluntness, cone half-angle, or vehicle length). In contrast, the nose geometry problem was perceived as logically more complicated. Not only would the nose program have to compute the intersections of more complicated functions, but the capability to round or fillet sharp corners was deemed essential. However, the analytic geometry of rounds and fillets - determining the equation and intersection points of a circle tangent to two intersecting curves - can result in very high degree polynomials along with the attendant logic problems of root sorting to choose the correct intersection. The computational complexity of the intersection problem coupled with the fairly large number of ways a given geometry can be specified via various combinations of known parameters forces the geometry program to accept many different input combinations from which must be computed the same basic set of QUICK input coordinates and slopes. If the nose code did not accept a fairly broad range of input combinations, then the typical user (who didn't have quite the right input combination) might have to spend a great deal of time deriving and debugging the intersection equations. After he had done all this, the computation and assembly of the final QUICK input would seem a fairly minor task. In the meantime, he would have surely forgotten all about the nose code.

Such is the logic upon which is based the structure of the nose geometry code KWIKNOSE, the subject of this user's manual. KWIKNOSE should be viewed as a complement to PREKWIK: whereas the latter is intended to model the overall vehicle, KWIKNOSE is specialized for arbitrary asymmetrical nose shapes. With these two programs, then, a flow program deriving its geometric data from QUICK could handle, as an example, the symmetrically ablated bent cone of Fig. 2. A KWIKNOSE/QUICK/BLUNT/STEIN chain could carry the computation to some point on the forecone, and a PREKWIK/QUICK/STEIN chain could proceed from that point through the bend to the end of the aft cone. (The user may have to construct a fictitious sphere-cone forebody on the PREKWIK model to

place the aft body at the same position relative to the geometric origin of the KWIKNOSE model:.)

The asymmetrical nose geometries which KWIKNOSE can generate are limited to cases where the top, bottom, and side body lines can be described in terms of sequential series of straight line and circle segments (see Figs. 3 and 4) as was done for the symmetrical case. The upper and lower half of every cross section must be describable in terms of an ellipse, though the ellipses at successive cross sections need not be concentric or geometrically similar (may have different centers and different major and minor axes). All filleting, rounding, and slicing capabilities are available in the asymmetrical case. However, if the tip of the asymmetric nose does not fall on the centerline of the afterbody (or the first conical section of a stack), the resulting geometry model will be incompatible with the PREKWIK geometry model for the afterbody. However, KWIKNOSE does not distinguish between nose and afterbody sections, and a fairly general asymmetric nose atop a stack of cones and ogives (with bends) can be modeled entirely with KWIKNOSE without reference to PREKWIK (Fig. 5). Some additional geometric calculation may have to be done by the user.

The remainder of this report describes in greater detail the programmed geometries and the code itself. A description of the input and output is given, and a sample case is presented to illustrate use of the code. The necessary processing information is listed, and certain portions of the programming logic are elaborated upon.

2.0 PROGRAMMED GEOMETRIES

The configurations which can be modeled via KWIKNOSE fall into two general categories: axisymmetric and nonaxisymmetric. The symmetric configuration is an arbitrary sequence of conical and ogival sections which may or may not be tangent at the intersection points. The ogival

sections may be convex or concave relative to the centerline of the vehicle. Nontangent intersections may, at the user's option, be smoothed with circular fillets or rounds. The fillets (or rounds) may be inserted between cone-cone intersections, cone-ogive intersections, or ogive-ogive intersections. The fillets need not be very small in relation to the intersecting curves and may therefore form major segments of the body. The only restriction is that the adjacent segments being filleted intersect in the real plane. For some cone-ogive and ogive-ogive combinations, there may exist a maximum filleting radius beyond which a real solution does not exist. In general, no segment may contain a point with infinite slope, the nosetip being the one exception. The tip of the nose may be either sharp or blunt and is always used as the geometric origin for the entire geometry. No segment may wrap back on itself so as to require a double-valued radius from any single point on the centerline axis. The program dimensions limit the geometry model to ten or fewer segments, and each fillet occupies one segment (see Table 1, note 8). Since KWIKNOSE does not distinguish between nose and afterbody portions of the vehicle, the user may allocate the ten segments between these two portions as needed.

The program contains logic to insert slicing planes into the basic axisymmetric vehicle described above. Provisions are made for an upper slice, a side slice, and a lower slice, each being composed of nine or fewer straight-line segments. In the side view the top and bottom slicing planes are seen as segmented lines, and the side slicing plane is seen as a segmented line in the top view. Thus, there are no provisions for skewed slices. The slices may not cross the centerline of the vehicle and may not intersect each other inside the region of the unsliced vehicle at a particular station. Corners at the intersection of adjacent segments may be rounded or filleted at the user's option. Although the slices are assumed to be planar, the fillets need not be small and may therefore comprise major portions of the slicing segments. It should be understood that the slicing body lines are input to QUICK

independently of the other body lines and thus do not have to intersect the unsliced vehicle at all points or any points. QUICK automatically checks whether the slicing body lines intersect the unsliced body and chooses the appropriate curve for the outer skin of the vehicle. The slices therefore may enter and leave the body as needed. There are no provisions for filleting or rounding in the circumferential direction (as viewed in planes normal to the centerline of the vehicle).

The asymmetrical case retains all combinations of cone and ogive intersection options (or line-circle intersections in a particular radial plane), the rounding and filleting of the outer body lines, and slicing with rounds and fillets. The asymmetrical vehicles treatable by KWIKNOSE must be such that the top, bottom, and side surface body lines may be represented by segments of lines and circles. When viewed in cross section, the vehicle must be describable with two ellipses (not necessarily identical) for the upper and lower half-planes. At a given cross section, the horizontal line dividing the two half-planes need not intersect the centerline (or geometric axis through the nosetip), and provisions are available for defining this deviation. For the asymmetrical case the top and bottom surface body lines may cross the geometric axis, and the map axis (from which QUICK swings the radius vector which must remain single valued) may also depart from the geometric axis if needed. Thus, the asymmetrical option allows, for example, the modeling of an ablated nose atop a bent cone.

For certain asymmetric configurations there is a potential inaccuracy of which the user should be aware. Consider a geometry such as is illustrated in Fig. 3 but without the round between the fore and aft sections. This geometric model was constructed with the geometric axis (X-axis) through the nosetip and parallel to (but not coincident with) the aft cone centerline. In the actual hardware (before rounding), the intersection of the fore and aft sections when viewed from the side would appear as a slanted line. Discontinuities in the surface slope

would occur at this line in moving along a body line. In the math model, however, the slope discontinuity would occur only on the upper, lower, and side body lines and would be smeared everywhere in between. This smearing results from two conditions: (1) QUICK assembles cross sections in planes normal to the geometric axis; (2) as modeled in Fig. 3, the slanted line between the fore and aft bodies is oblique to the geometric axis. While a carefully constructed manual model for input to QUICK probably could represent this case accurately, programming of the general case for arbitrary intersections of cones and ogives was considered beyond the scope of KWIKNOSE.

The user has one alternative if he requires an accurate model from KWIKNOSE: provide the input data relative to a geometric axis normal to the slanted line. Note, however, that the geometric axis might penetrate the skin and require bending of the map axis to keep QUICK's radius vector single-valued. In addition, once the geometric axis has been made normal to a particular intersection plane, no other similar intersection with a different slant angle can be modeled with equal accuracy. For two conical sections of different half-angles and oblique centerlines, the intersection "plane" is really curved. Thus, reorienting the geometric axis can only reduce (but not eliminate) the inaccuracy in modeling.

3.0 PROGRAM DESCRIPTION

3.1 INPUT

The input to KWIKNOSE is described in Tables 1 through 4, and a listing of the KWIKNOSE/QUICK jobstream with the input for the sample case of Figs. 4 through 6 is given in Table 5. It is suggested that the user, prior to assembling the input deck, make a qualitatively accurate sketch of the upper, lower, and side body lines and indicate the known geometric data. This task has been found useful in choosing appropriate

input options and deciding on the values of the logical input variables (DIR and SLOPE), which are needed by the code for selection of a segment intersection from multiple roots.

The first input card is for identification and may contain any 80 characters, though QUICK will carry only 60 of them. The remaining input is via namelist /INPUT/. For a symmetrical body (with or without asymmetrical slices), only one namelist input is required. For an asymmetrical body three separate namelists must follow the single identification card. The first must define the upper body line (ZUCL, see Figs. 7 and 8), the second the lower body line (ZLCL), and the last the side body line (YSID). Note that the input variable Z is used for the radial coordinate regardless of whether the input data represents the Y or Z direction. The slicing body lines (ZUSL, ZLSL, and YSSL) and the Z coordinates of the side and map axis (ZSID and ZMAPAX) may be included in any of the three namelists, if they are input at all. The upper, lower, and side body lines are processed independently of each other and need have no particular correspondence with each other except for the same final X coordinate at the end of the last segment. Thus, each may be an entirely different combination of lines, circles, and fillets. The sign notation for input coordinates and angles is that of a cylindrical coordinate system. Angles (TH(I)) are positive away from the centerline and negative toward it relative to a line parallel to the centerline.

The input rules for the three main body lines are somewhat different than for the slicing body lines and will be discussed separately. For each of the three main body lines (upper, lower, side) to be defined, the user must input the element of the character array CURVE(I) as 'LINE' or 'OGIV' for each segment. Based on this choice for each individual segment, the user then chooses an input option from Table 2 for each of the segments, preferably an option which matches the combination of geometric data he has for the segment. If any redundant data is to

be input, case 17 or 18 (see ICASE values in Table 2) must be chosen. If the redundant data is not self consistent to within ERRMAX, execution will be halted. All data items indicated by X's in Table 2 for a particular ICASE value must be input or execution will be halted. Note that ICASE(I) is not an input variable. The data elements described in Table 1 are then coded according to the standard rules for namelist input as found in any Fortran manual. The user should note that X(I), Z(I), and TH(I) are the coordinates and slope at the end of the Ith segment (CURVE(I)). Further, certain input cases may not be used following certain other cases. For example, cases 5 and 9 constitute inputting only the equation of the segment and in themselves do not include sufficient information to locate the intersection of the end of the segment. Thus, those cases which assume that the coordinates of the end of the previous segment are known may not follow cases 5 and 9. Such limitations, the need for which will be clarified in Section 3.4, are given in the footnotes to Tables 1 and 2.

The direction indicator variable DIR(I) requires additional clarification. Whenever ogival segments are involved, the possibility of multiple intersections with adjacent segments exists. In addition, when an ogive is to be inserted between two given points, there are usually two circle equations which will satisfy all the numeric data. The user therefore must indicate to KWIKNOSSE which of the possibilities is to be chosen using DIR(I). Table 2 indicates which input cases require DIR(I) and Table 3 defines the acceptable values of DIR(I) for each ICASE value. The nomenclature for these acceptable values is defined in Table 4. The 'OPUP', 'OPDN', 'TOPS', and 'BOTS' values apply to all three main body lines (upper, lower, and side) as though they were rotated about the centerline for viewing as oriented in Table 4. That is, DIR(I) = 'OPUP' indicates that the centerline sees a convex segment and 'OPDN' a concave segment. Similarly, 'TOPS' always indicates the intersection farther from the centerline than the 'BOTS' intersection.

If a nontangent intersection is to be filleted or rounded, this fact is indicated by giving the radius of the round or fillet $RHO(I)$ to be inserted at the end of the I th segment. Fillets/rounds are not included when numbering segments for input, and no data other than the radius are input for the fillet. The intersections and center are computed internally by KWIKNOSE. There are several restrictions on the filleting (or rounding) process. Segments already tangent may not be filleted. Only adjacent segments which actually intersect may be filleted because the location of this intersection is used to choose between multiple roots. Further, some types of intersections may have maximum filleting radii beyond which real solutions do not exist. If KWIKNOSE detects any of these conditions, usually through negative square roots or division by zero, execution will be halted.

For asymmetrical bodies, such as a bent cone, the user may opt to define the Z-coordinate of the side (ZSID, Figs. 6 and 8) as different from zero, in which case he must input two or more values of XSID and ZSID as described in Table 1. XSID(1) and ZSID(1) must both be zero; and if ZSID is to be input at all, at least two values of each (XSID(1) and XSID(2), ZSID(1) and ZSID(2) must be given. If the vehicle is bent so far that the geometric axis penetrates the skin, the user must bend the map axis via XMAPAX and ZMAPAX so that a radius vector swung about the map axis at any station along the vehicle will be single-valued. Care should be taken in using models generated by QUICK in which the map axis is bent. The stock version of the QUICKLOK phase of QUICK returns to the user program radius vectors relative to the map axis rather than the geometric axis. Thus, a bent cone whose map axis follows the centerlines of the cones may appear to the user program as not being bent at all. Input follows the same rules as for XSID and ZSID above. Note that both ZSID and ZMAPAX may be negative. These variables are used only if TYPE = 'ASYM', for which case ZSID defaults to zero if not input, and ZMAPAX defaults to ZSID if not input. These input variables may be included in any one of the three namelists.

Input for the slicing planes as well as the processing of the data by KWIKNOSE is completely independent of the main body lines. Any combination of the three slices may be used regardless of the symmetry aside from the slices (the value of TYPE has nothing to do with symmetry of the slices). The minimum slicing plane input, if present at all, is the coordinates of the beginning and end of one segment. At the user's option, different combinations of X and Z (or Y) coordinates and slope angle may be input. Note that TUSL(I) is the slope angle of the segment ending at XUSL(I), ZUSL(I), for $I > 1$. Thus, $TUSL(1) = TUSL(2)$, and similarly for the other two slices. Redundant data will be checked for consistency; but if any one of the three is not input for a given segment, it will be computed. The slicing planes need not cover the entire length of the vehicle, but the first and last segments will be extrapolated to the extremities if not. Nontangent intersections of adjacent segments may be filleted or rounded by inputting the fillet/round radius as indicated in Table 1. Note that RHOUS(I) fillets the intersection at XUSL(I), which is the end of the I-1st segment; otherwise, all the rules of body line filleting/rounding apply.

The remaining input variables are program control integers. IPRINT set different than zero will cause KWIKNOSE to add cards to the QUICK input deck to exercise the math model and generate printouts of coordinate and slope data at various points along each body line segment. IPRINT=0 (the default) suppresses such printout. The variable IR is the unit number (Data Set Reference Number) of the facility card reader, and IW is the unit number of the printer. IF is the unit number of the KWIKNOSE output file of card images to be passed to and read by the QUICKDEF phase of QUICK. IP is the unit number for the facility card punch. If IF and IP are different, the data from unit IR is copied as card images with an 80-character array to IF, from which the input cards are then printed on IW. IF is then rewound and read as namelist data.

3.2 OUTPUT

Table 6 is the KWIKNOSE printout for the sample case. The printout is composed of three sections: the input data, the results of the computations, and the QUICK input file. The first section, labeled "INPUT CARDS" is a listing of the data cards exactly as punched. The data between the first two lines of periods is the input data after the first namelist has been read. Notice that geometric variables not input through the namelist default to 1.E70. This defaulting is critical to the program's logic and should not be trampled with by inputting user-chosen default values which have no geometric meaning. The printout from this point to the third line of periods contains the results of the geometric computations. First, the number of segments exclusive of rounds and fillets is printed followed by the code's determination as to the input options the user has chosen for each segment. Next, the results of the computation of segment intersections and equations are printed. In the printout labeled "SEGMENT END POINT COORDINATES AND SLOPES", X and Z are the coordinates at the beginning of the segment, except for the last point (with values less than 1.E70), which is the end of the last segment. TA and TF are the slope angles in degrees at the aft and fore ends, respectively, of the segment. The segment equations in the next section of the printout are defined in terms of the center coordinates and radius for ogival sections, and slope angle and Z-intercept for straight-line segments. The X value printed here is the coordinate of the end of the segment. In these terms the line and ogive equations are

$$\begin{aligned} \text{line: } Z &= X \tan T + B \\ \text{ogive: } (X-H)^2 + (Z-K)^2 &= R^2 \end{aligned}$$

Next, the results of the fillet and round computation is printed. The equation of the fillet is defined as

$$\text{fillet: } (X-KSI)^2 + (Z-ETA)^2 = RHO^2$$

and straddles the printed X value. X1, Z1, T1, and X2, Z2, T2 are the coordinates and slope angle (degrees) at the fore and aft intersections, respectively, that is, the points of tangency with the intersected segments. The fillet/round intersections are then printed as additional segments included with the previous unfilleted body line under the heading "FINAL QUICK ARRAY". Note that the original intersection point has been removed from the arrays. This completes the geometric computation for the upper body line (ZUCL). When TYPE = 'ASYM' in the namelist input, this computation and printout are repeated first for the lower body line (ZLCL) and then for the side (YSID). The third printout includes the results of the geometrical calculation for slices. The section of the printout headed "SLICE GEOMETRY" gives the slope angle for each segment ending at the indicated coordinate values. These results are exclusive of rounds or fillets, for which immediately follows the computational results. BI and BJ are the Z-intercepts of the two lines formed by the locus of the fillet/round center while rolling on the two intersecting segments. A3 and A4 are the Z and X coordinates of the fillet/round center; and X1, Z1, T1, etc., are the tangency points as described for the main body line fillet/round printout. The final portion of the computational printout is the slicing body lines with the rounds and fillets inserted. The KWIKNOSE printout concludes with a listing of the card images of the QUICK input file. This listing will not be generated if IF = IP. The contents of this output deck may be understood from Table 7 and Refs. 1 through 3 and will not be discussed here. The user should understand that this deck comprises the input to the QUICKDEF phase of QUICK and not the QUICKLOK phase, which is the portion of QUICK employed in the fluid dynamics programs to obtain geometric information.

3.3 PROCESSING INFORMATION

KWIKNOSE is a Fortran IV program containing about 2,000 cards (Table 8). The code was developed on an IBM 370/165 computer system

under the H Level 21.7 compiler. Compilation of the entire deck requires 230 K bytes of core and 40 sec of central processing unit (CPU) time. Execution of the code requires 80 K bytes of core, and all geometries processed to date have required less than 2 sec of CPU time. Users should note that because of the character manipulation performed in the program, it cannot be compiled under the G level 21 compiler. Execution requires one peripheral storage device for unit IF (defaulted to unit 1). This should be a sequential file formatted for 80-character records as illustrated in Table 5.

3.4 PROGRAMMING CONSIDERATIONS

The overall flow of logic in KWIKNOSE is basically simple and can be best understood via the series of CALL statements in the main program (Table 8) and the subroutine description of Table 9. Execution begins with the initialization of certain data (subroutine INIT), after which the first namelist is read (INPT). Next, the input option chosen by the user is determined for each segment (CASE) in terms of the value of ICASE(I). This variable is then used to control branching in determining the segment equations (SEGE) and locating the segment intersections (INTR). This completed, the results of the computation to this point are copied into the QUICK arrays XQ, ZQ, and TQ (QARY) and printed (PRNT). Next, the round/fillet computations are performed and added to the QUICK arrays (ROFL). If slices are present (indicated by ISL $\neq$ 0), the necessary computations are performed and the results assembled in the QUICK slicing array V (SLCE). If rounds or fillets are present on the slices, these computations are performed and the results inserted into the original arrays (RFSL). If the case is symmetrical, the QUICK input card image file is generated (QUCK) and printed (LIST). If the case is asymmetrical, the completed body line with rounds and fillets is copied into the three-dimensional QUICK array VB (COPY). When all three main body lines are complete, the slicing with optional rounds/filletts is performed and copied into the three-dimensional arrays. Next, ZSID and

ZMAPAX are defaulted if not input, and the remaining body lines are checked for validity (BLCK). Finally, the QUICK input file for the asymmetrical case is generated (QK3D) and listed (LIST), thus completing the processing.

For a more detailed examination of the code, the variable description in Table 10 may be useful. Users interested in the intersection computation should find the logic as coded in SEGE and INTR fairly straightforward, but a few clarifying comments about the rounding and filleting logic are necessary. The straightforward approach to locating the center and tangency points of a circle filleting two other circles is to use analytic geometry, by writing four circle equations and two tangency equations in terms of six unknowns. However, this is equivalent to a polynomial of degree 48 for one unknown, which is an absurd thing to try to solve. The approach used in ROFL is to write the equation of the locus of fillet centers as the fillet rolls on the two adjacent segments, where the intersection of these two loci of centers is the center of the filleting circle. This approach reduces the problem to second degree but requires additional logic to decide when to add or subtract the fillet radius to or from the segment radius (ogive) or Z-intercept (line). Fortunately, the information necessary to make this decision can be extracted from the slopes of the adjacent segments at the intersection point. For the line-ogive intersection, the final second degree problem can then be solved by choosing the set of tangency points which straddle the intersection point. For the ogive-ogive intersection, the filleting circle is chosen whose center is closest to the original intersection point. The advantages of using the loci-of-centers approach are clear; but the major disadvantage is that segments which do not intersect cannot be filleted, whereas the general 48th degree problem includes this case.

Because of the complexity of the above logic as coded and to the many input options available, KWIKNOSE contains many small blocks of

coding to check on the validity of the input and the running computation. When a violation of the rules is detected, a condition code is generated with the Fortran statement STOP xxxx, where xxxx is the four-digit condition code. Execution is halted and an error message is usually generated. A program-generated condition code without an accompanying error message would indicate either a programming error or computer error (unlikely). The condition code is usually the same as the statement number of the format that generated the accompanying message. For five-digit statement numbers, the left most digit is truncated to obtain the condition code.

The accuracy of the code's geometry computation has been checked by comparison with manual computation on an electronic calculator. To the best of the author's knowledge, each block of coding has been checked against a tailored check case. However, the user is encouraged to carefully scrutinize the final QUICK math model for accuracy. The author has, to date, constructed about 30 manual card input decks to QUICK and hundreds of PREKWIK and KWIKNOSE decks, and no errors have been traced to QUICK. Thus, errors in the final math model will most likely be caused by either problems in the setup codes or their input decks. Checking of the QUICK math model can be effectively accomplished if a plotting device is available with stock software. With very little effort, the author has constructed a cathode ray tube (CRT) plotting program using the QUICKLOK portion of QUICK and has found it indispensable in debugging the geometry models. This approach is highly recommended to all users.

4.0 SUMMARY

A user's manual has been presented for a Fortran computer program KWIKNOSE which, for certain axisymmetric and asymmetric nose configurations, provides simplified geometric input to the Grumman QUICK-geometry code, which in turn provides a geometric math model to various numerical flow-field programs. A discussion was presented as to how KWIKNOSE may easily be used to model any vehicle whose upper, lower, and side body lines can be described in terms of sequential segments of circles and straight lines and whose upper and lower half cross sections can be modeled with ellipses. It was further shown how the body may be sliced on three sides. A description was given of the input and output, and a sample case was presented to illustrate use of the code.

REFERENCES

1. Vachris, A. and Yaeger, L. "QUICK-GEOMETRY User's Manual." Grumman Aerospace/Aerodynamic Section Technical Data Report No. 393-74-1, 1974.
2. Vachris, A. F. and Yaeger, L. S. "QUICK-GEOMETRY: A Rapid Response Method for Mathematically Modeling Configuration Geometry." NASA SP-390, 1975, pp. 49-73.
3. Marconi, F. and Yaeger, L. "Development of a Computer Code for Calculating the Steady Super/Hypersonic Inviscid Flow Around Real Configurations, Volume II - Code Description." NASA CR-2676, May 1976.
4. Marconi, F., Salas, M., and Yaeger, L. "Development of a Computer Code for Calculating the Steady Super/Hypersonic Inviscid Flow Around Real Configurations, Volume I - Computational Techniques." NASA CR-2675, April 1976.

5. Kutler, P., Reinhardt, W. A., and Warming, R. F. "Numerical Computation of Multishocked, Three-Dimensional, Supersonic Flow Fields with Real Gas Effects." AIAA Paper No. 72-702, June 1972.

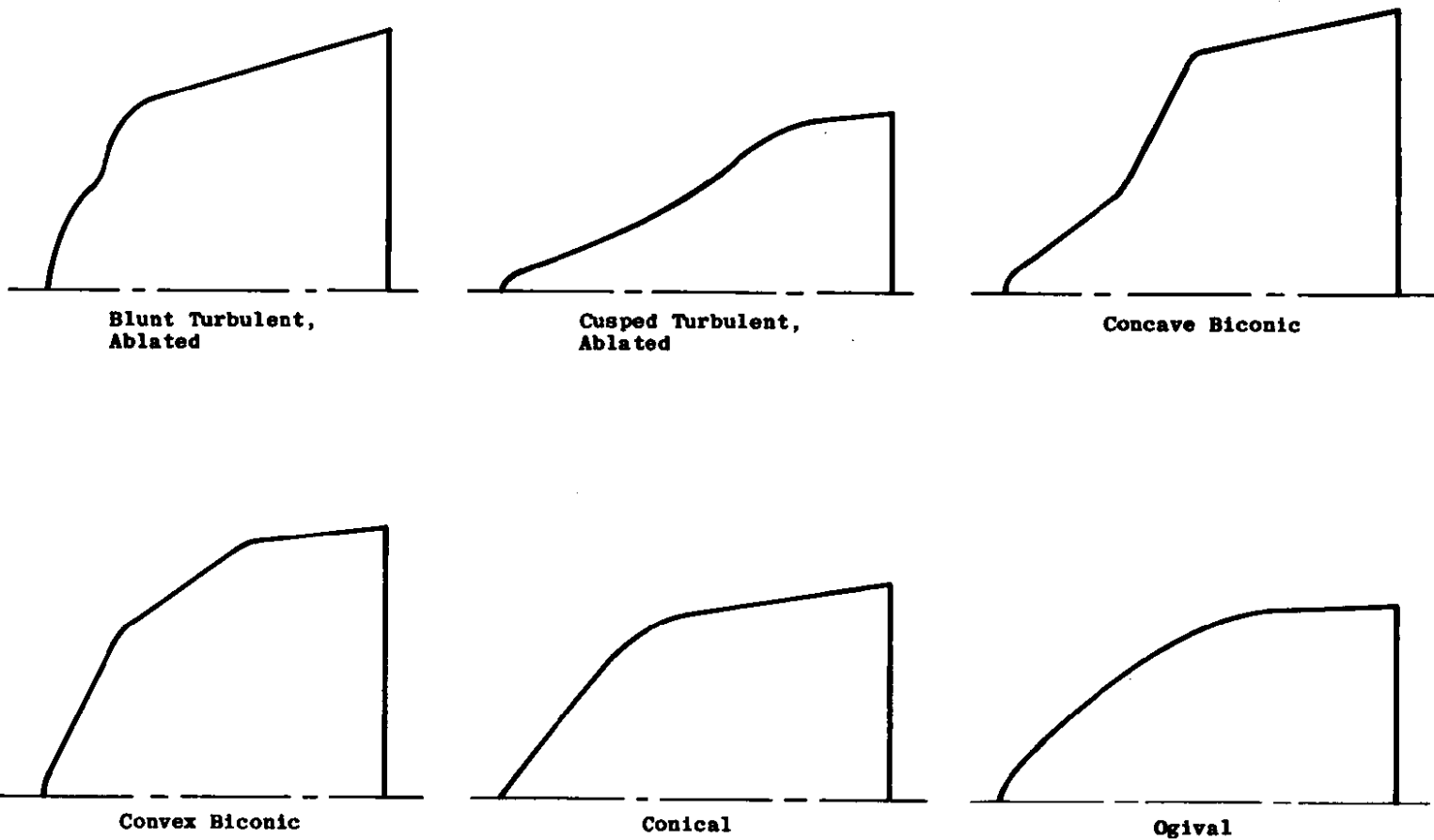


Figure 1. Typical geometries available via KWIKNOSE.

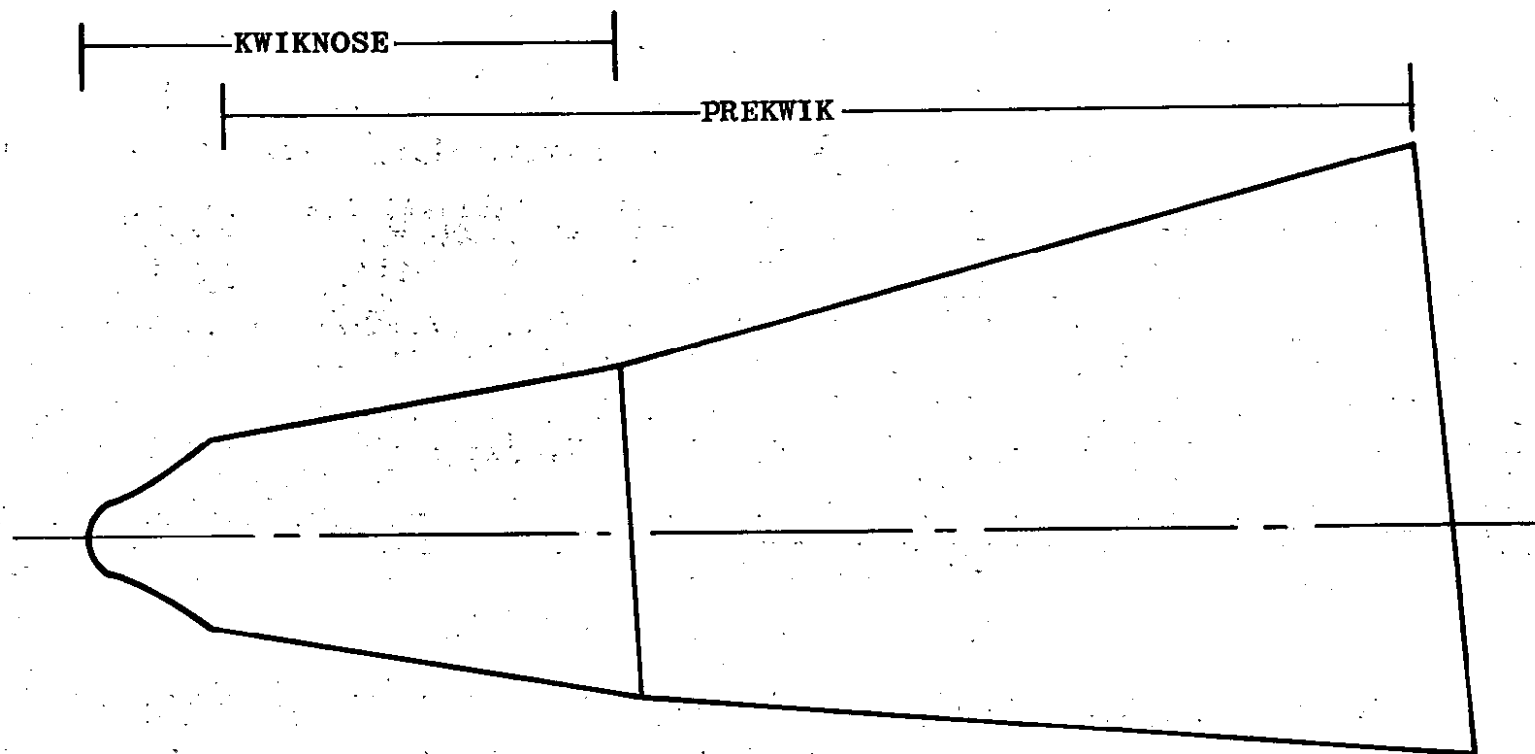


Figure 2. Relative regions of applicability of programs KWIKNOSE and PREKWIK.

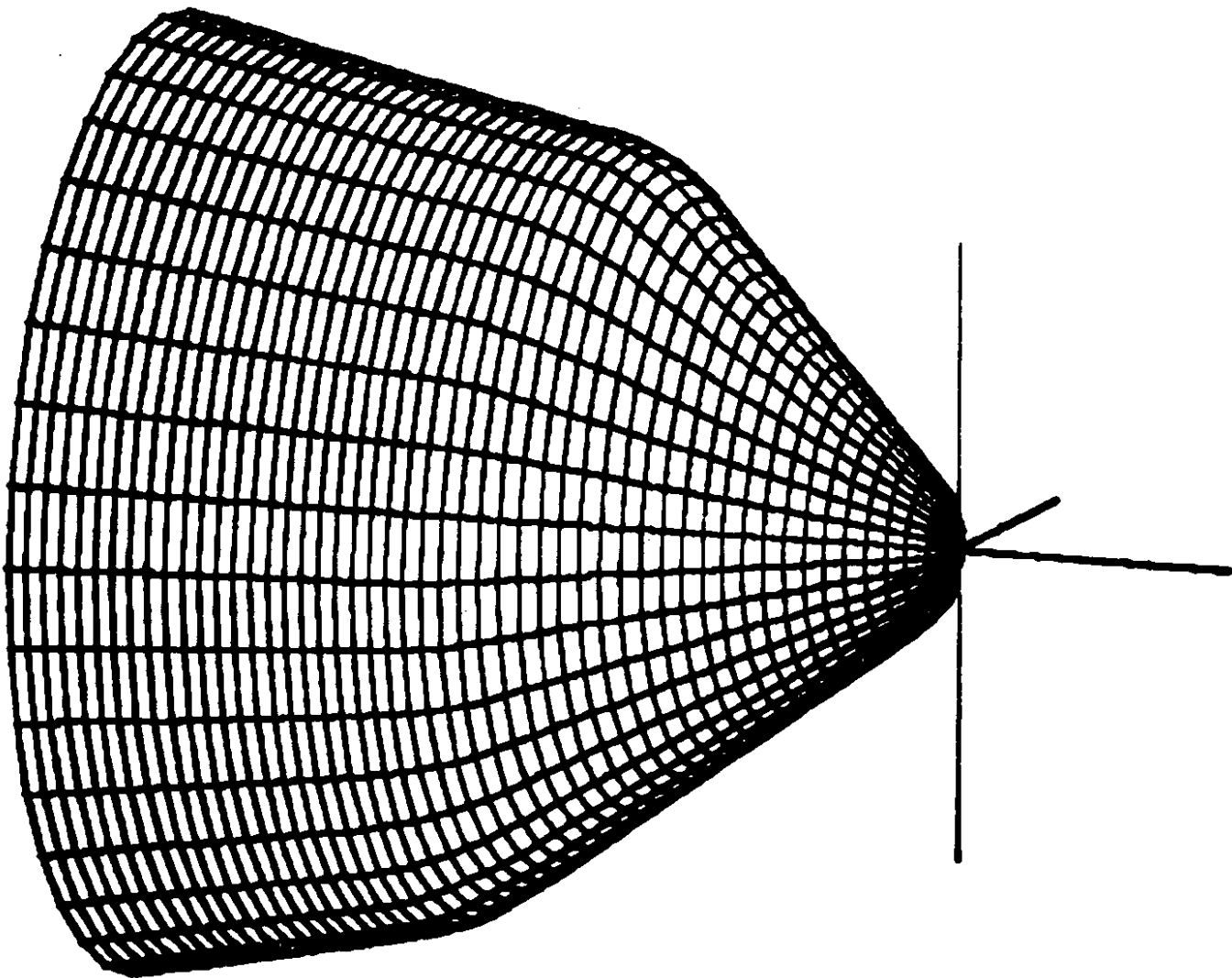


Figure 3. Oblique view of asymmetrically ablated nose.

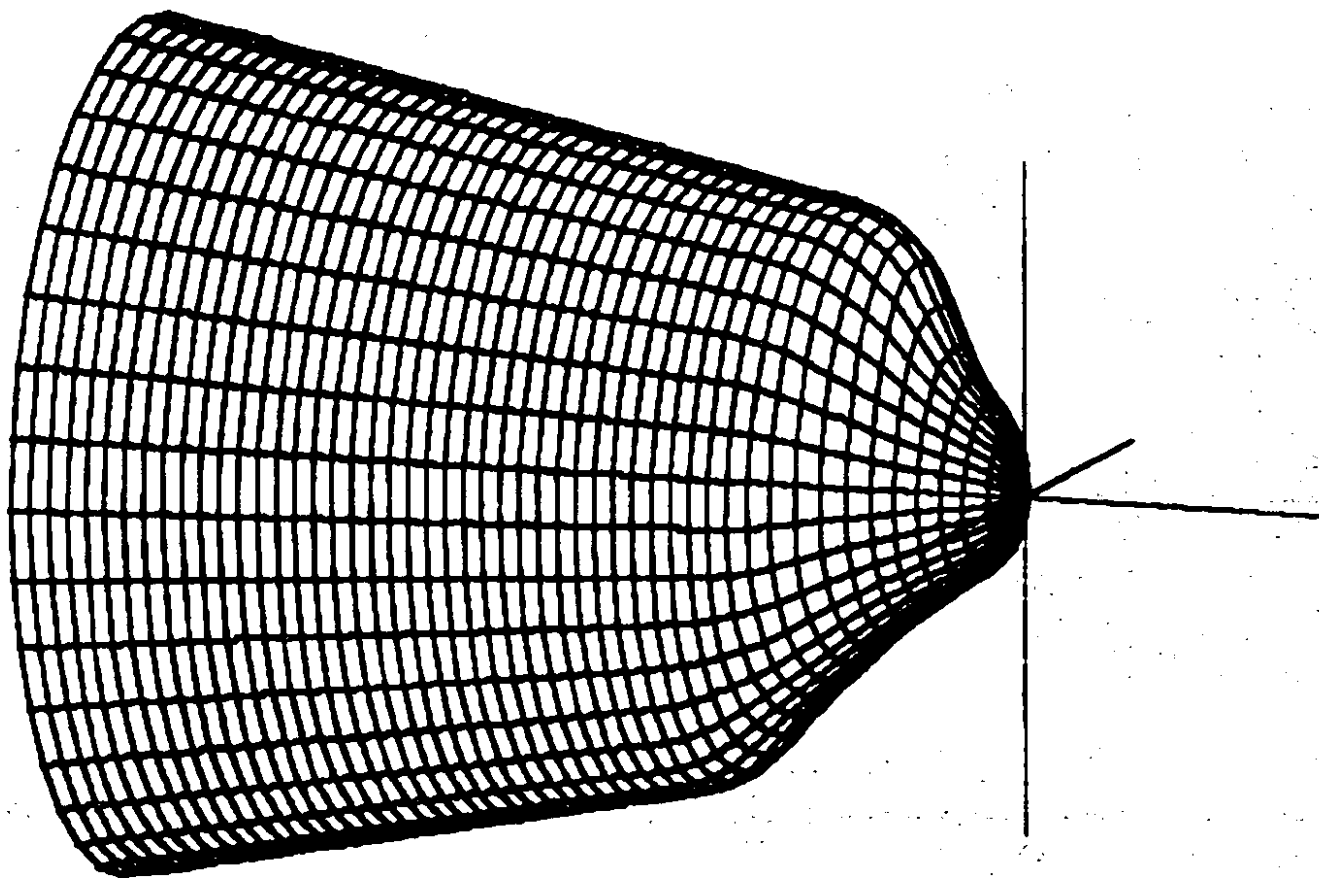


Figure 4. Oblique view of asymmetrical nose from sample case.

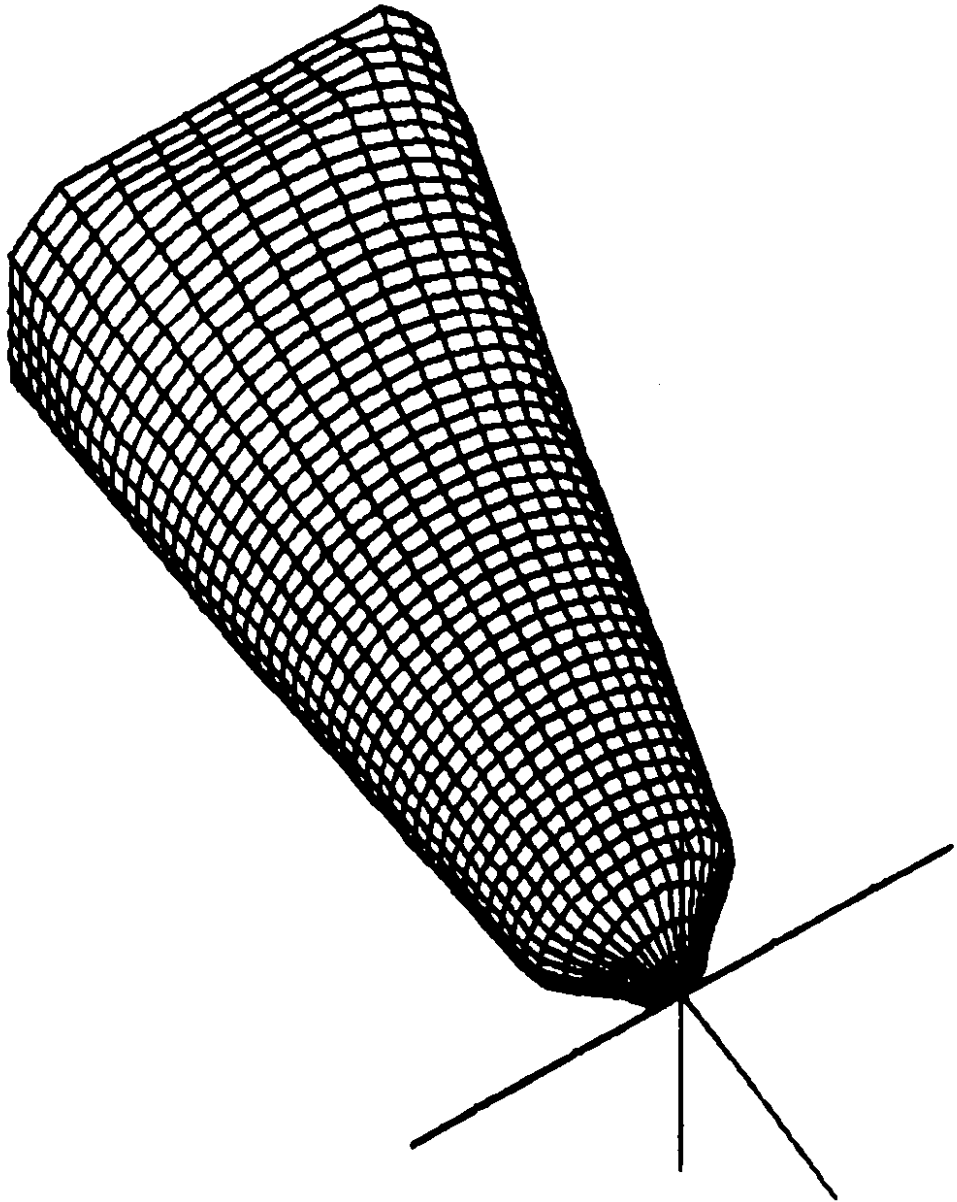


Figure 5. Oblique view of asymmetrically ablated, bent, and sliced cone (sample case).

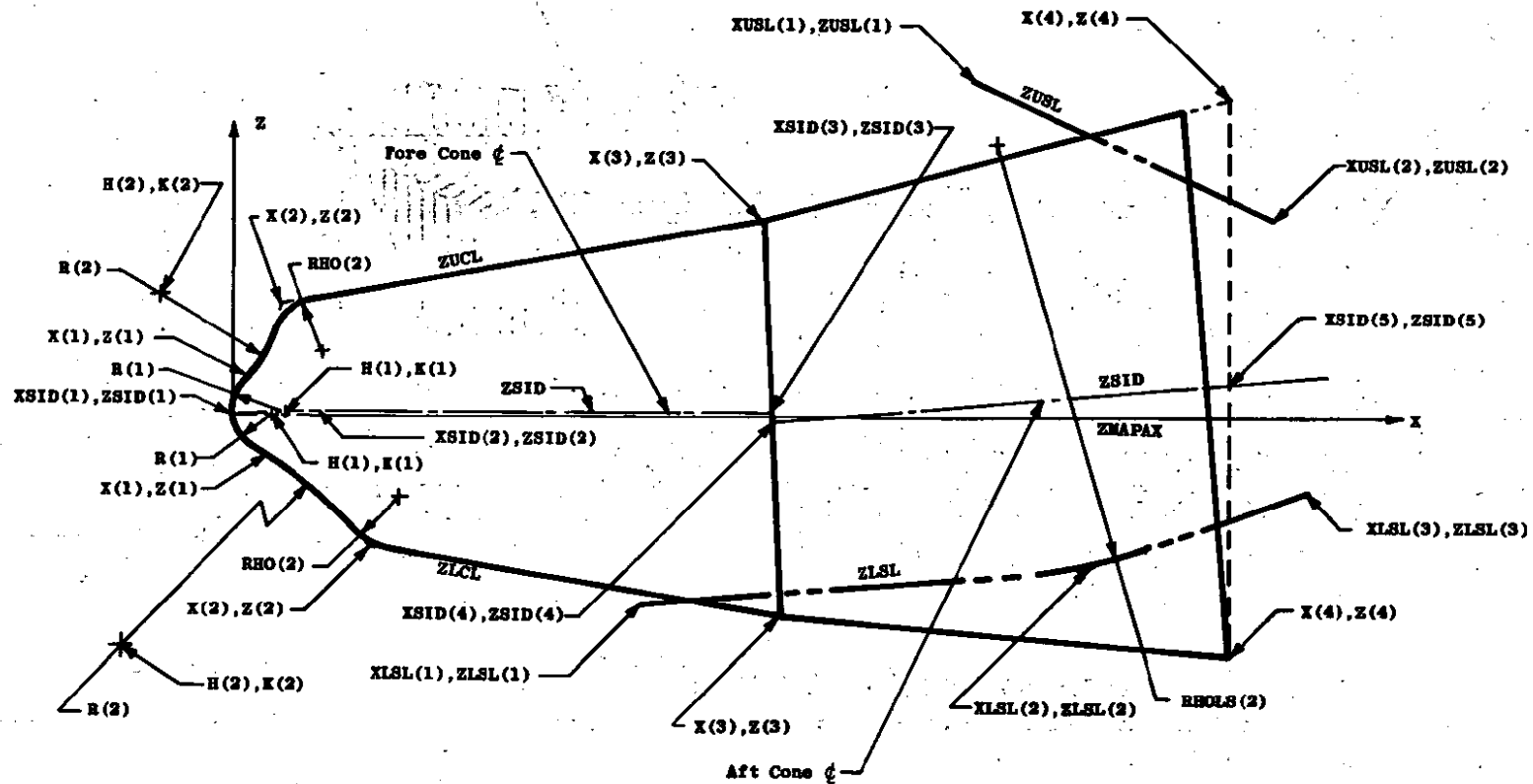


Figure 6. Nose-afterbody sample case with notation for input variables.

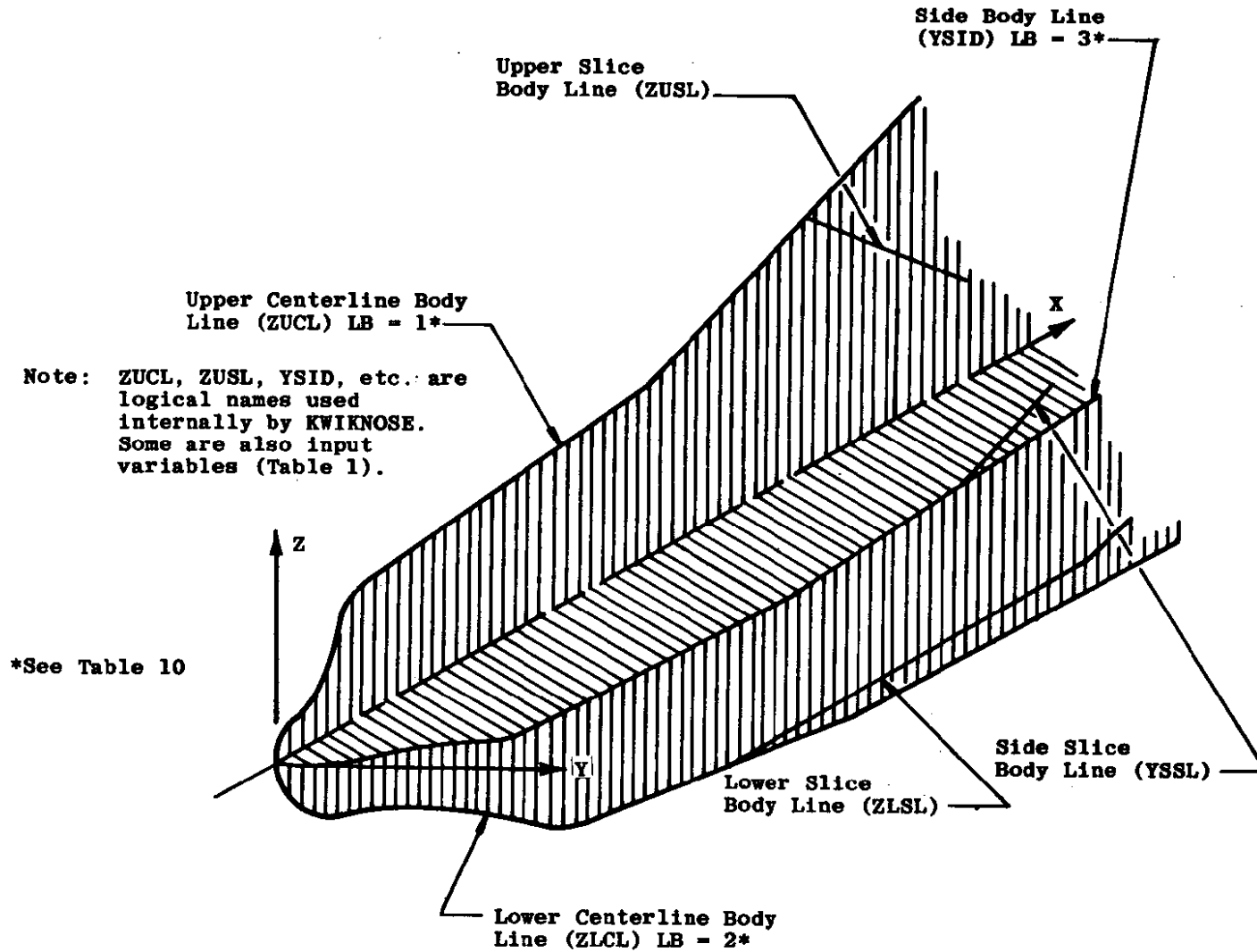


Figure 7. Body line notation.

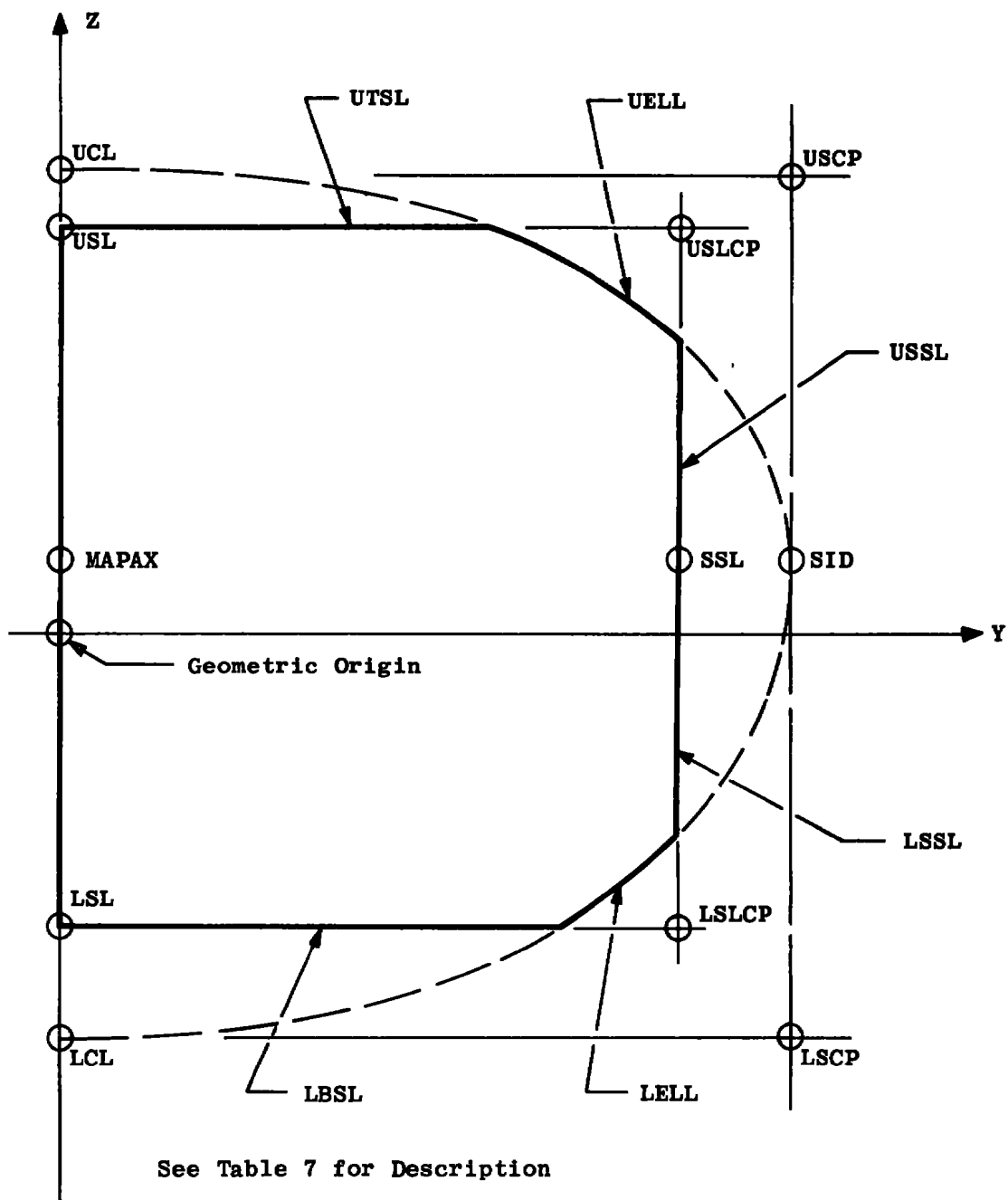


Figure 8. Cross section and body line models.

Table 1. Input to KWIKNOSE via Namelist/INPUT/

Namelist Variable	Value	Default Value	Meaning	TYPE ⁷
X(I)	(0,∞)	1.E70	Centerline distance to end of Ith segment ($I \leq 10$)	R+4
Z(I)	[0,∞) ⁹	1.E70	Radial distance (normal to centerline) to end of Ith segment	R+4
TH(I) ²	(-90,90)	1.E70	Slope angle (deg) at end of Ith segment, positive away from centerline	R+4
CURVE(I)	'LINE'	'bbbb' ¹	Segment I is a straight line	R+4
	'OGIVE'		Segment I is an ogive	
R(I) ³	(0,∞)	1.E70	Radius of ogive of Ith segment (CURVE(I)='OGIVE')	R+4
H(I) ³	(-∞,∞)	1.E70	X coordinate of center of ogive (CURVE(I)='OGIVE')	R+4
K(I) ³	(-∞,∞)	1.E70	Z coordinate of center of ogive (CURVE(I)='OGIVE')	R+4
B(I) ²	(-∞,∞)	1.E70	Z intercept of line segment (CURVE='LINE')	R+4
DIR(I) ⁴	See Table 3	'OPDN'	Direction indicator for CURVE(I)='OGIVE' when aft intersection with adjacent segment may have two solutions	R+4
SLOPE(I)	'T'	'F'	Indicates ogive is to be made tangent to aft end of previous segment (do not use for segment I=1)	L+1
ERRMAX	>10 ⁻⁷	5.E-5	Error criterion for checking consistency of redundant input data	R+4
RH(I) ^{5,8}	(0,∞)	1.E70	Fillet/round radius at end of Ith segment	R+4
XUSL(I) ⁵	[0,∞)	1.E70	} X and Z coordinates and slope (deg) of upper slicing plane, $I \leq 10$	R+4
ZUSL(I) ⁵	[0,∞)	1.E70		R+4
TUSL(I) ⁵	(-90°, +90°)	1.E70		R+4
XLSL(I) ⁵	[0,∞)	1.E70	} X and Z coordinates and slope (deg) of lower slicing plane, $I \leq 10$	R+4
ZLSL(I) ⁵	[0,∞)	1.E70		R+4
TLSL(I) ⁵	(-90°, +90°)	1.E70		R+4
XSSL(I) ⁵	[0,∞)	1.E70	} X and Y coordinates and slope (deg) of side slicing plane, $I \leq 10$	R+4
YSSL(I) ⁵	[0,∞)	1.E70		R+4
TSSL(I) ⁵	(-90°, 90°)	1.E70		R+4

Table 1. Continued

Namelist Variable	Value	Default Value	Meaning	TYPE ⁷
RH ϕ US(I) ⁵	(0, ∞)	1.E70	Fillet/round radius on the upper, lower, and side slices, respectively. RH ϕ —(I) fillets or rounds the <u>beginning</u> of segment I and the end of segment I-1, 2<I<10.	R+4
RH ϕ LS(I) ⁵	(0, ∞)	1.E70		R+4
RH ϕ SS(I) ⁵	(0, ∞)	1.E70		R+4
XSID(I) ⁶	[0, ∞)		X and Z coordinates of side surface body line. Input only if body is asymmetrical and side surface	R+4
ZSID(I)	($-\infty$, ∞)		body line is not in some horizontal plane as the geometric X axis. Defaults to X axis (see Fig. 8).	R+4
XMAPAX(I)	[0, ∞)		X and Z coordinates of QUICK map axis (the axis relative to which the vehicle must be describable in	R+4
ZMAPAX(I)	($-\infty$, ∞)		terms of a single-valued radius vector). Input only if body is asymmetrical and single-valued radius	R+4
			requirement cannot be met by default. Default to XSID, ZSID.	
TYPE	'SYMM'	'SYMM'	Vehicle is axially symmetric. Provide input for one body line.	R+4
	'ASYM'		Vehicle is asymmetric. Provide input for three body lines:	
			UCL, LCL, and SID (see Fig. 8). Input the needed combinations of CURVE, X, Z, TH, R, H, K, B,	
			DIR, and SLOPE in three successive &INPUT namelists. Provide slicing data, XSID, ZSID, XMAPAX,	
			ZMAPAX with any one of three if required. First of three namelists is upper centerline, second is	
			lower centerline, third is side.	
IR ¹⁰	[1,99]	5	Unit number for namelist input	I+4
IW ¹⁰	[1,99]	6	Unit number for printed output	I+4
IF ¹¹	[1,99]	1	Unit number for output of QUICK input data	I+4
	$\neq$ IP		Unit IF will be rewound and printed on unit IW	
	-IP		Unit IF will not be rewound or printed. Punch QUICK input data	
IP	[1,99]	7	Unit number for card punch (need not be dummied if not used)	I+4
IPRINT	$\neq$ 0	0	Cards will be appended to QUICK input (unit IF) to exercise QUICK math model for checking	I+4
	-0		No exercise cards generated	

Table 1. Concluded

Notes:

¹'bbbb' - four blank characters.

²TH(I) and B(I) constitute segment equation for CURVE(I) -'LINE', i.e., $Z(I) = X(I) * \tan(TH(I)) + B(I)$.

³R(I), H(I), and K(I) constitute segment equation for CURVE(I) -'φGIV', i.e., $(X(I)-H(I))^{**2} + (Z(I)-K(I))^{**2} = R(I)^{**2}$.

⁴See Tables 3 and 4 for complete explanation.

⁵Do not input these variables unless rounds, fillets, or slices are present.

⁶XSID(1) and ZSID(1) must both be input as zero. If input at all, at least two values of each must be input.

⁷Variable type: L - logical, I - integer, R - real. Number following * is number of bytes.

⁸When defining a series of segments via X, Z, TH, R, H, K, and B, do not skip an I value when RHφ(I) is given. RHφ(I) will fillet/round the intersection of the Ith and I+1st segments. For example, to fillet/round two line segments, input X(1), Z(1), X(2), Z(2), and RHφ(1). Note further that each fillet/round reduces by one the total number of allowable segments: ten segments with two fillets will overrun QUICK's dimensions, whereas eight segments with two fillets will not. Though fillets/rounds are not treated as segments in the namelist input, they are treated as separate segments in the output after their intersections with adjacent segments have been computed and inserted into the XQ, ZQ, and TQ arrays.

⁹May be negative when TYPE -'ASYM' and if map axis is bent.

¹⁰If the unit numbers are to be changed, they must also be changed in subroutine INIT because data are read from IR and written on IW before the namelist data are read as namelists.

¹¹If IF were set equal to IP(07) in the namelist, the input data would still be read from unit IR(05) copied to unit 01.

Note: Character "φ" means capital letter O

"0" means numeric zero 0

| | means end points are included; () means end points are excluded.

Table 2. KWIKNOSE Input Options¹ for Three Main Body Lines (ZUCL, ZLCL, YSID)

CURVE(I)	X(I)	Z(I)	H(I)	K(I)	R(I)	SLOPE(I)	TH(I)	B(I)	DIR(I) ³	May Follow Case 5, 9, 19?	ICASE(I) ²	ΣL ²
'ØGIV'	X	X	---	---	X	---	---	---	X	No	1	582
	X	X	---	---	---	X	---	---	---	No	2 ⁶	774
	X	---	---	---	X	X	---	---	X	No	3 ⁶	834
	X	X	---	---	---	---	X	---	---	No	4	526
	---	---	X	X	X	---	---	---	X	Yes	5 ⁴	624
	X	X	---	---	X	---	X	---	X	Yes	10 ⁵	590
	X	---	X	X	X	---	---	---	X	Yes	11 ⁵	626
	---	X	X	X	X	---	---	---	X	Yes	12 ⁵	628
	---	---	X	X	X	---	X	---	X	Yes	13 ⁵	632
	X	X	X	X	X	---	X	---	---	Yes	18 ^{5,7}	638
	---	---	---	---	X	X	---	---	X	No	19 ⁶	832
	X	X	---	---	X	X	---	---	X	Yes	21 ^{5,6}	838
	'LINE'	X	X	---	---	---	---	---	---	No	6	6
		X	---	---	---	---	X	---	---	No	7	10
		---	X	---	---	---	X	---	---	No	8	12
		---	---	---	---	---	X	X	X	Yes	9 ⁴	136
		X	X	---	---	---	X	---	---	Yes	14 ⁵	14
		X	---	---	---	---	X	X	---	Yes	15 ⁵	138
		---	X	---	---	---	X	X	---	Yes	16 ⁵	140
		X	X	---	---	---	X	X	---	Yes	17 ^{5,7}	142
		X	X	---	---	---	X	---	X	No	20 ⁶	262

L:	512 ⁸	2 ⁹	4	16	32	64	256	8	128
----	------------------	----------------	---	----	----	----	-----	---	-----

¹One of these options must be chosen for each segment. Insufficient or redundant data will cause execution to be halted.

²ICASE(I) and L are not input variables but are used for program control. See subroutines CASE, SEGE, and INTR in Table 8.

³DIR(I) is used only when multiple solutions or intersections can occur. When multiple roots are equal, DIR(I) is ignored. See Tables 3 and 4.

⁴These options are not acceptable for last segment of vehicle.

^{4,5}Use one of these options if previous segment was given via ICASE=5,9, or 19.

⁶These options are not acceptable for first segment of vehicle.

⁷These options contain redundant data, which if not self consistent to within ERRMAX, will cause execution to halt.

⁸CURVE(I)=ØGIV', L=512; ='LINE'; L=0.

⁹If X(I) < 1.E70, L=2; IF X(I) ≥ 1.E70 (default), L=0. Same for remaining variables but with higher powers of 2.

Table 3. Input Description for Variable DIR(I)¹

ICASE(I)	ICASE(I+1)	ICASE(I-1)	DIR(I)	Meaning
1, 3, 10, 11, 13	—	—	' ϕ PUP' ² ' ϕ PDN' ²	Ogive opens up Ogive opens down
5	5	—	'F ϕ RE' 'AFTE' 'T ϕ PS' 'B ϕ TS'	Use fore solution Use aft solution Use top solution Use bottom solution } Use when X coordinate of fore and aft solutions are equal
5 9	9 5	— — }	'F ϕ RE' 'AFTE'	Use fore solution Use aft solution
12	—	—	'F ϕ RE' 'AFTE'	Use fore solution Use aft solution
20	—	5	'T ϕ PS' 'B ϕ TS'	Use top solution Use bottom solution
21 5	— 21	5 }	' ϕ PUP' ² ' ϕ PDN' ²	Ogive opens up Ogive opens down
21	—	9	—	DIR(I) not needed
19	—	—	'UPF ϕ ' 'UPAF' 'UPT ϕ ' 'UPB ϕ ' 'DNF ϕ ' 'DNAF' 'DNT ϕ ' 'DNB ϕ '	Ogive opens up, use fore solution Ogive opens up, use aft solution Ogive opens up, use top solution Ogive opens up, use bottom solution Ogive opens down, use fore solution Ogive opens down, use aft solution Ogive opens down, use top solution Ogive opens down, use bottom solution

For DIR(I) notation, assume
'UP'=' ϕ PUP', 'DN'=' ϕ PDN',
'F ϕ '='F ϕ RE', 'AF'='AFTE',
'T ϕ '='T ϕ PS', and 'B ϕ '='B ϕ TS'
and use Table 4.

¹For the ICASE(I) value, one of the indicated DIR(I) values must be chosen. See Table 4 for clarification.

²Except for the noted values, DIR(I) refers to the aft intersection of the Ith segment.

Table 4. Nomenclature for Input Variable DIR(I)

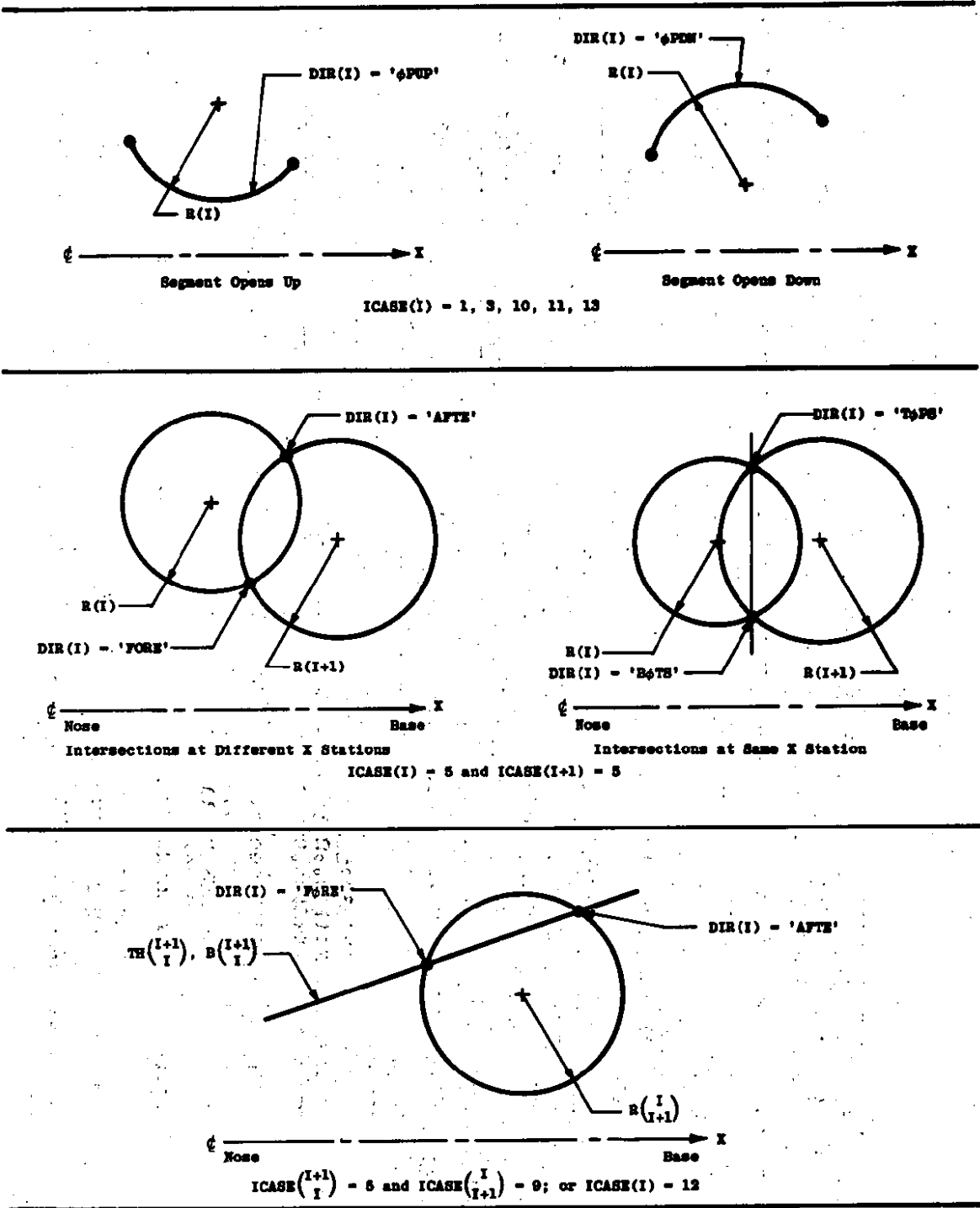


Table 5. Jobstream for Sample Case

```

/*PRIORITY      5
//VKF05333      JOB          (XXX,
//   VRV00146,01,PROJ=NBR), '00000 USER NAME',
//   MSGLEVEL=(2,0),
//   REGION=160K,
//   CLASS=X,
//   TIME=(,15)
// EXEC FORTEPOS,PGMNO=VRV00146.
//GO.FT01F001 DD UNIT=WORK,DISP=(NEW,PASS,DELETE),DSN=66KWIKIN,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),SPACE=(CYL,1)
//GO.FT05F001 DD *
HYPOTHETICAL ASYMMETRICALLY ABLATED, SLICED, BENT CONE
&INPUT
TYPE='ASYM',IPRINT=1,
CURVE=2*'OGIV',2*'LINE',DIR(2)='AFTE',
H(1)=.5,K(1)=0.,R(1)=.5,TH(1)=45.,H(2)=-.64905,K(2)=1.14905,R(2)=1.125,
X(3)=4.92,Z(3)=1.84,TH(3)=10.,RHO(2)=.5,X(4)=9.22,Z(4)=2.92218,TH(4)=15.,
&END
&INPUT
CURVE=2*'OGIV',2*'LINE',DIR(2)='AFTE',
H(1)=.375,K(1)=0.,R(1)=.375,TH(1)=30.,H(2)=-1.8125,K(2)=3.78886,R(2)=4.,
X(3)=5.08242,Z(3)=1.88,TH(3)=10.,RHO(2)=.5,X(4)=9.22,Z(4)=2.24199,TH(4)=5.,
&END
&INPUT
CURVE=2*'OGIV',3*'LINE',DIR='OPDN','UPAF',
H(1)=.4375,K(1)=0.,R(1)=.4375,TH(1)=36.325,SLOPE(2)='T',
R(2)=2.5625,X(3)=4.99909,Z(3)=1.8253,RHO(2)=.5,X(4)=5.00333,Z(4)=1.8253,
X(5)=9.22,Z(5)=2.57165,TH(3)=10.,
XSIO=0.,.85535,4.99909,5.00333,9.22,ZSID=0.,.02866,.02866,-.06865,.30026,
XMAPAX=0.,9.22,ZMAPAX=0.,0.,
XUSL=6.86,9.65,ZUSL=3.11,1.83,
XLSL=3.79,7.97,9.94,ZLSL=1.75,1.4,.7,RHOLS(2)=4.,
XSSL=0.,9.22,YSSL=2.3,2.3,
&END
// EXEC FORTEPOS,PGMNO=VRV00099,COND=(8,LE)
//GO.FT05F001 DD DSN=66KWIKIN,DISP=(OLD,DELETE)
//GO.FT07F001 DD UNIT=3330,VOL=SER=666666,DSN=JCA07115,
//   DCB=(RECFM=U,BLKSIZE=80),DISP=(OLD,KEEP)
/*

```

Table 6. KWIKNOSE Printout for Sample Case

INPUT CARDS

```

HYPOTHETICAL ASYMMETRICALLY ABLATED, SLICED, BENT CONE
&INPUT
TYPE='ASYM',IPRINT=1,
CURVE=2*'OGIV',2*'LINE',DIR(2)='AFTE',
H(1)=.5,K(1)=0.,R(1)=.5,TH(1)=45.,H(2)=-.64905,K(2)=1.14905,R(2)=1.125,
X(3)=4.92,Z(3)=1.84,TH(3)=10.,RHO(2)=.5,X(4)=9.22,Z(4)=2.92218,TH(4)=15.,
&END
&INPUT
CURVE=2*'OGIV',2*'LINE',DIR(2)='AFTE',
H(1)=.375,K(1)=0.,R(1)=.375,TH(1)=30.,H(2)=-1.8125,K(2)=3.78886,R(2)=4.,
X(3)=5.08242,Z(3)=1.88,TH(3)=10.,RHO(2)=.5,X(4)=9.22,Z(4)=2.24199,TH(4)=5.,
&END
&INPUT
CURVE=2*'OGIV',3*'LINE',DIR='OPDN','UPAF',
H(1)=.4375,K(1)=0.,R(1)=.4375,TH(1)=36.325,SLOPE(2)='T',
R(2)=2.5625,X(3)=4.99909,Z(3)=1.8253,RHO(2)=.5,X(4)=5.00333,Z(4)=1.8253,
X(5)=9.22,Z(5)=2.57165,TH(3)=10.,
XSID=0.,.85535,4.99909,5.00333,9.22,ZSID=0.,.02866,.02866,-.06865,.30026,
XMAPAX=0.,9.22,ZMAPAX=0.,0.,
XUSL=6.86,9.65,ZUSL=3.11,1.83,
XLSL=3.79,7.97,9.94,ZLSL=1.75,1.4,.7,RHOLS(2)=4.,
XSSL=0.,9.22,YSSL=2.3,2.3,
&END

```

INPUT DATA: 1 ZUCL

I	X	Z	TH	H	K	R	B	RHO	CURVE DIR SLOPE
1	9.99999E 69	9.99999E 69	4.50000E 01	5.00000E-01	0.0	5.00000E-01	9.99999E 69	9.99999E 69	OGIV OPDN F
2	9.99999E 69	9.99999E 69	9.99999E 69	-6.49050E-01	1.14905E 00	1.12500E 00	9.99999E 69	5.00000E-01	OGIV AFTE F
3	4.92000E 00	1.84000E 00	1.00000E 01	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	LINE OPDN F
4	9.22000E 00	2.92218E 00	1.50000E 01	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	LINE OPDN F
5	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	OPDN F
6	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	OPDN F
7	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	OPDN F
8	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	OPDN F
9	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	OPDN F
10	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	OPDN F

```

&INPUT1
IR= 1,IR= 6,IF= 1,IP= 7,IPRINT= 1,ERRMAX= 0.49999988E-04
&END

```

Table 6. Continued

.....
 NUMBER OF SEGMENTS NSEQ = 4

ICASE(1) = 13
 ICASE(2) = 5
 ICASE(3) = 14
 ICASE(4) = 14
 ICASE(5) = 0
 ICASE(6) = 0
 ICASE(7) = 0
 ICASE(8) = 0
 ICASE(9) = 0
 ICASE(10) = 0

SEGMENT END POINT COORDINATES AND SLOPES:

	X	Z	TA	TF
1	0.0	0.0	4.499998E 01	8.999899E 01
2	1.464467E-01	3.535535E-01	8.524072E 01	4.500000E 01
3	4.720713E-01	1.055710E 00	9.999997E 00	9.999997E 00
4	4.919999E 00	1.839999E 00	1.412625E 01	1.412625E 01
5	9.219999E 00	2.922179E 00	9.999994E 69	9.999994E 69
6	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
7	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
8	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
9	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
10	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
11	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69

SEGMENT EQUATIONS:

	X	M	K	R	M	T	B
1	1.464467E-01	5.000000E-01	0.0	5.000000E-01	9.999994E 69	4.500000E 01	9.999994E 69
2	4.720713E-01	-6.490500E-01	1.149050E 00	1.125000E 00	9.999994E 69	9.999994E 69	9.999994E 69
3	4.919999E 00	9.999994E 69	9.999994E 69	9.999994E 69	1.763268E-01	1.000000E 01	9.724715E-01
4	9.219999E 00	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	1.500000E 01	4.516907E-01
5	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
6	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
7	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
8	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
9	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
10	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69

FILLET/ROUND EQUATIONS AND INTERSECTIONS:

	X	KSI	ETA	RHO	X1	Z1	T1	X2	Z2	T2
1	1.464467E-01	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
2	4.72071E-01	8.87869E-01	6.21313E-01	5.00000E-01	4.14969E-01	7.83694E-01	7.10488E 01	8.01045E-01	1.11372E 00	1.00000E 01
3	4.92000E 00	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
4	9.22000E 00	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
5	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
6	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
7	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
8	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
9	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
10	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69

Table 6. Continued

FINAL QUICK ARRAY:

	X	Z	TA	TF
1	0.0	0.0	4.499998E 01	8.999899E 01
2	1.464467E-01	3.535535E-01	7.104884E 01	4.500000E 01
3	4.149692E-01	7.836936E-01	1.000000E 01	7.104884E 01
4	8.010449E-01	1.113717E 00	9.999997E 00	1.000000E 01
5	4.919999E 00	1.839999E 00	1.412625E 01	1.412625E 01
6	9.219999E 00	2.922179E 00	9.999994E 69	9.999994E 69
7	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
8	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
9	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
10	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69

INPUT DATA: 2 ZLCL

I	X	Z	TH	H	K	R	B	RHO	CURVE DIR SLOPE
1	9.99999E 69	9.99999E 69	3.00000E 01	3.75000E-01	0.0	3.75000E-01	9.99999E 69	9.99999E 69	OGIV OPDN F
2	9.99999E 69	9.99999E 69	9.99999E 69	-1.81250E 00	3.78886E 00	4.00000E 00	9.99999E 69	5.00000E-01	OGIV AFTE F
3	5.08242E 00	1.88000E 00	1.00000E 01	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	LINE OPDN F
4	9.22000E 00	2.24199E 00	5.00000E 00	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	LINE OPDN F
5	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	OPDN F
6	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	OPDN F
7	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	OPDN F
8	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	OPDN F
9	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	OPDN F
10	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	OPDN F

4INPUT1

IR=

1.1W=

6.1F=

1.1P=

7.1PRINT=

1.1ERRMAX= 0.4999998E-04

6END

NUMBER OF SEGMENTS NSEG = 4

ICASE(1) = 13

ICASE(2) = 5

ICASE(3) = 14

ICASE(4) = 14

ICASE(5) = 0

ICASE(6) = 0

ICASE(7) = 0

ICASE(8) = 0

ICASE(9) = 0

ICASE(10) = 0

SEGMENT END POINT COORDINATES AND SLOPES:

	X	Z	TA	TF
1	0.0	0.0	2.999998E 01	8.999899E 01
2	1.875001E-01	3.247595E-01	4.971005E 01	3.000002E 01
3	1.238627E 00	1.202235E 00	9.999997E 00	9.999997E 00
4	5.082419E 00	1.879999E 00	4.999980E 00	4.999980E 00
5	9.219999E 00	2.241989E 00	9.999994E 69	9.999994E 69
6	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
7	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69

Table 6. Continued

```

8 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69
9 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69
10 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69
11 9.999994E 69 9.999994E 69 9.999994E 69 9.999994E 69

```

SEGMENT EQUATIONS:

	X	M	K	R	M	T	B
1	1.875001E-01	3.750000E-01	0.0	3.750000E-01	9.999994E 69	3.000000E 01	9.999994E 69
2	1.238627E 00	-1.812500E 00	3.788859E 00	4.000000E 00	9.999994E 69	9.999994E 69	9.999994E 69
3	5.082419E 00	9.999994E 69	9.999994E 69	9.999994E 69	1.763268E-01	1.000000E 01	9.838324E-01
4	9.219999E 00	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	5.000000E 00	1.435345E 00
5	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
6	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
7	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
8	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
9	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
10	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69

FILLET/ROUND EQUATIONS AND INTERSECTIONS:

	X	KSI	ETA	RHO	X1	Z1	T1	X2	Z2	T2
1	1.87500E-01	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
2	1.23863E 00	1.49739E 00	7.40149E-01	5.00000E-01	1.12962E 00	1.07890E 00	4.73521E 01	1.41057E 00	1.23255E 00	1.00000E 01
3	5.08242E 00	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
4	9.22000E 00	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
5	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
6	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
7	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
8	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
9	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
10	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69

FINAL QUICK ARRAY:

	X	Z	TA	TF
1	0.0	0.0	2.999998E 01	8.999899E 01
2	1.875001E-01	3.247595E-01	4.735210E 01	3.000002E 01
3	1.129624E 00	1.078896E 00	1.000000E 01	4.735210E 01
4	1.410566E 00	1.232553E 00	9.999997E 00	1.000000E 01
5	5.082419E 00	1.879999E 00	4.999980E 00	4.999980E 00
6	9.219999E 00	2.241989E 00	9.999994E 69	9.999994E 69
7	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
8	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
9	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
10	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69

INPUT DATA: 3 YSID

I	X	Z	TH	M	K	R	B	RHO	CURVE DIR SLOPE
1	9.99999E 69	9.99999E 69	3.63250E 01	4.37500E-01	0.0	4.37500E-01	9.99999E 69	9.99999E 69	OBIV OPDN F
2	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	2.56250E 00	9.99999E 69	5.00000E-01	OBIV UPAF T
3	4.99999E 00	1.82530E 00	1.00000E 01	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	LINE OPDN F
4	5.06333E 00	1.82530E 00	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	LINE OPDN F
5	9.22000E 00	2.57185E 00	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	LINE OPDN F

Table 6. Continued

6	9.99999E 69	0.0	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
7	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
8	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
9	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
10	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69

OPDN
OPDN
OPDN
OPDN
OPDN

L
L
L
L
L

ZUSL	X-SL	Z-SL	T-SL	RHO-S
1	6.86000E 00	3.11000E 00	9.99999E 69	9.99999E 69
2	9.65000E 00	1.83000E 00	9.99999E 69	9.99999E 69
3	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
4	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
5	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
6	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
7	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
8	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
9	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
10	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69

ZLSL	X-SL	Z-SL	T-SL	RHO-S
1	3.79000E 00	1.75000E 00	9.99999E 69	9.99999E 69
2	7.97000E 00	1.40000E 00	9.99999E 69	4.00000E 00
3	9.94000E 00	7.00000E-01	9.99999E 69	9.99999E 69
4	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
5	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
6	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
7	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
8	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
9	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
10	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69

YSSL	X-SL	Z-SL	T-SL	RHO-S
1	0.0	2.30000E 00	9.99999E 69	9.99999E 69
2	9.22000E 00	2.30000E 00	9.99999E 69	9.99999E 69
3	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
4	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
5	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
6	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
7	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
8	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
9	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
10	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69

XMAPAX	ZMAPAX	XSID	ZSID
1	0.0	0.0	0.0
2	9.22000E 00	0.0	8.55350E-01
3	9.99999E 69	9.99999E 69	4.99900E 00
4	9.99999E 69	9.99999E 69	5.00333E 00
5	9.99999E 69	9.99999E 69	9.22000E 00
6	9.99999E 69	9.99999E 69	3.00260E-01
7	9.99999E 69	9.99999E 69	9.99999E 69
8	9.99999E 69	9.99999E 69	9.99999E 69
9	9.99999E 69	9.99999E 69	9.99999E 69
10	9.99999E 69	9.99999E 69	9.99999E 69

Table 6. Continued

&INPUT1
 IR= 1, IW= 0, IF= 1, IP= 7, IPRINT= 1, ERRMAX= 0.49999988E-04
 &END

NUMBER OF SEGMENTS NSEG = 5

ICASE(1) = 13
 ICASE(2) = 19
 ICASE(3) = 14
 ICASE(4) = 6
 ICASE(5) = 6
 ICASE(6) = 0
 ICASE(7) = 0
 ICASE(8) = 0
 ICASE(9) = 0
 ICASE(10) = 0

SEGMENT END POINT COORDINATES AND SLOPES:

	X	Z	TA	TF
1	0.0	0.0	3.632498E 01	8.999899E 01
2	1.783406E-01	3.524806E-01	5.893269E 01	3.632498E 01
3	0.553475E-01	1.094646E 00	9.999994E 00	9.999994E 00
4	4.999089E 00	1.825299E 00	0.0	0.0
5	5.003329E 00	1.825299E 00	1.003739E 01	1.003739E 01
6	9.219999E 00	2.571650E 00	9.999994E 69	9.999994E 69
7	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
8	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
9	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
10	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
11	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69

SEGMENT EQUATIONS:

	X	M	K	R	M	T	B
1	1.783406E-01	4.375000E-01	0.0	4.375000E-01	9.999994E 69	3.632500E 01	9.999994E 69
2	0.553475E-01	-1.339593E 00	2.417009E 00	2.562500E 00	9.999994E 69	9.999994E 69	9.999994E 69
3	4.999089E 00	9.999994E 69	9.999994E 69	9.999994E 69	1.763268E-01	1.000000E 01	9.438258E-01
4	5.003329E 00	9.999994E 69	9.999994E 69	9.999994E 69	0.0	0.0	1.825299E 00
5	9.219999E 00	9.999994E 69	9.999994E 69	9.999994E 69	1.769999E-01	1.003739E 01	9.397106E-01
6	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
7	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
8	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
9	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
10	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69

FILLET/ROUND EQUATIONS AND INTERSECTIONS:

	X	KS1	ETA	RHO	X1	Z1	T1	X2	Z2	T2
1	1.78341E-01	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
2	0.55347E-01	1.15439E 00	6.39662E-01	5.00000E-01	7.47200E-01	9.29842E-01	5.45243E 01	1.06754E 00	1.13207E 00	1.00000E 01
3	4.99909E 00	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
4	5.00333E 00	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
5	9.22000E 00	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
6	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69
7	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69	9.99999E 69

Table 6. Continued

```

8 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69
9 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69
10 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69 9.99999E 69

```

FINAL QUICK ARRAY:

	X	Z	TA	TF
1	0.0	0.0	3.632498E 01	8.999899E 01
2	1.783406E-01	3.524806E-01	5.452428E 01	3.632498E 01
3	7.472076E-01	9.298420E-01	1.000000E 01	5.452428E 01
4	1.067565E 00	1.132066E 00	9.999994E 00	1.000000E 01
5	4.999089E 00	1.825299E 00	0.0	0.0
6	5.003329E 00	1.825299E 00	1.003739E 01	1.003739E 01
7	9.219999E 00	2.571650E 00	9.999994E 69	9.999994E 69
8	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
9	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
10	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69

SLICE GEOMETRY:

	X	Z	T	X	Z	T	X	Y	T
1	0.0	6.257238E 00	-2.464478E 01	0.0	2.067344E 00	-4.786331E 00	0.0	2.299999E 00	0.0
2	9.650000E 00	1.830000E 00	-2.464478E 01	7.969999E 00	1.400000E 00	-4.786331E 00	9.219999E 00	2.299999E 00	0.0
3	9.999994E 69	9.999994E 69	9.999994E 69	9.940000E 00	7.000000E-01	-1.956165E 01	9.999994E 69	9.999994E 69	9.999994E 69
4	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
5	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
6	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
7	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
8	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
9	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
10	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69

SLICE ROUND/FILLET CENTERS AND INTERSECTIONS:

	B1	BJ	A3	A4	X1	Z1	T1	X2	Z2	T2
22	-1.94665E 00	-1.30367E-02	-2.54278E 00	7.11942E 00	7.45318E 00	1.44327E 00	-4.78633E 00	8.45870E 00	1.22635E 00	-1.95616E 01

FINAL QUICK ARRAYS FOR ROUNDED/FILLETED SLICES:

	X	Z	T	X	Z	T	X	Y	T
1	0.0	6.257238E 00	-2.464478E 01	0.0	2.067344E 00	-4.786331E 00	0.0	2.299999E 00	0.0
2	9.650000E 00	1.830000E 00	-2.464478E 01	7.453176E 00	1.443274E 00	-4.786331E 00	9.219999E 00	2.299999E 00	0.0
3	9.999994E 69	9.999994E 69	9.999994E 69	8.458698E 00	1.226351E 00	-1.956165E 01	9.999994E 69	9.999994E 69	9.999994E 69
4	9.999994E 69	9.999994E 69	9.999994E 69	9.940000E 00	7.000000E-01	-1.956165E 01	9.999994E 69	9.999994E 69	9.999994E 69
5	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
6	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
7	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
8	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
9	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69
10	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69	9.999994E 69

LISTING OF QUICK INPUT DECK ON UNIT 11

Table 6. Continued

HYPOTHETICAL ASYMMETRICALLY ABLATED, SLICED, BENT CONE										
1 1 6 NONCIRCULAR CROSS SECTION										
1LELL	1ELLI	PIEC	BLCL	SID	LSCP					
1UELL	2ELLI	PIEC	SID	TUCL	USCP					
1LSSL	3LINE	PIEC	BLSL	LSLCP					LELL	
1LSSL	4LINE	PIEC	LSLCP	SSL						UELL
1USSL	5LINE	PIEC	SSL	USLCP						
1UTSL	6LINE	PIEC	USLCP	TUSL						
1 1 MAP										
1 1 1		0.0	9.22000							
1YUCL										
1 1 LINE	PIEC	KV5								
0.0	0.0		9.22000	0.0	A	0.0	A	0.0		
1-1										
1ZUCL										
1 1 ELLX	PIEC	KV0								
0.0	0.0		0.14645	0.35355A	89.99899A	44.99998				
2 1 ELLX	PIEC	KV0								
0.14645	0.35355		0.41497	0.78369A	45.00000A	71.04884				
3 1 ELLX	PIEC	KV0								
0.41497	0.78369		0.80104	1.11372A	71.04884A	10.00000				
4 1 LINE	PIEC	KV5								
0.80104	1.11372		4.92000	1.84000A	10.00000A	10.00000				
5 1 LINE	PIEC	KV5								
4.92000	1.84000		9.22000	2.92218A	14.12625A	14.12625				
1-1										
1ZLCL										
1 1 ELLX	PIEC	KV0								
0.0	0.0		0.18750	-0.32476A	-89.99899A	-29.99998				
2 1 ELLX	PIEC	KV0								
0.18750	-0.32476		1.12962	-1.07890A	-30.00002A	-47.35210				
3 1 ELLX	PIEC	KV0								
1.12962	-1.07890		1.41057	-1.23255A	-47.35210A	-10.00000				
4 1 LINE	PIEC	KV5								
1.41057	-1.23255		5.08242	-1.88000A	-10.00000A	-10.00000				
5 1 LINE	PIEC	KV5								
5.08242	-1.88000		9.22000	-2.24199A	-4.99998A	-4.99998				
1-1										
1YSID										
1 1 ELLX	PIEC	KV0								
0.0	0.0		0.17834	0.35248A	89.99899A	36.32498				
2 1 ELLX	PIEC	KV0								
0.17834	0.35248		0.74721	0.92984A	36.32498A	54.52428				
3 1 ELLX	PIEC	KV0								
0.74721	0.92984		1.06756	1.13207A	54.52428A	10.00000				
4 1 LINE	PIEC	KV5								
1.06756	1.13207		4.99909	1.82530A	10.00000A	9.99999				
5 1 LINE	PIEC	KV5								
4.99909	1.82530		5.00333	1.82530A	0.0	A	0.0			
6 1 LINE	PIEC	KV5								
5.00333	1.82530		9.22000	2.57165A	10.03739A	10.03739				
1-1										
1ZUSL										
1 1 LINE	PIEC	KV5								
0.0	0.25724		9.65000	1.83000A	-24.64478A	-24.64478				
1-1										
1ZLSL										
1 1 LINE	PIEC	KV5								
0.0	-2.06734		7.45318	-1.44327A	4.78633A	4.78633				

Table 6. Concluded

2	ELLX	PIEC	KV0						
	7.45318	-1.44327		8.45870	-1.22635A	4.78633A	19.56165		
3	LINE	PIEC	KV5						
	8.45870	-1.22635		9.94000	-0.70000A	19.56165A	19.56165		
-1									
1	YSSL								
	1	LINE	PIEC	KV5					
		0.0	2.30000	9.22000	2.30000A	0.0	A	0.0	
-1									
1	ZSID								
	1	LINE	PIEC	KV5					
		0.0	0.0	0.85535	0.02866A	1.91908A	-1.91908		
	2	LINE	PIEC	KV5					
		0.85535	0.02866	4.99999	0.02866A	0.0	A	0.0	
	3	LINE	PIEC	KV5					
		4.99999	0.02866	8.00333	-0.06865A	-87.50502A	-87.50502		
	4	LINE	PIEC	KV5					
		5.00333	-0.06865	9.22000	0.30026A	4.99999A	4.99999		
-1									
1	ZMAP								
	1	LINE	PIEC	KV5					
		0.0	0.0	9.22000	0.0	A	0.0	A	0.0
-1									
1	YUSL	YUCL							
1	YLSL	YUCL							
1	YLCL	YUCL							
1	YNAP	YUCL							
1	YUSCP	YSID							
1	YLSCP	YSID							
1	YUSLCP	YSSL							
1	YLSLCP	YSSL							
1	ZUSCP	ZUCL							
1	ZLSCP	ZLCL							
1	ZUSLCP	ZUSL							
1	ZLSLCP	ZLSL							
1	ZSSL	ZSID							
1	2		0.00100	9.22100	0.09218				
0									

**Table 7. Definitions of Logical Names for QUICK
Cross-Section and Body Line Models**

Cross-Section Models

Logical Name	Definition
LELL	Lower ellipse
UELL	Upper ellipse
UTSL	Upper top slicing plane
USSL	Upper side slicing plane
LSSL	Lower side slicing plane
LBSL	Lower bottom slicing plane

Body Line Models

LCL	Lower centerline
LSCP	Lower slope control point
MAPAX	Map axis
SID	Side
UCL	Upper centerline
USCP	Upper slope control point
LSLCP	Control point for lower horizontal and vertical slicing planes
LSL	Centerline on lower slice
SSL	Side on side slice
USL	Centerline on upper slice
USLCP	Control point for upper horizontal and vertical slicing planes

Note: See Fig. 8 also.

Table 8. KWIKNOSE Source Listing

```

C KWIKNOSE - PROGRAM TO SET UP QUICK INPUT FOR AN AXISMETRIC STACK OF
C CONES AND OGIVES. NOT NECESSARILY TANGENT
C ORIGINAL PROGRAM VERSION 2/18/77
C-----
C INPUT INSTRUCTIONS (SEE AEDC-TR-77-89 FOR DETAILS)
C FIRST INPUT CARD IS ID DATA (80A1, QUICK READS ONLY 60A1)
C REMAINING INPUT IS VIA NAMELIST /INPUT/
C SET IPRINT = 1 TO MAKE QUICK EXERCISE MATH MODEL
C PROVIDE DD CARD FOR UNIT IF (DEFAULT = 1) WITH DCB=(RECFM=FB,
C LRECL=80), 1 CYL LONG SEQUENTIAL FILE
C IF IP=IF QUICK INPUT DECK WILL BE WRITTEN ON IP WITHOUT REWINDING. USE
C TO PUNCH.
C-----
COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),M(10),
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
4 VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
5 BLB(8)
COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
1 YUCL,ZERO,FORE,AFT,ETOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,MI,
2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,A1,MJ,RJ,A4,DX
3 TYPE,ASYM,UPFD,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGPI,ISTOP,JSTOP,KSTOP,
1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
COMMON /LS/II,T
REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
LOGICAL*1 SLOPE,IDEN,CARD,TI,T
LB=1
CALL INIT
CALL INPT
IF (TYPE.EQ. ASYM) GO TO 100
CALL CASE
CALL SEGE
CALL INTR
CALL QARY
CALL PRNT
CALL ROFL
CALL SLCE
CALL RFSL
CALL QUCK
CALL LIST
STOP
100 DO 120 LB=1,3
IF (LB.EQ.1) GO TO 110
CALL INIT
CALL INPT1
110 CALL CASE
CALL SEGE
CALL INTR
CALL QARY
CALL PRNT
CALL ROFL
CALL COPY

```

Table 8. Continued

120	CONTINUE	5700
	CALL SLCE	5800
	CALL RPSL	5900
	DO 130 LB=4,8	6000
	CALL COPY	6100
130	CONTINUE	6200
	CALL BLCK	6300
	CALL QK3D	6400
	CALL LIST	6500
	STOP	6600
	END	6700
C-----		6800
SUBROUTINE INIT		6900
C-----		7000
C INITIALIZE VARIABLES		7100
C-----		7200
	COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),N(10),	7300
	1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),	7400
	2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),	7500
	3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)	7600
	4,VB(10,3,2,8),CRVB(10,8),XMAPAX(10),ZMAPAX(10),XSIO(10),ZSIO(10),	7700
	5 BLB(8)	7800
	COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)	7900
	COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,	8000
	1 YUCL,ZERO,FORE,AFT,ETOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,MI,	8100
	2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,A1,MJ,RJ,A4,DX	8200
	3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO	8300
	COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,	8400
	1 I,J,L,IPI,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3	8500
	COMMON /LA/SLOPE(10),IDEN(80),CARD(80)	8600
	COMMON /LS/TT,T	8700
	REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV	8800
	LOGICAL*1 SLOPE,IDEN,CARD,TT,T	8900
1000	CONTINUE	9000
	IF(LB.GT.1)GO TO 1100	9100
	IPRINT=0	9200
	IR=5	9300
	IW=6	9400
	IF=1	9500
	IP=7	9600
	ERRMAX=5.E-5	9700
1100	NSEG=10	9800
	PIO180=3.1415926/180.	9900
	ISTOP=0	10000
	JSTOP=0	10100
	KSTOP=0	10200
	RETURN	10300
	END	10400
C-----		10500
SUBROUTINE INPT		10600
C-----		10700
C OBTAIN INPUT DATA		10800
C-----		10900
	COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),N(10),	11000
	1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),	11100
	2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),	11200

Table 8. Continued

```

3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3) 11300
4,VB(10,3,2,8),CRVB(10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10), 11400
5 BLB(8) 11500
COMMON /IA/(CASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8) 11600
COMMON /RS/KI,MJ,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPON,ELLX, 11700
1 YUCL,ZERO,FORE,AFFE,TPPS,BOTS,OPUP,OGIV,P10180,XJ,ZJ,XI,ZI,MI, 11800
2 RI,BI,THI,8J,A1,A2,A3,8IJ,THJ,S,THIJ,SIJ,A1,MJ,RJ,A4,DX 11900
3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,ONBO 12000
COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP, 12100
1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3 12200
COMMON /LA/SLOPE(10),IDEN(80),CARD(80) 12300
COMMON /LS/TL,T 12400
REAL K,M,KS1,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV 12500
LOGICAL*1 SLOPE,IDEN,CARD,TL,T 12600
DIMENSION XUSL(10),ZUSL(10),TUSL(10), 12700
1 XLSL(10),ZLSL(10),TLSL(10), 12800
2 XSSL(10),YSSL(10),TSSL(10) 12900
EQUIVALENCE 13000
1 (V(1,1,1),XUSL(1)),(V(1,2,1),ZUSL(1)),(V(1,3,1),TUSL(1)) 13100
2,(V(1,1,2),XLSL(1)),(V(1,2,2),ZLSL(1)),(V(1,3,2),TLSL(1)) 13200
3,(V(1,1,3),XSSL(1)),(V(1,2,3),YSSL(1)),(V(1,3,3),TSSL(1)) 13300
DIMENSION RHOU(10),RHOLS(10),RHOSS(10) 13400
EQUIVALENCE (RHOSL(1,1),RHOU(1)) 13500
1 (RHOSL(1,2),RHOLS(1)) 13600
2 (RHOSL(1,3),RHOSS(1)) 13700
NAMELIST/INPUT/X,Z,TH,CURVE,R,H,K, B,IR,IW,IF,ERRMAX,DIR,IPRINT, 13800
1 IP,RHO,SLOPE,XUSL,ZUSL,TUSL,XLSL,ZLSL,TLSL,XSSL,YSSL,TSSL 13900
2,RHOU,RHOLS,RHOSS,TYPE,XMAPAX,ZMAPAX,XSID,ZSID 14000
NAMELIST/INPUT1/IR,IW,IF,IP,IPRINT,ERRMAX 14100
2000 CONTINUE 14200
IF(IF.EQ.IP)GO TO 2005 14300
REWIND IF 14400
2002 READ(IR,90001,END=2004)CARD 14500
WRITE(IF,90001,END=2003)CARD 14600
GO TO 2002 14700
2003 STOP 2003 14800
2004 IR=IF 14900
REWIND IF 15000
2005 CONTINUE 15100
2010 FORMAT(80A1) 15200
WRITE(IW,2001) 15300
2001 FORMAT('1',30X,23('(>'))/31X,23('(>'))/ 15400
- 31X,1'(>)',63X,1'(>))/ 15500
- 31X,1'(>)' PROGRAM KWIKNOSE 15600
1 TO SET UP QUICK INPUT FOR NOSE GEOMETRIES '(>))/ 15700
2 31X,1'(>)',63X,1'(>))/ 15800
3 31X,1'(>)' VON KARMAN GAS DYNAMICS FACILITY (VKF/ADP) 15900
4 '(>))/ 16000
5 31X,1'(>)',63X,1'(>))/ 16100
6 31X,1'(>)' ARNOLD ENGINEERING DEVELOPMENT CENTER, TENNESSEE 37389 16200
7. 7X,1'(>))/ 16300
8 31X,1'(>)',63X,1'(>))/ 16400
9 31X,23('(>'))/31X,23('(>'))/ 16500
IF(IF.EQ.IP)GO TO 2013 16600
REWIND IF 16700
WRITE(IW,2006) 16800

```

Table 8. Continued

2006	FORMAT(' INPUT CARDS')	16900
	WRITE(IW,2007)	17000
2007	FORMAT(' ',82(' '))	17100
2008	READ(IR,90001,END=2011)CARD	17200
	WRITE(IW,2009)CARD	17300
2009	FORMAT(' ',80A1,' ')	17400
	GO TO 2008	17500
2011	WRITE(IW,2007)	17600
2013	CONTINUE	17700
	IF (IF.NE.IP)REWIND IF	17800
	READ(IR,2010,END=2020)IDEN	17900
	READ(IR,INPUT,END=2040)	18000
2019	WRITE(IW,9050)	18100
	WRITE(IW,2111)LB,BLB(LB)	18200
2111	FORMAT(' INPUT DATA:',12,2X,A4)	18300
	WRITE(IW,2112)	18400
2112	FORMAT('0 I', 6X,' X',11X,' Z',11X,' TM',11X,' H',11X,' K',11X,	18500
	1 ' R',11X,' B',11X,' RHO')	18600
	WRITE(IW,2113)	18700
2113	FORMAT(' ',117X,'CURVE DIR SLOPE')	18800
	WRITE(IW,2114)	18900
2114	FORMAT(' ',132(' '))	19000
	WRITE(IW,2115)(I,X(I),Z(I),TM(I),H(I),K(I),R(I),B(I),RHO(I),	19100
	1CURVE(I),DIR(I),SLOPE(I),	19200
	2 I=1,10)	19300
2115	FORMAT(10(' ',12,1P0E13.5,11X,2(1X,A4),4X,A1/))	19400
	IF (TYPE.EQ.ASYM.AND.LB.LT.3)GO TO 2023	19500
	DO 2118 L=1,3	19600
	WRITE(IW,2116)SL(L)	19700
2116	FORMAT('0',A4, 3X,'X-SL', 9X,'Z-SL', 9X,'T-SL', 8X,'RHO-S'/	19800
	1 ' ',54(' '))	19900
	WRITE(IW,2117)(I,(V(I,J,L),J=1,3),RHOSL(I,L),I=1,10)	20000
2117	FORMAT(10(' ',12,1P4E13.5/))	20100
2118	CONTINUE	20200
	WRITE(IW,2021)	20300
2021	FORMAT('0',6X,'XMAPAX',7X,'ZMAPAX',8X,'XSID',9X,'ZSID'	20400
	1/' ',54(' '))	20500
	WRITE(IW,2022)(I,XMAPAX(I),ZMAPAX(I),XSID(I),ZSID(I),I=1,10)	20600
2022	FORMAT(10(' ',12,1P4E13.5/))	20700
2023	WRITE(IW,INPUT1)	20800
	WRITE(IW,9050)	20900
	GO TO 3000	21000
2020	WRITE(IW,2030)	21100
2030	FORMAT(' COND CODE 2020: MISSING ID CARD')	21200
	WRITE(IW,2035)	21300
2035	FORMAT('EXECUTION HALTED')	21400
	STOP 2020	21500
2040	WRITE(IW,2050)IR	21600
2050	FORMAT(' COND CODE 2040: EOF ON UNIT IR=',12)	21700
	WRITE(IW,2035)	21800
	STOP 2040	21900
3000	RETURN	22000
	ENTRY INPT1	22100
	IF (IF.NE.IP)IR=IF	22200
	READ(IR,INPUT,END=2040)	22300
	GO TO 2019	22400

Table 8. Continued

```

9050 FORMAT(' ',132(' '))
90001 FORMAT(80A1)
90002 FORMAT('1')
END
C-----
SUBROUTINE CASE
C-----
C DETERMINE ICASE
C-----
COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),M(10),
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
4 VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
5 BLB(8)
COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
1 YUCL,ZERO,FORE,AFFE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,MI,
2 RI,BI,THI,BJ,A1,A2,A3,B1J,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,OX
3 TYPE,ASYN,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTD,DNBO
COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEOP1,ISTOP,JSTOP,KSTOP,
1 I,J,L,IP1,LH1,I1,NPTS,I2,ISL,LB,N,NM1,I3
COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
COMMON /LS/TI,T
REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
LOGICAL*1 SLOPE,IDEN,CARD,TI,T
DIMENSION E(10,7)
EQUIVALENCE (E(1,1),X(1))
1      ,E(1,2),Z(1))
2      ,E(1,3),TH(1))
3      ,E(1,4),M(1))
4      ,E(1,5),K(1))
5      ,E(1,6),R(1))
6      ,E(1,7),B(1))
3000 CONTINUE
C DETERMINE NO. SEGMENTS NSEG
DO 3010 J=1,10
I=10-J+1
ICASE(I)=0
IF(CURVE(I),NE,BLNK)GO TO 3010
NSEG=I-1
3010 CONTINUE
WRITE(IW,3015)NSEG
3015 FORMAT('NUMBER OF SEGMENTS NSEG =',I3)
C MAKE SURE CURVE='LINE' OR 'OGIV' SPECIFIED FOR EACH SEG
DO 3040 I=1,NSEG
DO 3020 J=1,2
IF(CURVE(I).EQ,KRV(J))GO TO 3040
3020 CONTINUE
WRITE(IW,3030)I,CURVE(I),KRV(1),KRV(2)
3030 FORMAT(' COND CODE 3030: CURVE('',I2,'')='',A4,' NOT = TO '',A4,' OR '
1,A4)
ISTOP=1
3040 CONTINUE
IF(ISTOP.EQ.1)STOP 3030
DO 3080 I=1,NSEG

```

Table 8. Continued

```

L=0
DO 3050 J=1,7
IF (E(I,J).LT.RINF) L=L+2**J
3050 CONTINUE
IF (SLOPE(I).EQ.T) L=L+256
IF (CURVE(I).EQ.KRV(2)) L=L+512
DO 3060 J=1,21
IF (L.NE.JCASE(J)) GO TO 3060
ICASE(I)=J
GO TO 3080
3060 CONTINUE
ISTOP=1
WRITE(IW,3070) I
3070 FORMAT(' COND CODE 3070: INPUT DATA FOR SEGMENT ',I2,' DOES NOT MA
ITCH ANY OF PROGRAMMED OPTIONS')
3080 CONTINUE
WRITE(IW,3090) (I,ICASE(I),I=1,10)
3090 FORMAT(10('/' ICASE('I2,') ='I3'))
IF (ISTOP.EQ.0) GO TO 4000
WRITE(IW,2035)
STOP 3070
4000 RETURN
2035 FORMAT('0EXECUTION HALTED')
END
C-----
SUBROUTINE SEGE
C-----
C DETERMINE SEGMENT EQUATION
C-----
COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),M(10),
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
4 VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
5 BLB(8)
COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
1 YUCL,ZERO,FORE,AFFE,TOPS,BOTS,OPUP,OGIV,P10180,XJ,ZJ,XI,ZI,MI,
2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,A1,MJ,RJ,A4,DX
3 TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,ONTO,DNBO
COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
1 I,J,L,IPI,IMI,I1,NPTS,I2,ISL,LB,N,NMI,I3
COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
COMMON /LS/TT,T
REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
LOGICAL*1 SLOPE,IDEN,CARD,TT,T
4000 CONTINUE
XJ=0.
ZJ=0.
DO 4990 I=1,NSEG
I1=ICASE(I)
IF (I.EQ.1) GO TO 4007
IF (ICASE(I-1).NE.5.AND.ICASE(I-1).NE.9) GO TO 4007
DO 4006 J=1,7
IF (I1.NE.NCASE(J)) GO TO 4006
WRITE(IW,4005) ICASE(I),ICASE(I-1)

```

Table 8. Continued

4005	FORMAT(' COND CODE 4005: CASE ',I2,' MAY NOT BE USED AFTER CASE ',	33700
	I12)	33800
	ISTOP=1	33900
	GO TO 4007	34000
4006	CONTINUE	34100
4007	CONTINUE	34200
	XI=X(I)	34300
	ZI=Z(I)	34400
	MI=M(I)	34500
	KI=K(I)	34600
	RI=R(I)	34700
	BI=B(I)	34800
	TI=SLOPE(I)	34900
	THI=TH(I)	35000
C		35100
C		35200
	GO TO(4070,4090,4115,4120,4150,4010,4020,4030,4150,4160,	35300
	1 4170,4180,4190,4200,4300,4400,4410,4420,4430,4460,4490	35400
	2),I1	35500
C	ICASE=6	35600
	4010 M(I)=(ZI-ZJ)/(XI-XJ)	35700
	TH(I)=ATAN(M(I))/PI0180	35800
	B(I)=ZI-M(I)*XI	35900
	GO TO 4040	36000
C	ICASE=7	36100
	4020 M(I)=TAN(TH(I)*PI0180)	36200
	Z(I)=M(I)*(X(I)-XJ)+ZJ	36300
	B(I)=Z(I)-M(I)*XI	36400
	GO TO 4040	36500
C	ICASE=8	36600
	4030 M(I)=TAN(TH(I)*PI0180)	36700
	X(I)=XJ+(ZI-ZJ)/M(I)	36800
	B(I)=ZI-M(I)*X(I)	36900
	GO TO 4040	37000
C	CHECK REDUNDANT INPUT DATA	37100
4040	JSTOP=0	37200
	IF(XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTOP=1	37300
	IF(JSTOP.EQ.1)WRITE(IW,4050)JSTOP,I,ERRMAX	37400
	IF(ZI.LT.RINF.AND.ABS(ZI-Z(I)).GT.ERRMAX)JSTOP=2	37500
	IF(JSTOP.EQ.2)WRITE(IW,4050)JSTOP,I,ERRMAX	37600
	IF(BI.LT.RINF.AND.ABS(BI-B(I)).GT.ERRMAX)JSTOP=3	37700
	IF(JSTOP.EQ.3)WRITE(IW,4050)JSTOP,I,ERRMAX	37800
	IF(THI.LT.RINF.AND.ABS(THI-TH(I)).GT.ERRMAX)JSTOP=4	37900
	IF(JSTOP.EQ.4)WRITE(IW,4050)JSTOP,I,ERRMAX	38000
4050	FORMAT(' COND CODE 4050-' ,I1,' REDUNDANT INPUT DATA FOR SEGMENT '	38100
	1,I2,' NOT SELF CONSISTENT TO WITHIN ',1PE15.7)	38200
	IF(JSTOP.NE.0)WRITE(IW,4055)	38300
4055	FORMAT(' SUGGEST USER CHECK IS INPUT DATA')	38400
	GO TO 4150	38500
C	ICASE=1	38600
	4070 MJ=-(XI-XJ)/(ZI-ZJ)	38700
	BJ=ZI-ZJ-MJ*(XI-XJ)	38800
	BJ=0.5*BJ	38900
	A1=(XI+MJ*(ZI-BJ))/(1.+MJ**2)	39000
	A2=(XI+MJ*(ZI-BJ))**2-(1.+MJ**2)*(XI**2+(ZI-BJ)**2-R1**2)	39100
	A2=SQRT(A2)/(1.+MJ**2)	39200

Table 8. Continued

```

M(I)=A1+A2
K(I)=MJ*M(I)+BJ
IF(DIR(I).EQ.OPDN.AND.K(I).LT.ZI)GO TO 4080
M(I)=A1-A2
K(I)=MJ*M(I)+BJ
4080 M(I)=-(X(I)-H(I))/(Z(I)-K(I))
TH(I)=ATAN(M(I))/PI0180
GO TO 4130
C ICASE=2
4090 IF(I.GT.1)GO TO 4110
WRITE(IW,4100)
4100 FORMAT(' COND CODE 4100: FIRST SEGMENT MAY NOT BE SPECIFIED TANGEN
IT TO PREVIOUS SEGMENT')
WRITE(IW,2035)
STOP 4100
4110 MJ=-(XI-XJ)/(ZI-ZJ)
BJ=ZI-ZJ-MJ*(XI+XJ)
BJ=0.5*BJ
MI=-1./TAN(PI0180*TH(I-1))
BI=ZJ-MI*XJ
M(I)=-(BJ-BI)/(MJ-MI)
K(I)=MJ*M(I)+BJ
R(I)=SQRT((XI-M(I))**2+(ZI-K(I))**2)
TH(I)=ATAN((XI-M(I))/(ZI-K(I)))/PI0180
GO TO 4130
C ICASE=3
4115 S=1.
IF(DIR(I).NE.OPDN)S=-1.
M(I)=XJ+RI*SIN(PI0180*THJ)*S
K(I)=ZJ-RI*COS(PI0180*THJ)*S
Z(I)=K(I)+S*SQRT(RI**2-(XI-M(I))**2)
TH(I)=ATAN((XI-M(I))/(Z(I)-K(I)))/PI0180
GO TO 4130
C ICASE=4
4120 MJ=-1./TAN(PI0180*TH(I))
BJ=ZI-MJ*XI
MIJ=-(XI-XJ)/(ZI-ZJ)
BIJ=ZI-ZJ-MIJ*(XI+XJ)
BIJ=0.5*BIJ
M(I)=-(BJ-BIJ)/(MJ-MIJ)
K(I)=MJ*M(I)+BJ
R(I)=SQRT((XI-M(I))**2+(ZI-K(I))**2)
GO TO 4130
C CHECK REDUNDANT DATA
4130 KSTOP=0
IF(XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)KSTOP=1
IF(KSTOP.EQ.1)WRITE(IW,*140)KSTOP,I,ERRMAX
IF(ZI.LT.RINF.AND.ABS(ZI-Z(I)).GT.ERRMAX)KSTOP=2
IF(KSTOP.EQ.2)WRITE(IW,*140)KSTOP,I,ERRMAX
IF(THI.LT.RINF.AND.ABS(THI-TH(I)).GT.ERRMAX)KSTOP=3
IF(KSTOP.EQ.3)WRITE(IW,*140)KSTOP,I,ERRMAX
IF(MI.LT.RINF.AND.ABS(MI-M(I)).GT.ERRMAX)KSTOP=4
IF(KSTOP.EQ.4)WRITE(IW,*140)KSTOP,I,ERRMAX
IF(XI.LT.RINF.AND.ABS(XI-K(I)).GT.ERRMAX)KSTOP=5
IF(KSTOP.EQ.5)WRITE(IW,*140)KSTOP,I,ERRMAX
IF(RI.LT.RINF.AND.ABS(RI-R(I)).GT.ERRMAX)KSTOP=6

```

39300
39400
39500
39600
39700
39800
39900
40000
40100
40200
40300
40400
40500
40600
40700
40800
40900
41000
41100
41200
41300
41400
41500
41600
41700
41800
41900
42000
42100
42200
42300
42400
42500
42600
42700
42800
42900
43000
43100
43200
43300
43400
43500
43600
43700
43800
43900
44000
44100
44200
44300
44400
44500
44600
44700
44800

Table 8. Continued

IF(KSTOP.EQ.6)WRITE(IW,140)KSTOP,I,ERRMAX	44900
4140 FORMAT(' COND CODE 4140=',I1,' REDUNDANT INPUT DATA FOR SEGMENT '	45000
1,12,' NOT SELF CONSISTENT TO WITHIN ',1PE15.7)	45100
IF(KSTOP.NE.0)WRITE(IW,4055)	45200
GO TO 4150	45300
4150 CONTINUE	45400
XJ=X(I)	45500
ZJ=Z(I)	45600
THJ=TH(I)	45700
GO TO 4990	45800
C ICASE=10	45900
4160 S=1.	46000
IF(DIR(I).EQ.OPUP)S=-1.	46100
H(I)=XI*S*RI*SIN(PI0180*THI)	46200
K(I)=ZI-S*RI*COS(PI0180*THI)	46300
GO TO 4130	46400
C ICASE=11	46500
4170 S=1.	46600
IF(DIR(I).EQ.OPUP)S=-1.	46700
Z(I)=KI*S*SQRT(RI**2-(XI-HI)**2)	46800
TH(I)=-ATAN((XI-HI)/(Z(I)-KI))/PI0180	46900
GO TO 4130	47000
C ICASE=12	47100
4180 S=1.	47200
IF(DIR(I).EQ.FORE.OR.DIR(I).EQ.AFTE)GO TO 4186	47300
WRITE(IW,4185)I	47400
4185 FORMAT(' COND CODE 4185: DIR MUST BE EITHER 'FORE' OR 'AFTE' F	47500
LORE SEGMENT',I3)	47600
WRITE(IW,2035)	47700
STOP 4185	47800
4186 CONTINUE	47900
IF(DIR(I).EQ.FORE)S=-1.	48000
X(I)=HI*S*SQRT(RI**2-(ZI-KI)**2)	48100
TH(I)=-ATAN((X(I)-HI)/(ZI-KI))/PI0180	48200
GO TO 4130	48300
C ICASE=13	48400
4190 S=1.	48500
IF(DIR(I).EQ.OPUP)S=-1.	48600
X(I)=HI-S*RI*SIN(PI0180*THI)	48700
Z(I)=KI-S*RI*COS(PI0180*THI)	48800
GO TO 4130	48900
C ICASE=14	49000
4200 B(I)=ZI-XI*TAN(PI0180*THI)	49100
GO TO 4040	49200
C ICASE=15	49300
4300 Z(I)=XI*TAN(PI0180*THI)+HI	49400
GO TO 4040	49500
C ICASE=16	49600
4400 X(I)=(ZI-BI)/TAN(PI0180*THI)	49700
GO TO 4040	49800
C ICASE=17	49900
4410 IF(ABS(Z(I)-X(I)*TAN(PI0180*TH(I))-B(I)).LT.ERRMAX)GO TO 4150	50000
JSTOP=5	50100
WRITE(IW,4050)JSTOP,I,ERRMAX	50200
GO TO 4150	50300
C ICASE=18	50400

Table 8. Continued

```

4420 IF (ABS((X(I)-H(I))*2+(Z(I)-K(I))*2-R(I)*2).LT.ERRMAX)GO TO 4150 50500
      KSTOP=7 50600
      WRITE(IW,4140)KSTOP,I,ENRMAX 50700
      GO TO 4150 50800
C ICASE=19 50900
4430 S=0. 51000
      IF (DIR(I).EQ.UPFO)S=1. 51100
      IF (DIR(I).EQ.UPAF)S=1. 51200
      IF (DIR(I).EQ.UPTO)S=1. 51300
      IF (DIR(I).EQ.UPBO)S=1. 51400
      IF (DIR(I).EQ.ONFO)S=1. 51500
      IF (DIR(I).EQ.ONAF)S=1. 51600
      IF (DIR(I).EQ.ONTO)S=1. 51700
      IF (DIR(I).EQ.ONBO)S=1. 51800
      IF (S.NE.0.)GO TO 4440 51900
      WRITE(IW,4431)I 52000
4431 FORMAT(' COND CODE 4431: DIR(I) EQUAL TO ILLEGAL VALUE FOR CASE 19 52100
1 FOR SEGMENT',I3) 52200
      WRITE(IW,2035) 52300
      STOP 4431 52400
4440 IF (I.NE.NSEG)GO TO 4450 52500
      WRITE(IW,4441)I 52600
4441 FORMAT(' COND CODE 4441: ICASE=19 MAY NOT BE USED FOR LAST SEGMENT 52700
1',I3) 52800
      WRITE(IW,2035) 52900
      STOP 4441 53000
4450 H(I)=XJ-S*R1 *SIN(PI0180*THJ) 53100
      K(I)=ZJ-S*R1 *COS(PI0180*THJ) 53200
      IF (DIR(I).EQ.UPFO)DIR(I)=FORE 53300
      IF (DIR(I).EQ.UPAF)DIR(I)=AFTE 53400
      IF (DIR(I).EQ.UPTO)DIR(I)=TOPS 53500
      IF (DIR(I).EQ.UPBO)DIR(I)=BOTS 53600
      IF (DIR(I).EQ.ONFO)DIR(I)=FORE 53700
      IF (DIR(I).EQ.ONAF)DIR(I)=AFTE 53800
      IF (DIR(I).EQ.ONTO)DIR(I)=TOPS 53900
      IF (DIR(I).EQ.ONBO)DIR(I)=BOTS 54000
      ICASE(I)=5 54100
      GO TO 4130 54200
C ICASE=20 54300
4460 IM1=1-1 54400
      A1=SQRT((X1-H(IM1))*2+(Z1-K(IM1))*2) 54500
      A2=ARCOS(R(IM1)/A1)/PI0180 54600
      A3=ATAN((Z1-K(IM1))/(X1-H(IM1)))/PI0180 54700
      S=0. 54800
      IF (DIR(I).EQ.TOPS)S=1. 54900
      IF (DIR(I).EQ.BOTS)S=1. 55000
      IF (S.NE.0.)GO TO 4470 55100
      WRITE(IW,4461)I 55200
4461 FORMAT(' COND CODE 4461: DIR MUST BE EITHER TOPS OR BOTS FOR SEGME 55300
NT',I3) 55400
      WRITE(IW,2035) 55500
      STOP 4461 55600
4470 IF (I.NE.1)GO TO 4480 55700
      WRITE(IW,4471) 55800
4471 FORMAT(' COND CODE 4471: ICASE=20 MAY NOT BE USED FOR FIRST SEGME 55900
NT') 56000

```

Table 8. Continued

```

WRITE(IW,2035)
STOP 4471
4480 TH(I)=A3-S*(90.-A2)
M(I)=TAN(PI0180*TH(I))
B(I)=Z1-M(I)*XI
GO TO 4040
C ICASE=21
4490 IF(I.NE.1)GO TO 4500
WRITE(IW,4491)
4491 FORMAT(' COND CODE 4491: ICASE=21 MAY NOT BE USED WITH FIRST SEGME
1ENT')
WRITE(IW,2035)
STOP 4491
4500 IF(CURVE(I-1).EQ.0GIV)GO TO 4540
C LINE=OGIVE
IM1=I-1
MJ=TAN(PI0180*TH(IM1))
A1=MJ*X(I)+B(IM1)
S=1.
IF(A1.GT.Z(I))S=-1.
BJ=B(IM1)+S*R(I)/COS(PI0180*TH(IM1))
A1=1.-MJ**2
A2=2.*(BJ-Z(I))*MJ-X(I)
A3=X(I)**2+(BJ-Z(I))**2-R(I)**2
A4=A2/2./A1
A5=A3/A1
A5=A4**2-A5
IF(A5.GE.0.)GO TO 4515
IF(ABS(A5).GE.ERRMAX)GO TO 4510
WRITE(IW,4501)
4501 FORMAT(' COND CODE 4501: SEGMENT',I3,' LOCUS OF CIRCLE CENTERS DO
IES NOT INTERSECT')
WRITE(IW,2035)
STOP 4501
4510 A5=0.
4515 A5=SQRT(A5)
A1=-A4+A5
A2=-A4-A5
A3=A1*MJ+BJ
A4=A2*MJ+BJ
IF(A1.GT.A2)GO TO 4520
M(I)=A1
K(I)=A3
GO TO 4530
4520 M(I)=A2
K(I)=A4
4530 TH(I)=-ATAN((X(I)-M(I))/(Z(I)-K(I)))/PI0180
GO TO 4130
C OGIVE=OGIVE
4540 IM1=I-1
IF(DIR(IM1).EQ.OPDN.OR.UIR(IM1).EQ.OPUP)GO TO 4550
4541 WRITE(IW,4542)IM1
4542 FORMAT(' COND CODE 4542: SEGMENT',I3,' DIR NOT EQUAL TO OPUP OR OP
ION')
WRITE(IW,2035)
STOP 4542

```

56100
56200
56300
56400
56500
56600
56700
56800
56900
57000
57100
57200
57300
57400
57500
57600
57700
57800
57900
58000
58100
58200
58300
58400
58500
58600
58700
58800
58900
59000
59100
59200
59300
59400
59500
59600
59700
59800
59900
60000
60100
60200
60300
60400
60500
60600
60700
60800
60900
61000
61100
61200
61300
61400
61500
61600

Table 8. Continued

4550	IF(DIR(I).EQ.OPUP.OR.DIR(I).EQ.OPDN)GO TO 4560	61700
	IM1=I	61800
	GO TO 4541	61900
4560	S=-1.	62000
	IF(DIR(IM1).EQ.DIR(I))S=-1.	62100
	IF(Z(I).EQ.K(IM1))GO TO 4630	62200
	MJ=(X(I)-H(IM1))/(Z(I)-K(IM1))	62300
	BJ=X(I)**2+Z(I)**2-H(IM1)**2-K(IM1)**2+R(IM1)*(R(IM1)+2.*S*R(I))	62400
	BJ=BJ*.5/(Z(I)-K(IM1))	62500
	A1=1-MJ**2	62600
	A2=2.*(BJ-Z(I))*MJ-X(I)	62700
	A3=X(I)**2+(BJ-Z(I))**2-R(I)**2	62800
	A4=.5*A2/A1	62900
	A5=A3/A1	63000
	A5=A4**2-A5	63100
	IF(A5.GE.0.)GO TO 4580	63200
	IF(ABS(A5).GE.ERRMAX)GO TO 4570	63300
	WRITE(IW,4561)I	63400
4561	FORMAT(' COND CODE 4561: SEGMENT',I3,' LOCUS OF CIRCLE CENTERS DO	63500
	IES NOT INTERSECT')	63600
	WRITE(IW,2035)	63700
	STOP 4561	63800
4570	A5=0.	63900
4580	A5=SQRT(A5)	64000
	A1=-A4+A5	64100
	A2=-A4-A5	64200
	A3=MJ*A1+BJ	64300
	A4=MJ*A2+BJ	64400
4585	IF(DIR(IM1).EQ.DIR(I))GO TO 4622	64500
	IF(DIR(IM1).EQ.OPDN.AND.DIR(I).EQ.OPUP)GO TO 4590	64600
	IF(DIR(IM1).EQ.OPUP.AND.DIR(I).EQ.OPDN)GO TO 4620	64700
	STOP 4581	64800
4590	IF(A4.GT.A3)GO TO 4610	64900
4600	M(I)=A1	65000
	K(I)=A3	65100
	GO TO 4660	65200
4610	M(I)=A2	65300
	K(I)=A4	65400
	GO TO 4660	65500
4620	IF(A4.GT.A3)GO TO 4600	65600
	GO TO 4610	65700
4630	A1=(X(I)**2+Z(I)**2-H(IM1)**2-K(IM1)**2+R(IM1)*(R(IM1)+2.*	65800
	S*R(I))*.5/(X(I)-H(IM1))	65900
	A2=R(I)**2-(A1-X(I))**2	66000
	IF(A2.GT.0.)GO TO 4650	66100
	IF(ABS(A2).GE.ERRMAX)GO TO 4640	66200
	WRITE(IW,4561)I	66300
	STOP 4562	66400
4640	A2=0.	66500
4650	A2=SQRT(A2)	66600
	A3=Z(I)*A2	66700
	A4=Z(I)-A2	66800
	A2=A1	66900
	GO TO 4585	67000
4622	IF(DIR(IM1).EQ.OPDN.AND.DIR(I).EQ.OPDN)GO TO 4620	67100
	IF(DIR(IM1).EQ.OPUP.AND.DIR(I).EQ.OPUP)GO TO 4590	67200

Table 8. Continued

```

      STOP 4623
4660 TH(I)=-ATAN((X(I)-H(I))/(Z(I)-K(I)))/PI*180
      GO TO 4130
4990 CONTINUE
      IF(JSTOP.NE.0)WRITE(IW,2035)
      IF(KSTOP.NE.0)WRITE(IW,2035)
      IF(ISTOP.NE.0)WRITE(IW,2035)
      IF(JSTOP.NE.0)STOP 4050
      IF(KSTOP.NE.0)STOP 4140
      IF(ISTOP.NE.0)STOP 4005
      RETURN
2035 FORMAT('0EXECUTION HALTED')
      END
C-----
SUBROUTINE INTR
C-----
C COMPUTE SEGMENT INTERSECTIONS FOR ICASE=5,9
C-----
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
3 KV(2),V(10,3,3),SL(3),R-OSL(10,3),CRVSL(10,3),
4 VB(10,3,2,8),CRVB(10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
1 YUCL,ZERO,FORE,AFFE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,MI,
2 RI,BI,THI,BJ,AI,A2,A3,HIJ,THJ,S,THIJ,SIJ,AI,MJ,RJ,A4,DX
3 TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IUVN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
5000 CONTINUE
      DO 5320 I=1,NSEG
      IP1=I+1
      IF(ICASE(I).NE.5.AND.ICASE(I).NE.9)GO TO 5320
      IF(I.NE.NSEG)GO TO 5020
      WRITE(IW,5010)
5010 FORMAT(' COND CODE 5010: EQUATION IS INSUFFICIENT FOR LAST SEGMENT
1')
      WRITE(IW,2035)
      STOP 5010
5020 CONTINUE
      IF(ICASE(I).EQ.5.AND.CURVE(IP1).EQ.OGIV)GO TO 5040
      IF(ICASE(I).EQ.5.AND.CURVE(IP1).EQ.LINE)GO TO 5250
      IF(ICASE(I).EQ.9.AND.CURVE(IP1).EQ.OGIV)GO TO 5260
      IF(ICASE(I).EQ.9.AND.CURVE(IP1).EQ.LINE)GO TO 5240
      WRITE(IW,5025)
5025 FORMAT(' COND CODE 5030: OOPS! PROGRAMMER ERROR,')
      WRITE(IW,2035)
5030 STOP 5030
C OGIVE-OGIVE INTERSECTION
5040 CONTINUE

```

Table 8. Continued

THIJ=ATAN2(K(IP1)-K(I),H(IP1)-H(I))/PI0180	72900
SIJ=SQRT((H(IP1)-H(I))**2+(K(IP1)-K(I))**2)	73000
AI=(R(I)**2-SIJ**2-R(IP1)**2)/(2.*R(I)*SIJ)	73100
IF(ABS(AI).GT.1.)AI=SIGN(1.,AI)	73200
AI=ARCOS(AI)/PI0180	73300
XI=M(I)+R(I)*COS((THIJ+AI)*PI0180)	73400
X(I)=M(I)+R(I)*COS((THIJ-AI)*PI0180)	73500
IF(ABS(X(I)-XI).LE.ERRMAX)GO TO 5090	73600
IF(I.LT.NSEQ.AND.SLOPE(I+1).EQ.T)GO TO 5070	73700
IF(DIR(I).EQ.FORE)GO TO 5060	73800
IF(DIR(I).EQ.AFTE)GO TO 5080	73900
WRITE(IW,5050)I	74000
5050 FORMAT(' COND CODE 5050: DIR MUST BE EITHER **FORE** OR **AFTE** F	74100
1OR SEGMENT ',I2)	74200
WRITE(IW,2035)	74300
STOP 5050	74400
5060 IF(XI.GE.X(I))GO TO 5070	74500
5065 X(I)=XI	74600
Z(I)=K(I)+R(I)*SIN(PI0180*(THIJ+AI))	74700
GO TO 5230	74800
5070 Z(I)=K(I)+R(I)*SIN(PI0180*(THIJ-AI))	74900
GO TO 5230	75000
5080 IF(XI.GT.X(I))GO TO 5065	75100
GO TO 5070	75200
5090 ZI=K(I)+R(I)*SIN(PI0180*(THIJ+AI))	75300
Z(I)=K(I)+R(I)*SIN(PI0180*(THIJ-AI))	75400
IF(ABS(Z(I)-ZI).LE.ERRMAX)GO TO 5230	75500
IF(DIR(I).EQ.TOP5)GO TO 5200	75600
IF(DIR(I).EQ.BOTS)GO TO 5220	75700
WRITE(IW,5100)I	75800
5100 FORMAT(' COND CODE 5100: DIR MUST BE EITHER **TOP5** OR **BOTS** F	75900
1OR SEGMENT ',I2)	76000
WRITE(IW,2035)	76100
STOP 5100	76200
5200 IF(ZI.GT.Z(I))GO TO 5210	76300
5205 X(I)=M(I)+R(I)*COS(PI0180*(THIJ+AI))	76400
GO TO 5230	76500
5210 Z(I)=ZI	76600
X(I)=M(I)+R(I)*COS(PI0180*(THIJ+AI))	76700
GO TO 5230	76800
5220 IF(ZI.GT.Z(I))GO TO 5205	76900
GO TO 5210	77000
5230 TH(I)=-ATAN((X(I)-M(I))/(Z(I)-K(I)))/PI0180	77100
GO TO 5320	77200
C LINE-LINE INTERSECTION	77300
5240 M(I)=TAN(PI0180*TH(I))	77400
M(IP1)=TAN(PI0180*TH(IP1))	77500
X(I)=(B(IP1)-B(I))/(M(IP1)-M(I))	77600
Z(I)=(M(I)*B(IP1)-M(IP1)*B(I))/(M(I)-M(IP1))	77700
GO TO 5320	77800
C OGIVE-LINE INTERSECTION	77900
5250 CONTINUE	78000
M(IP1)=TAN(PI0180*TH(IP1))	78100
MI=M(IP1)	78200
BI=B(IP1)	78300
NJ=M(I)	78400

Table 8. Continued

```

      KJ=K(I)
      RJ=R(I)
      GO TO 5270
C LINE-OGIVE INTERSECTION
5260 CONTINUE
      M(I)=TAN(PI*180*TH(I))
      MI=M(I)
      BI=B(I)
      HJ=H(IP1)
      KJ=K(IP1)
      RJ=R(IP1)
      GO TO 5270
5270 A1=1.-MI**2
      A2=MI*(BI-KJ)-HJ
      A3=HJ**2*(BI-KJ)**2-RJ**2
      A4=A2**2-A1*A3
      IF(A4.GT.0.)GO TO 5273
      IF(ABS(A4).LT.ERRMAX)GO TO 5272
      WRITE(IW,5271)I
5271 FORMAT(' COND CODE 5271: SQRT(<0) FOR SEGMENT',I3,' NO INTERSECTIO
IN.')
      WRITE(IW,2035)
      STOP 5271
5272 A4=0.
5273 A4=SQRT(A4)
      XI=(-A2+A4)/A1
      X1=(-A2-A4)/A1
      IF(ABS(X(I)-X1).LE.ERRMAX)GO TO 5300
      IF(1.LT.NSEQ.AND.SLOPE(I+1).EQ.T)GO TO 5300
      IF(DIR(I).EQ.FORE)GO TO 5290
      IF(DIR(I).EQ.AFT)GO TO 5310
      WRITE(IW,5280)I
5280 FORMAT(' COND CODE 5280: DIR MUST BE EITHER **FORE** OR **AFT** F
OR SEGMENT',I2)
      WRITE(IW,2035)
      STOP 5280
5290 IF(X1.GT.X(I))GO TO 5300
5295 X(I)=X1
5300 Z(I)=MI*X(I)+BI
      GO TO 5320
5310 IF(X1.GT.X(I))GO TO 5295
      GO TO 5300
5320 CONTINUE
      RETURN
2035 FORMAT('0EXECUTION HALTED')
      END
C*****
      SUBROUTINE QARY
C-----
C SET UP FINAL ARRAYS FOR QUICK MODEL
C-----
      COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),M(10),
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
3 KV(2),V(10,3,3),SL(3),R OSL(10,3),CRVSL(10,3)
4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),

```

Table 8. Continued

```

5 8LB(8)                                84100
COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8) 84200
COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX, 84300
1 YUCL,ZERO,FORE,AFFE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,MI, 84400
2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,MJ,RJ,A4,DX 84500
3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTD,DNBO 84600
COMMON /IS/IR,IV,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP, 84700
1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3 84800
COMMON /LA/SLOPE(10),IDEN(80),CARD(80) 84900
COMMON /LS/II,T 85000
REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV 85100
LOGICAL*1 SLOPE,IDEN,CARD,II,T 85200
C TQ(I,1) IS AT BEGINNING OF SEGMENT I 85300
C TQ(I,2) IS AT END SEGMENT I 85400
C XQ(I) AND ZQ(I) ARE AT BEGINNING OF SEGMENT I 85500
6000 CONTINUE 85600
XQ(I)=0. 85700
ZQ(I)=0. 85800
DO 6010 I=1,NSEG 85900
IF(CURVE(I).EQ.KRV(2))CURVE(I)=ELLX 86000
XQ(I+1)=X(I) 86100
ZQ(I+1)=Z(I) 86200
6010 CONTINUE 86300
DO 6030 I=1,NSEG 86400
IP1=I+1 86500
IF(CURVE(I).NE.LINE)GO TO 6020 86600
TQ(I,1)=ATAN((ZQ(IP1)-ZQ(I))/(XQ(IP1)-XQ(I)))/PIO180 86700
TQ(I,2)=TQ(I,1) 86800
GO TO 6030 86900
6020 IF(ZQ(I).EQ.K(I))GO TO 6025 87000
TQ(I,1)=-ATAN((XQ(I)-H(I))/(ZQ(I)-K(I)))/PIO180 87100
GO TO 6024 87200
6025 TQ(I,1)=90. 87300
6024 TQ(I,2)=-ATAN((XQ(IP1)-H(I))/(ZQ(IP1)-K(I)))/PIO180 87400
6030 CONTINUE 87500
NSEGP1=NSEG+1 87600
C SET ALL RIGHT ANGLES (90 DEG) TO "18M RIGHT ANGLES" (89.999 DEG) 87700
DO 6040 I=1,NSEGP1 87800
DO 6040 J=1,2 87900
IF(TQ(I,J).GT.RINF)GO TO 6040 88000
IF(ABS(TQ(I,J)).LE.89.999)GO TO 6040 88100
TQ(I,J)=SIGN(89.999,TQ(I,J)) 88200
6040 CONTINUE 88300
RETURN 88400
END 88500
C----- 88600
SUBROUTINE PRNT 88700
C----- 88800
C PRINT RESULTS 88900
C----- 89000
COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),M(10), 89100
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10), 89200
2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2), 89300
3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3) 89400
4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10), 89500
5 8LB(8) 89600

```

Table 8. Continued

```

COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8) 89700
COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX, 89800
1 YUCL,ZERO,FORE,AFFE,TOPS,BOTS,OPUP,OGIV,PI0180,XJ,ZJ,XI,ZI,MI, 89900
2 RI,BI,THI,BJ,AI,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX 90000
3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO 90100
COMMON /IS/IR,IN,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP, 90200
1 I,J,L,IP1,LM1,II,NPTS,IZ,ISL,LSB,N,NM1,I3 90300
COMMON /LA/SLOPE(10),IDEN(80),CARD(80) 90400
COMMON /LS/TT,T 90500
REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV 90600
LOGICAL*1 SLOPE,IDEN,CARD,TT,T 90700
7000 CONTINUE 90800
WRITE(IW,7005) 90900
7005 FORMAT('0SEGMENT END POINT COORDINATES AND SLOPES:') 91000
WRITE(IW,7010) 91100
7010 FORMAT('0',9X,'X',13X,'Z',12X,'TA',12X,'TF',/,' ',58(' '),) 91200
DO 7020 I=1,11 91300
WRITE(IW,90008)I,XQ(I),ZQ(I),TQ(I,2),TQ(I,1) 91400
7020 CONTINUE 91500
WRITE(IW,7025) 91600
7025 FORMAT('0SEGMENT EQUATIONS:') 91700
WRITE(IW,7030) 91800
7030 FORMAT('0',9X,'X',13X,'H',13X,'K',13X,'R',13X,'M',13X,'T', 91900
2 13X,'B',/,' ',100(' '),) 92000
DO 7040 I=1,10 92100
WRITE(IW,90008)I,X(I),M(I),K(I),R(I),M(I),TH(I),B(I) 92200
7040 CONTINUE 92300
RETURN 92400
90008 FORMAT(' ',I2,1P9E14,6) 92500
END 92600
C***** 92700
SUBROUTINE ROFL 92800
C----- 92900
C INSERT ROUNDS AND FILLETS 93000
C----- 93100
COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),M(10), 93200
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10), 93300
2 KSI(10),ETA(10),RMO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2), 93400
3 KV(2),V(10,3,3),SL(3),RMOSL(10,3),CRVSL(10,3) 93500
4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10), 93600
5 BLB(8) 93700
COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8) 93800
COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX, 93900
1 YUCL,ZERO,FORE,AFFE,TOPS,BOTS,OPUP,OGIV,PI0180,XJ,ZJ,XI,ZI,MI, 94000
2 RI,BI,THI,BJ,AI,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX 94100
3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO 94200
COMMON /IS/IR,IN,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP, 94300
1 I,J,L,IP1,LM1,II,NPTS,IZ,ISL,LSB,N,NM1,I3 94400
COMMON /LA/SLOPE(10),IDEN(80),CARD(80) 94500
COMMON /LS/TT,T 94600
REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV 94700
LOGICAL*1 SLOPE,IDEN,CARD,TT,T 94800
6100 CONTINUE 94900
ISKIP=0 95000
DO 6800 I=1,9 95100
IP1=I+1 95200

```

Table 8. Continued

```

IF(RHO(I).GT.RINF)GO TO 6800                                95300
ISKIP=1                                                       95400
IF(CURVE(I).EQ.LINE.AND.CURVE(IP1).EQ.LINE)GO TO 6110      95500
IF(CURVE(I).EQ.LINE.AND.CURVE(IP1).EQ.ELLX)GO TO 6200      95600
IF(CURVE(I).EQ.ELLX.AND.CURVE(IP1).EQ.LINE)GO TO 6300      95700
IF(CURVE(I).EQ.ELLX.AND.CURVE(IP1).EQ.ELLX)GO TO 6600      95800
WRITE(IW,6101)I                                              95900
6101 FORMAT(' COND CODE 6101: PROGRAM OR USER ERROR, SEGMENT',I3) 96000
WRITE(IW,2035)                                              96100
STOP 6101                                                    96200
C LINE-LINE FILLETTING/ROUNDING                               96300
6110 CONTINUE                                                96400
IF(ABS(TQ(I,2)-TQ(IP1,1)).GT.ERRMAX)GO TO 6120             96500
WRITE(IW,6111)I,IP1                                          96600
6111 FORMAT(' COND CODE 6111: FILLETS/ROUNDS MAY NOT BE INSERTED BETWEEN 96700
IN LINE SEGMENTS WITH SAME SLOPE, SEGMENTS',2I4)           96800
WRITE(IW,2035)                                              96900
STOP 6111                                                    97000
6120 S=1.                                                     97100
IF(TQ(IP1,1).GT.TQ(I,2))S=-1.                                97200
BI=B(I)-S*RHO(I)/COS(PI0180*TQ(I,2))                        97300
BJ=B(IP1)-S*RHO(I)/COS(PI0180*TQ(IP1,1))                    97400
MI=TAN(PI0180*TH(I))                                         97500
M(IP1)=TAN(PI0180*TH(IP1))                                   97600
ETA(I)=(M(IP1)*BI-M(I)*BJ)/(M(IP1)-M(I))                    97700
KSI(I)=-(BJ-BI)/(M(IP1)-M(I))                                97800
X1(I)=KSI(I)-S*RHO(I)*SIN(PI0180*TH(I))                     97900
X2(I)=KSI(I)-S*RHO(I)*SIN(PI0180*TH(IP1))                   98000
Z1(I)=ETA(I)+S*RHO(I)*COS(PI0180*TH(I))                     98100
Z2(I)=ETA(I)+S*RHO(I)*COS(PI0180*TH(IP1))                   98200
T1(I)=TH(I)                                                  98300
T2(I)=TH(IP1)                                                 98400
GO TO 6800                                                    98500
C LINE-OGIVE FILLETTING/ROUNDING                               98600
6200 CONTINUE                                                98700
C OGIVE-LINE FILLETTING/ROUNDING                               98800
6300 CONTINUE                                                98900
6201 S1=0.                                                    99000
S2=0.                                                         99100
S3=0.                                                         99200
I1=I                                                           99300
IF(CURVE(IP1).EQ.ELLX)I1=IP1                                  99400
I2=I                                                           99500
IF(I1.EQ.I)I2=IP1                                             99600
IF(TQ(I,2)-TQ(IP1,1))6210,6205,6230                          99700
6205 WRITE(IW,6206)I,IP1                                      99800
6206 FORMAT(' COND CODE 6206: FILLETS/ROUNDS MAY NOT BE INSERTED BETWEEN 99900
IN SEGMENTS WITH SAME SLOPE, SEGMENTS',2I3)                 100000
WRITE(IW,2035)                                              100100
STOP 6206                                                    100200
C TH(IP1)>TH(I)                                                 100300
6210 IF(K(I1)-ZQ(IP1))6220,6215,6225                         100400
6215 WRITE(IW,6216)I,IP1                                      100500
6216 FORMAT(' COND CODE 6216: K = Z NOT A PROGRAMMED OPTION, SEGMENTS', 100600
1 2I3)                                                       100700
WRITE(IW,2035)                                              100800

```

Table 8. Continued

	STOP 6216	100900
6220	S1=1.	101000
	S2=1.	101100
	S3=1.	101200
	GO TO 6250	101300
6225	S1=1.	101400
	S2=-1.	101500
	S3=1.	101600
	GO TO 6250	101700
6230	IF (K(I1)-Z0(IPI))6240,6235,6245	101800
C	TH(IPI)<TH(I)	101900
6240	S1=-1.	102000
	S2=-1.	102100
	S3=-1.	102200
	GO TO 6250	102300
6235	WRITE(IW,6236)I,IPI	102400
6236	FORMAT(' COND CODE 6236: K = Z NOT A PROGRAMMED OPTION, SEGMENTS',	102500
	1 2I3)	102600
	WRITE(IW,2035)	102700
	STOP 6236	102800
6245	S1=-1.	102900
	S2=1.	103000
	S3=-1.	103100
	GO TO 6250	103200
6250	CONTINUE	103300
	B1=B(I2)+S1*RHO(I)/COS(PI0180*TH(I2))	103400
	A1=1.+M(I2)**2	103500
	A2=2.*((B1-K(I1))*M(I2)-H(I1))	103600
	A3=M(I1)**2*(B1-K(I1))**2-(R(I1)+S2*RHO(I))**2	103700
	A4=0.5*A2/A1	103800
	A5=A3/A1	103900
	A5=A4**2-A5	104000
	IF (A5.GE.0.)GO TO 6260	104100
	IF (ABS(A5).GT.ERRMAX)GO TO 6254	104200
	A5=0.	104300
	GO TO 6260	104400
6254	WRITE(IW,6255)I,IPI	104500
6255	FORMAT(' COND CODE 6255: REAL SOLUTION TO FILLET/ROUND PROBLEM DO	104600
	ES NOT EXIST. POSSIBLE PROGRAM ERROR. SEGMENTS ',2I2)	104700
	WRITE(IW,2035)	104800
	STOP 6255	104900
6260	A5=SQRT(A5)	105000
	A1=A4-A5	105100
	A2=M(I2)*A1*B1	105200
	A6=M(I1)+R(I1)*(A1-H(I1))/SQRT((A1-H(I1))**2+(A2-K(I1))**2)	105300
	A7=(M(I2)*(A2-B(I2))+A1)/(1.+M(I2)**2)	105400
	IF (X(I).GE.AMIN1(A6,A7).AND.X(I).LE.AMAX1(A6,A7))GO TO 6265	105500
	A1=A4+A5	105600
	A2=M(I2)*A1*B1	105700
	A6=M(I1)+R(I1)*(A1-H(I1))/SQRT((A1-H(I1))**2+(A2-K(I1))**2)	105800
	A7=(M(I2)*(A2-B(I2))+A1)/(1.+M(I2)**2)	105900
	IF (X(I).GE.AMIN1(A6,A7).AND.X(I).LE.AMAX1(A6,A7))GO TO 6265	106000
	WRITE(IW,6261)I	106100
6261	FORMAT(' COND CODE 6261: PROGRAM UNABLE TO CHOOSE BETWEEN TWO FILL	106200
	LET/ROUND SOLUTIONS, SEGMENT',I3)	106300
	WRITE(IW,2035)	106400

Table 8. Continued

```

        STOP 6261                                106500
6265 KSI(I)=A1                                    106600
      ETA(I)=A2                                    106700
      IF (CURVE(I).EQ.ELLX) GO TO 6270            106800
      X1(I)=A7                                      106900
      X2(I)=A6                                      107000
      Z2(I)=K(I1)+R(I1)*(ETA(I)-K(I1))/SQRT((A1-H(I1))**2+(A2-K(I1))**2) 107100
      Z1(I)=M(I2)*X1(I)+B(I2)                    107200
      T1(I)=TH(I)                                  107300
      T2(I)=-ATAN((X2(I)-H(I1))/(Z2(I)-K(I1)))/PI0180 107400
      GO TO 6800                                    107500
6270 X1(I)=A6                                      107600
      X2(I)=A7                                      107700
      Z2(I)=M(I2)*X2(I)+B(I2)                    107800
      Z1(I)=K(I1)+R(I1)*(ETA(I)-K(I1))/SQRT((A1-H(I1))**2+(A2-K(I1))**2) 107900
      T1(I)=-ATAN((X1(I)-H(I1))/(Z1(I)-K(I1)))/PI0180 108000
      T2(I)=TH(IP1)                                108100
      GO TO 6800                                    108200
6600 CONTINUE                                    108300
C OGIVE-OGIVE FILLETING ROUNDING                108400
      A1=TQ(IP1,1)-TQ(I,2)                        108500
      A2=Z(I)-K(I)                                108600
      A3=Z(I)-K(IP1)                              108700
      IF (A1.GT.ZERO.AND.A2.GT.ZERO.AND.A3.GT.ZERO) GO TO 6610 108800
      IF (A1.LT.ZERO.AND.A2.GT.ZERO.AND.A3.GT.ZERO) GO TO 6615 108900
      IF (A1.LT.ZERO.AND.A2.LT.ZERO.AND.A3.LT.ZERO) GO TO 6610 109000
      IF (A1.GT.ZERO.AND.A2.LT.ZERO.AND.A3.LT.ZERO) GO TO 6615 109100
      IF (A1.LT.ZERO.AND.A2.GT.ZERO.AND.A3.LT.ZERO) GO TO 6620 109200
      IF (A1.GT.ZERO.AND.A2.GT.ZERO.AND.A3.LT.ZERO) GO TO 6625 109300
      IF (A1.GT.ZERO.AND.A2.LT.ZERO.AND.A3.GT.ZERO) GO TO 6620 109400
      IF (A1.LT.ZERO.AND.A2.LT.ZERO.AND.A3.GT.ZERO) GO TO 6625 109500
      WRITE(IW,6601) I,IP1,A1,A2,A3              109600
6601 FORMAT(' COND CODE 6601: SEGMENTS',2I3,',', A1,A2,A3 MAY NOT BE ZERO 109700
      1, A1,A2,A3=',',1P3E15.7)                 109800
      WRITE(IW,2035)                              109900
      STOP 6601                                    110000
6610 S1=1.                                        110100
      S2=1.                                        110200
      GO TO 6630                                    110300
6615 S1=-1.                                       110400
      S2=-1.                                       110500
      GO TO 6630                                    110600
6620 S1=-1.                                       110700
      S2=1.                                        110800
      GO TO 6630                                    110900
6625 S1=1.                                        111000
      S2=-1.                                       111100
      GO TO 6630                                    111200
6630 CONTINUE                                    111300
      RI=R(I)*S1*RH0(I)                          111400
      RJ=R(IP1)*S2*RH0(I)                        111500
      IF (K(I).EQ.K(IP1)) GO TO 6645              111600
      MI=(H(IP1)-H(I))/(K(IP1)-K(I))             111700
      BI=(H(IP1)**2-H(I)**2+K(IP1)**2-K(I)**2-RJ**2+RI**2)/2./(K(IP1)- 111800
      1 K(I))                                       111900
      A1=1.+MI**2                                  112000

```

Table 8. Continued

```

      A2=2.*((B1-K(I))*M1-H(I))
      A3=H(I)**2*(B1-K(I))*2-R1**2
      A4=.5*A2/A1
      A5=A3/A1
      A5=A4**2-A5
      IF (A5.GE.0.) GO TO 6635
      IF (ABS(A5).LE.ERRMAX) GO TO 6634
      WRITE(IW,6631) I, IP1
6631 FORMAT(' COND CODE 6631: REAL SOLUTION TO FILLET/ROUND PROBLEM DOES
      IS NOT EXIST. POSSIBLE PROGRAM ERROR, SEGMENTS',2I3)
      WRITE(IW,2035)
      STOP 6631
6634 A5=0.
6635 A5=SQR(A5)
      A1=-A4*A5
      A2=-A4*A5
      A3=M1*A1*.8I
      A4=M1*A2*.8I
6636 A5=SQR((X(I)-A1)**2+(Z(I)-A3)**2)
      A6=SQR((X(I)-A2)**2+(Z(I)-A4)**2)
      IF (A6.LT.A5) GO TO 6640
      KSI(I)=A1
      ETA(I)=A3
      GO TO 6650
6640 KSI(I)=A2
      ETA(I)=A4
      GO TO 6650
6645 A1=(H(IP1)**2-H(I)**2-RJ**2+RI**2)/2./(H(IP1)-H(I))
      A2=A1
      A4=RI**2-(A1-H(I))**2
      IF (A4.GE.0.) GO TO 6648
      IF (ABS(A4).GT.ERRMAX) GO TO 6646
      A4=0.
6648 A4=SQR(A4)
      A3=K(I)-A4
      A4=K(I)+A4
      GO TO 6636
6646 WRITE(IW,6647) I, IP1
6647 FORMAT(' COND CODE 6647: FOR SEGMENTS',2I3,' K(I) = K(IP1); REAL S
      OLUTION TO FILLET/ROUND PROBLEM DOES NOT EXIST. POSSIBLE PROGRAM E
      ROR.')
      WRITE(IW,2035)
      STOP 6647
6650 DO 6660 J=I, IP1
      A1=SQR((KSI(I)-H(J))**2+(ETA(I)-K(J))**2)
      A2=H(J)+R(J)*(KSI(I)-H(J))/A1
      A3=K(J)+R(J)*(ETA(I)-K(J))/A1
      A4=-ATAN((A2-H(J))/(A3-K(J)))/PI*180
      IF (J.EQ.IP1) GO TO 6655
      X1(I)=A2
      Z1(I)=A3
      T1(I)=A4
      GO TO 6660
6655 X2(I)=A2
      Z2(I)=A3
      T2(I)=A4

```

Table 8. Continued

```

6660 CONTINUE
6800 CONTINUE
      IF(I$KIP.EQ.0)RETURN
C PRINT RESULTS OF FILLET/ROUND COMPUTATION
      WRITE(IW,6810)
6810 FORMAT('0FILLET/ROUND EQUATIONS AND INTERSECTIONS:/
1'0', 7X,' X ',10X,'KSI',10X,'ETA',10X,'RHO',10X,' X1',10X,' Z1',
2 10X,' T1',10X,' X2',10X,' Z2',10X,' T2'/
3' ',132('=-'))
      WRITE(IW,6815) (I,X(I),KSI(I),ETA(I),RHO(I),X1(I),Z1(I),T1(I),
1 X2(I),Z2(I),T2(I),I=1,10)
6815 FORMAT(10(' ',I2,1P10E13.5/))
C INSERT FILLET/ROUND DATA INTO QUICK ARRAYS
6900 CONTINUE
      J=0
      DO 6920 I=1,10
      J=J+1
      IF(RHO(I).GT.RINF)GO TO 6920
      J=J+1
      NSEG=NSEG+1
      NSEGP=NSEG+1
      IF(NSEG.LE.10)GO TO 6905
      WRITE(IW,6901)I
6901 FORMAT('0 COND CODE 6901: INSERTION OF ROUND',I3,' EXCEEDS DIMENSIO
INS (10) OF QUICK')
      WRITE(IW,2035)
      STOP 6901
6905 L=1
6910 L=L-1
      LM=L-1
      IF(L.LE.J)GO TO 6915
      XQ(L)=XQ(LM)
      ZQ(L)=ZQ(LM)
      TQ(L,1)=TQ(LM,1)
      TQ(L,2)=TQ(LM,2)
      CURVE(L)=CURVE(LM)
      GO TO 6910
6915 XQ(J)=X1(I)
      ZQ(J)=Z1(I)
      TQ(J,1)=T1(I)
      TQ(J,2)=T2(I)
      TQ(J-1,2)=T1(I)
      CURVE(J-1)=CURVE(J)
      XQ(J-1)=X2(I)
      ZQ(J-1)=Z2(I)
      TQ(J-1,1)=T2(I)
      CURVE(J)=ELLY
6920 CONTINUE
C PRINT RESULTS
      WRITE(IW,6930)
6930 FORMAT('0FINAL QUICK ARRAY:/
      WRITE(IW,7010)
      DO 6940 I=1,10
6940 WRITE(IW,90008)I,XQ(I),ZQ(I),TQ(I,2),TQ(I,1)
      RETURN
2035 FORMAT('0EXECUTION HALTED')

```

Table 8. Continued

```

7010 FORMAT('0',9 X,'X',13X,'Z',12X,'TA',12X,'TF',' ',58('-')) 123300
90008 FORMAT(' ',12,1P9E14.6) 123400
END 123500
C***** 123600
SUBROUTINE QUICK 123700
C----- 123800
C SET UP QUICK MODEL 123900
C----- 124000
COMMON /RA/X(10),Z(10),TM(10),M(10),K(10),R(10),B(10),N(10), 124100
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10), 124200
2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2), 124300
3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3) 124400
4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10), 124500
5 BLB(8) 124600
COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8) 124700
COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX, 124800
1 YUCL,ZERO,FORE,AFFE,TOPS,BOTS,OPUP,OGIV,PI0180,XJ,ZJ,XI,ZI,M1, 124900
2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,A1,MJ,RJ,A4,DX 125000
3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO 125100
COMMON /IS/IR,IR,IF,IP,IPRINT,NSEG,NSEGPI,ISTOP,JSTOP,KSTOP, 125200
1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3 125300
COMMON /LA/SLOPE(10),IDEN(80),CARD(80) 125400
COMMON /LS/TI,T 125500
REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV 125600
LOGICAL=1 SLOPE,IDEN,CARD,TI,T 125700
8000 CONTINUE 125800
IF(IF.NE.IP)REWIND IF 125900
C CROSS SECTION MODELS 126000
WRITE(IF,90001)IDEN 126100
IF(ISL.NE.0)GO TO 8001 126200
WRITE(IF,90002) 126300
WRITE(IF,90003) 126400
GO TO 8002 126500
8001 WRITE(IF,21700) 126600
8002 CONTINUE 126700
I1=1 126800
WRITE(IF,90004)I1,I1,XQ(1),XQ(NSEGPI) 126900
C BODY LINE MODELS 127000
WRITE(IF,90005)YUCL 127100
WRITE(IF,90006)I1,KRV(1),KV(1) 127200
WRITE(IF,90007)ZERO,ZERO,XQ(NSEGPI),ZERO,ZERO,ZERO 127300
WRITE(IF,90009) 127400
DO 8030 J=1,2 127500
WRITE(IF,90005)CL(J) 127600
DO 8010 I=1,NSEG 127700
IP1=I+1 127800
I1=1 127900
IF(CURVE(I).EQ.ELLX)I1=2 128000
WRITE(IF,90006)I,CURVE(I),KV(I1) 128100
GO TO (8006,8005),I1 128200
8005 CONTINUE 128300
WRITE(IF,90007)XQ(I),ZQ(I),XQ(IP1),ZQ(IP1),TQ(I,1),TQ(I,2) 128400
GO TO 8010 128500
8006 WRITE(IF,90014)XQ(I),ZQ(I),XQ(IP1),ZQ(IP1) 128600
8010 CONTINUE 128700
WRITE(IF,90009) 128800

```

Table 8. Continued

DO 8020 I=1,NSEGPI	128900
ZQ(I)=-ZQ(I)	129000
TQ(I,1)=-TQ(I,1)	129100
TQ(I,2)=-TQ(I,2)	129200
8020 CONTINUE	129300
8030 CONTINUE	129400
WRITE(IF,90010)	129500
IF(ISL.EQ.0)GO TO 8033	129600
DO 8032 L=1,3	129700
WRITE(IF,90005)SL(L)	129800
I1=NS(L)	129900
DO 8031 I=1,I1	130000
I2=1	130100
IF(CRVSL(I,L).EQ.ELLX)I2=2	130200
WRITE(IF,90006)I,CRVSL(I,L),KV(I2)	130300
WRITE(IF,90007)V(I,1,L),V(I,2,L),V(I+1,1,L),V(I+1,2,L),V(I,3,L),	130400
1 V(I+1,3,L)	130500
8031 CONTINUE	130600
WRITE(IF,90009)	130700
8032 CONTINUE	130800
WRITE(IF,21900)	130900
8033 CONTINUE	131000
WRITE(IF,90011)	131100
IF(IPRINT.EQ.0)GO TO 8050	131200
DX=.00001	131300
NPTS=5	131400
C SET UP CARDS TO EXERCISE MODEL	131500
I1=1	131600
I2=2	131700
DO 8040 I=1,NSEG	131800
A1=XQ(I)*DX	131900
A2=XQ(I+1)*DX	132000
A3=(A2-A1)/(NPTS-1)	132100
A2=XQ(I+1)*DX	132200
WRITE(IF,90012)I1,I2,A1,A2,A3	132300
8040 CONTINUE	132400
IF(ISL.EQ.0)GO TO 8050	132500
I1=2	132600
I2=2	132700
I4=NP(I)	132800
A3=2.	132900
A4=-90.	133000
A5=90.	133100
A6=5.	133200
DO 8044 L=1,3	133300
I3=NP(L)	133400
DO 8043 I=2,I3	133500
IF(L.EQ.1)GO TO 8042	133600
IF(I.EQ.J)GO TO 8043	133700
DO 8041 J=1,I4	133800
IF(ABS(V(J,1,1)-V(I,1,L)).LE.ERRMAX)GO TO 8043	133900
8041 CONTINUE	134000
8042 A1=V(I,1,L)	134100
A2=A1*1.	134200
WRITE(IF,90012)I1,I2,A1,A2,A3,A4,A5,A6	134300
8043 CONTINUE	134400

Table 8. Continued

```

8044 CONTINUE                                     134500
8050 WRITE(1F,90013)                             134600
      RETURN                                     134700
C-----
C QUICK INPUT FORMATS                          134800
C-----
90001 FORMAT(80A1)                               134900
90002 FORMAT(' 1'/' 1 2')                       135000
90003 FORMAT('LELL 1ELLI PIEC BLCL SID LSCP'/ 135100
      1 'UELL 2ELLI PIEC SID TUCL USCP'/
      2 ' 1 MAPAX')                             135200
90004 FORMAT(2I2,6X,2F10,5)                     135300
90005 FORMAT(A4)                                 135400
90006 FORMAT(I2,1X,A4,' PIEC ',A4)              135500
90007 FORMAT(3F10,5,2(F9,5,'A'),F10,5)          135600
90008 FORMAT(' ',I2,1P9E14,6)                   135700
90009 FORMAT('=-1')                              135800
90010 FORMAT('YLCL YUCL'/
      1 'YMAPAX YUCL'/
      2 'ZMAPAX YUCL'/
      3 'ZSID YUCL'/
      4 'YUSCP ZUCL'/
      5 'ZUSCP ZUCL'/
      6 'YLSCP ZUCL'/
      7 'YSID ZUCL'/
      8 'ZLSCP ZLCL')                             135900
90011 FORMAT(' ')                               136000
90012 FORMAT(I2,I3,5X,6F10,5)                   136100
90013 FORMAT(' 0')                              136200
90014 FORMAT(3F10,5,F9,5)                       136300
21700 FORMAT(' 1',78X/
      1' 1 6 NONCIRCULAR CROSS SECTION ',21X/
      2'LELL 1ELLI PIEC BLCL SID LSCP ',21X/
      3'UELL 2ELLI PIEC SID TUCL USCP ',21X/
      4'LBSL 3LINE PIEC BLSL LSLCP
      5LELL',11X/
      6'LSSL 4LINE PIEC LSLCP SSL LELL',21X/
      7'USSL 5LINE PIEC SSL USLCP
      8UELL',11X/
      9'UTSL 6LINE PIEC USLCP TUSL UELL',21X/
      A ' 1 MAPAX')
21900 FORMAT('YUSL YUCL',66X/
      1 'YLSL YUCL',66X/
      6 'YUSLCP YSSL',66X/
      7 'YLSLCP YSSL',66X/
      A 'ZUSLCP ZUSL',66X/
      8 'ZLSLCP ZLSL',66X/
      C 'ZSSL ZSID',66X)
      END
C-----
SUBROUTINE LIST                                139300
C-----
C LIST QUICK INPUT FILE                       139400
C-----
COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),M(10),
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),

```

Table 8. Continued

```

2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),      140100
3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)                      140200
4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),  140300
5 BLB(8)                                                                140400
COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)             140500
COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,           140600
1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,MI,      140700
2 RI,BI,THI,8J,A1,A2,A3,8IJ,THJ,S,THIJ,SIJ,AI,MJ,RJ,A4,DX             140800
3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO                    140900
COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,          141000
1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3                             141100
COMMON /LA/SLOPE(10),IDEN(80),CARD(80)                                141200
COMMON /LS/TT,T                                                         141300
REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV                             141400
LOGICAL*1 SLOPE,IDEN,CARD,TT,T                                         141500
IF(IF.EQ.IP)GO TO 9060                                                  141600
9000 CONTINUE                                                            141700
REWIND IF                                                                141800
WRITE(IW,9010)IF                                                        141900
9010 FORMAT('O LISTING OF QUICK INPUT DECK ON UNIT ',I2,':')          142000
WRITE(IW,2007)                                                          142100
9020 READ(IF,90001,END=9040)IDEN                                         142200
WRITE(IW,2009)IDEN                                                      142300
9030 FORMAT(' ',80A1)                                                    142400
GO TO 9020                                                              142500
9040 WRITE(IW,2007)                                                      142600
9050 FORMAT(' ',132(' ',))                                               142700
WRITE(IW,9055)                                                          142800
9055 FORMAT('1')                                                         142900
9060 RETURN                                                              143000
90001 FORMAT(80A1)                                                       143100
2007 FORMAT(' ',82(' ',))                                                143200
2009 FORMAT(' ',80A1,'1')                                               143300
END                                                                      143400
C-----                                                                143500
SUBROUTINE SLCE                                                           143600
C-----                                                                143700
C COMPUTE SLICE GEOMETRY                                                143800
C-----                                                                143900
COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),M(10),          144000
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),        144100
2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),        144200
3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)                      144300
4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),  144400
5 BLB(8)                                                                144500
COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)             144600
COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,           144700
1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,MI,      144800
2 RI,BI,THI,8J,A1,A2,A3,8IJ,THJ,S,THIJ,SIJ,AI,MJ,RJ,A4,DX             144900
3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO                    145000
COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,          145100
1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3                             145200
COMMON /LA/SLOPE(10),IDEN(80),CARD(80)                                145300
COMMON /LS/TT,T                                                         145400
REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV                             145500
LOGICAL*1 SLOPE,IDEN,CARD,TT,T                                         145600

```

Table 8. Continued

```

10000 CONTINUE
      IF(TYPE.NE.ASYM)GO TO 10006
C SET UP XQ AND ZQ FOR ASYM CASE
      NSEGP1=1
      XQ(1)=VB(NB(1)+1,1,1,1)
      A1=0.
      DO 10005 L=1,3
        I1=NB(L)+1
        DO 10005 I=1,I1
          A1=AMAX1(A1,VB(I,2,1,L))
10005 CONTINUE
      ZQ(1)=A1
10006 CONTINUE
C DETERMINE IF SLICES ARE PRESENT
      ISL=0
      DO 10010 I=1,10
        DO 10010 J=1,3
          DO 10010 L=1,3
            IF(V(I,J,L).GT.RINF)GO TO 10010
            ISL=1
            GO TO 10015
10010 CONTINUE
      IF(TYPE.EQ.ASYM.AND.ISL.EQ.0)GO TO 10015
      IF(ISL.EQ.0)RETURN
C DETERMINE NUMBER OF SEGMENTS IN EACH SLICE NS(L)=NP(L)-1
10015 DO 10020 L=1,3
      NP(L)=0
      DO 10020 I=1,10
        DO 10020 J=1,3
          IF(V(I,J,L).LT.RINF)NP(L)=1
10020 CONTINUE
      DO 10030 L=1,3
        NS(L)=NP(L)-1
10030 CONTINUE
C CHECK INPUT FOR SUFFICIENCY
      DO 10060 L=1,3
        I1=NP(L)
        IF(I1.EQ.0)GO TO 10060
        DO 10055 I=1,I1
          I2=0
          DO 10040 J=1,3
            IF(V(I,J,L).LT.RINF)I2=I2+1
10040 CONTINUE
        IF(I2.GE.2)GO TO 10055
        ISTOP=1
        WRITE(IW,10050)L,I
10050 FORMAT(' COND CODE 0050: SLICE',I2,'. POINT',I3,'. DATA INSUFFICIE
INT')
10055 CONTINUE
10060 CONTINUE
      IF(ISTOP.EQ.0)GO TO 10070
      WRITE(IW,2035)
      2035 FORMAT(' EXECUTION HALTED')
      STOP 0050
C COMPUTE REMAINING SLICE GEOMETRY
10070 DO 10160 L=1,3

```

Table 8. Continued

I1=NP(L)	151300
IF(I1.EQ.0)GO TO 10160	151400
DO 10150 I=2,I1	151500
IM1=I-1	151600
DO 10080 J=1,3	151700
IF(V(I,J,L).GT.RINF)GO TO(10100,10130,10140),J	151800
10080 CONTINUE	151900
A1=(V(I,2,L)-V(IM1,2,L))/(V(I,1,L)-V(IM1,1,L))	152000
A1=ATAN(A1)/PI0180	152100
IF(ABS(A1-V(I,3,L)).LE.ERRMAX)GO TO 10150	152200
ISTOP=1	152300
WRITE(IW,10090)L,I,ERRMAX	152400
10090 FORMAT(' COND CODE 0090: SLICE',I2,', POINT',I3,', REDUNDANT INPUT	152500
1 DATA NOT SELF CONSISTENT TO WITHIN ',1PE15.7)	152600
GO TO 10150	152700
10100 IF(V(I,3,L).EQ.0.)GO TO 10110	152800
V(I,1,L)=(V(I,2,L)-V(IM1,2,L))/TAN(PI0180*V(I,3,L))+V(IM1,1,L)	152900
GO TO 10150	153000
10110 WRITE(IW,10120)L,I	153100
10120 FORMAT(' COND CODE 0120: SLICE',I2,', POINT',I3,', X MUST BE INPUT	153200
1 WHEN SLOPE IS ZERO')	153300
WRITE(IW,2035)	153400
STOP 0120	153500
10130 V(I,2,L)=V(IM1,2,L)+(V(I,1,L)-V(IM1,1,L))*TAN(PI0180*V(I,3,L))	153600
GO TO 10150	153700
10140 V(I,3,L)=ATAN((V(I,2,L)-V(IM1,2,L))/(V(I,1,L)-V(IM1,1,L)))/PI0180	153800
10150 CONTINUE	153900
10160 CONTINUE	154000
IF(ISTOP.EQ.0)GO TO 10170	154100
WRITE(IW,2035)	154200
STOP 0090	154300
C EXTRAPOLATE BEGINNING OF FIRST SEGMENT TO X=0	154400
10170 DO 10180 L=1,3	154500
I1=NP(L)	154600
IF(I1.EQ.0)GO TO 10180	154700
IF(V(I,1,L).LE.0.) GO TO 10180	154800
V(I,2,L)=V(I,2,L)-V(I,1,L)*TAN(PI0180*V(I,3,L))	154900
V(I,1,L)=0.	155000
10180 CONTINUE	155100
C EXTRAPOLATE END OF LAST SEGMENT TO XQ(NSEGP1)	155200
DO 10190 L=1,3	155300
I1=NP(L)	155400
IF(I1.EQ.0)GO TO 10190	155500
IF(V(I,1,L).GE.XQ(NSEGP1))GO TO 10190	155600
V(I,2,L)=V(I,2,L)+(XQ(NSEGP1)-V(I,1,L))*TAN(PI0180*V(I,3,L))	155700
V(I,1,L)=XQ(NSEGP1)	155800
10190 CONTINUE	155900
C PLACE NON EXISTENT SLICES OUTSIDE OF VEHICLE	156000
A1=0.	156100
DO 10200 I=1,NSEGP1	156200
A1=AMAX1(A1,ABS(ZQ(I)))	156300
10200 CONTINUE	156400
A1=2.*A1	156500
DO 10210 L=1,3	156600
IF(NP(L).NE.0)GO TO 10210	156700
NP(L)=2	156800

Table 8. Continued

```

      NS(L)=1
      V(1,1,L)=0.
      V(1,2,L)=A1
      V(2,1,L)=XQ(NSEGP1)
      V(2,2,L)=A1
      V(1,3,L)=0.
      V(2,3,L)=0.
10210 CONTINUE
C BE SURE V(1,3,L) IS DEFINED
      DO 10220 L=1,3
      IF(V(1,3,L).GT.RINF)V(1,3,L)=V(2,3,L)
10220 CONTINUE
C PRINT RESULTS OF SLICE GEOMETRY CALCULATIONS
      WRITE(IW,10230)
10230 FORMAT('0SLICE GEOMETRY:','0',21X,'UPPER',39X,'LOWER',39X,'SIDE '
1/ 9X,'X',13X,'Z',13X,'T',15X,'X',13X,'Z',13X,'T',15X,'X',13X,
2'Y',13X,'T',1/ 132(' '))
      DO 10250 I=1,10
      WRITE(IW,10240)I,((V(I,J,L),J=1,3),L=1,3)
10240 FORMAT(' ',I2,2(1P3E14.6,2X),1P3E14.6)
10250 CONTINUE
C CHECK VALIDITY OF COORDINATES
      DO 10260 L=1,3
      I1=NP(L)
      DO 10270 I=1,11
      IF(V(I,2,L).GE.0.)GO TO 10270
      ISTOP=1
      WRITE(IW,10260)L,I
10260 FORMAT(' COND CODE 0260: SLICE',I2,', POINT',I3,', Z VALUE < 0. SL
      ICES MAY NOT CROSS VEHICLE CL.')
10270 CONTINUE
10280 CONTINUE
      IF(ISTOP.EQ.0)GO TO 10290
      WRITE(IW,2035)
      STOP 0260
C RESIGN LSL FOR QUICK INPUT
10290 I1=NP(2)
      IF(TYPE.EQ.ASYM)RETURN
      DO 10300 I=1,11
      DO 10300 J=2,3
      V(I,J,2)=-V(I,J,2)
10300 CONTINUE
      RETURN
      END
C-----
SUBROUTINE RFSL
C-----
C INSERT ROUNDS OR FILLETS BETWEEN SEQUENTIAL SLICING PLANES
C-----
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
3 KV(2),V(10,3,3),SL(3),R-OSL(10,3),CRVSL(10,3)
4 VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)

```

Table 8. Continued

```

COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,      162500
1 YUCL,ZERO,FORE,AFFE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,MI,    162600
2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,MJ,RJ,A4,DX        162700
3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,ONTO,DNBO              162800
COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,      162900
1 I,J,L,IP1,IM1,I1,NPTS,I2,ISL,LB,N,NM1,I3                        163000
COMMON /LA/SLOPE(I0),IDEN(80),CARD(80)                            163100
COMMON /LS/TI,T                                                    163200
REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV                        163300
LOGICAL*1 SLOPE,IDEN,CARD,TI,T                                    163400
11000 CONTINUE                                                    163500
      ISKIP=0                                                       163600
      IF (ISL.EQ.0) RETURN                                          163700
C DETERMINE IF ROUNDS OR FILLETS ARE PRESENT                       163800
      IRFSL = 0                                                     163900
      DO 11005 L=1,3                                                164000
      DO 11005 I=1,10                                               164100
      IF (RHOSL(I,L).GT.RINF) GO TO 11005                          164200
      IRFSL=1                                                       164300
      GO TO 11006                                                    164400
11005 CONTINUE                                                    164500
      RETURN                                                        164600
11006 WRITE(IW,11007)                                              164700
11007 FORMAT('OSLICE ROUND/FILLET CENTERS AND INTERSECTIONS:')   164800
      1 /'0', 9X,'BI',11X,'BJ',11X,'A3',11X,'A4',11X,'X1',11X,'Z1', 164900
      1 11X,'T1',11X,'X2',11X,'Z2',11X,'T2',/' ',                165000
      2 132(' ')                                                    165100
C COMPUTE ROUND/FILLET CENTERS AND INTERSECTIONS                  165200
      DO 11130 L=1,3                                                165300
      DO 11030 I=2,9                                                165400
      IP1=I+1                                                        165500
      IF (RHOSL(I,L).GT.RINF) GO TO 11030                          165600
      ISKIP=1                                                        165700
      IF (ABS(V(I,3,L)-V(IP1,3,L)).GE.ERRMAX) GO TO 11020         165800
      WRITE(IW,11010) I,IP1                                         165900
11010 FORMAT(' COND CODE 1010: FILLETS/ROUNDS MAY NOT BE INSERTED BETWEE 166000
      IN LINE SEGMENTS WITH SAME SLOPE, SEGMENTS',2(I3))         166100
      WRITE(IW,2035)                                                166200
2035 FORMAT(' EXECUTION HALTED')                                    166300
      STOP 1010                                                     166400
11020 S=1.                                                         166500
      IF (V(IP1,3,L).GT.V(I,3,L)) S=-1.                            166600
      MI=TAN(PIO180*V(I,3,L))                                       166700
      MJ=TAN(PIO180*V(IP1,3,L))                                     166800
      A1=V(I,2,L)-MI*V(I,1,L)                                       166900
      A2=V(IP1,2,L)-MJ*V(IP1,1,L)                                   167000
      BI=A1-S*RHOSL(I,L)/COS(PIO180*V(I,3,L))                     167100
      BJ=A2-S*RHOSL(I,L)/COS(PIO180*V(IP1,3,L))                   167200
      A3=(MJ*BI-MI*BJ)/(MJ-MI)                                       167300
      A4=-(BJ-BI)/(MJ-MI)                                           167400
      X1(I)=A4-S*RHOSL(I,L)*SIN(PIO180*V(I,3,L))                 167500
      X2(I)=A4-S*RHOSL(I,L)*SIN(PIO180*V(IP1,3,L))               167600
      Z1(I)=A3-S*RHOSL(I,L)*COS(PIO180*V(I,3,L))                 167700
      Z2(I)=A3-S*RHOSL(I,L)*COS(PIO180*V(IP1,3,L))               167800
      T1(I)=V(I,3,L)                                                 167900
      T2(I)=V(IP1,3,L)                                              168000

```

Table 8. Continued

```

WRITE(IW,11025)L,I,8I,8J,A3,A4,X1(I),Z1(I),T1(I),X2(I),Z2(I),T2(I) 168100
11025 FORMAT(1X,Z1I,1P10E13,5) 168200
11030 CONTINUE 168300
C INSERT ROUNDS/FILLET INTO QUICK ARRAYS 168400
DO 11035 I=1,9 168500
  IP1=I+1 168600
  RHOSL(I,L)=RHOSL(IP1,L) 168700
  X1(I)=X1(IP1) 168800
  Z1(I)=Z1(IP1) 168900
  T1(I)=T1(IP1) 169000
  X2(I)=X2(IP1) 169100
  Z2(I)=Z2(IP1) 169200
  T2(I)=T2(IP1) 169300
11035 CONTINUE 169400
  RHOSL(10,L)=1.E70 169500
  J=0 169600
  DO 11110 I=1,10 169700
    J=J+1 169800
    IF(RHOSL(I,L).GT.RINF)GO TO 11110 169900
    J=J+1 170000
    NS(L)=NS(L)+1 170100
    NP(L)=NP(L)+1 170200
    IF(NP(L).LE.10)GO TO 11050 170300
    WRITE(IW,11040)I 170400
11040 FORMAT(' COND CODE 1040: INSERTION OF ROUND/FILLET',I3,' EXCEEDS D
  1MENSIONS (10)') 170500
    WRITE(IW,2035) 170600
    STOP 1040 170700
11050 N=1 170800
11060 N=N+1 170900
    NM1=N-1 171000
    IF(N.LE.J)GO TO 11080 171100
    DO 11070 I2=1,3 171200
11070 V(N,I2,L)=V(NM1,I2,L) 171300
    CRVSL(N,L)=CRVSL(NM1,L) 171400
    GO TO 11060 171500
11080 CONTINUE 171600
    V(J,1,L)=X1(I) 171700
    V(J,2,L)=Z1(I) 171800
    V(J,3,L)=T1(I) 171900
    CRVSL(J,1,L)=CRVSL(J,L) 172000
    V(J,1,1,L)=X2(I) 172100
    V(J,1,2,L)=Z2(I) 172200
    V(J,1,3,L)=T2(I) 172300
    CRVSL(J,L)=ELLX 172400
11110 CONTINUE 172500
    WRITE(IW,11120) 172600
11120 FORMAT(' ') 172700
11130 CONTINUE 172800
C PRINT FINAL QUICK ARRAYS 172900
WRITE(IW,10230) 173000
10230 FORMAT('OFINAL QUICK ARRAYS FOR ROUNDED/FILLETED SLICES:',
  1 /'0',Z1X,'UPPER',39X,'LOWER',39X,'SIDE ' 173100
  1 / 9X,'X',13X,'Z',13X,'T',15X,'X',13X,'Z',13X,'T',15X,'X',13X, 173200
  2 'Y',13X,'T'/' ',132(' -')) 173300
  DO 10250 I=1,10 173400
    173500
    173600

```

Table 8. Continued

```

WRITE(IW,10240)I,((V(I,J,L),J=1,3),L=1,3)
10240 FORMAT(' ',I2,2(1P3E14.6,2X),1P3E14.6)
10250 CONTINUE
RETURN
END
C-----
SUBROUTINE COPY
C-----
C COPY DATA INTO VB ARRAY FOR QK3D
C-----
COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),M(10),
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
3 KV(2),V(10,3,3),SL(3),R-OSL(10,3),CRVSL(10,3)
4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
5 BLB(8)
COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
1 YUCL,ZERO,FORE,AFFE,TOPS,BOTS,OPUP,OGIV,P10180,XJ,ZJ,XI,ZI,MI,
2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,A1,HJ,RJ,A4,DX
3 TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,ONFO,DNAF,DNTO,DNBO
COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGPI,ISTOP,JSTOP,KSTOP,
1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
COMMON /LS/TI,T
REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
LOGICAL*1 SLOPE,IDEN,CARD,TI,T
DIMENSION E(10,7)
EQUIVALENCE (E(1,1),X (1))
1 ,E(1,2),Z (1))
2 ,E(1,3),TH(1))
3 ,E(1,4),M (1))
4 ,E(1,5),K (1))
5 ,E(1,6),R (1))
6 ,E(1,7),B (1))
DATA RINF1/1.E70/
LOGICAL*1 F/'F'/
12000 CONTINUE
C
1 2 3 4 5 6 7 8
GO TO(12010,12010,12010,12050,12050,12050,12080,12080
1),LB
C LB=1 ZUCL, LB=2 ZLCL, LB=3 YSID
12010 NB(LB)=NSEG
DO 12020 I=1,10
VB(I,1,1,LB)=XQ(I)
VB(I,2,1,LB)=ZQ(I)
VB(I,3,1,LB)=TQ(I,1)
VB(I,3,2,LB)=TQ(I,2)
CRVB (I,LB)=CURVE(I)
12020 CONTINUE
DO 12040 I=1,10
XQ(I)=RINF1
ZQ(I)=RINF1
TQ(I,1)=RINF1
TQ(I,2)=RINF1
DO 12030 J=1,7

```

Table 8. Continued

	E(I,J)=RINF1	179300
12030	CONTINUE	179400
	SLOPE(I)=F	179500
	RHO(I)=RINF1	179600
	X1(I)=RINF1	179700
	X2(I)=RINF1	179800
	Z1(I)=RINF1	179900
	Z2(I)=RINF1	180000
	T1(I)=RINF1	180100
	T2(I)=RINF1	180200
	KSI(I)=RINF1	180300
	ETA(I)=RINF1	180400
	CURVE(I)=BLNK	180500
	DIR(I)=OPDN	180600
12040	CONTINUE	180700
	RETURN	180800
C	LB=4 ZUSL, LB=5 ZLSL, LB=5 YSSL	180900
12050	I1=LB-3	181000
	NB(LB)=NS(I1)	181100
	DO 12070 I=1,10	181200
	DO 12060 J=1,3	181300
	VB(I,J,1,LB)=V(I,J,I1)	181400
	CRVB(I,LB)=CRVSL(I,I1)	181500
12060	CONTINUE	181600
	IF(I.LT.10)VB(I,3,2,LB)=V(I+1,3,I1)	181700
12070	CONTINUE	181800
	RETURN	181900
12080	DO 12090 I=1,10	182000
	VB(I,1,1,7)=XSID(I)	182100
	VB(I,2,1,7)=ZSID(I)	182200
	VB(I,1,1,8)=XMAPAX(I)	182300
	VB(I,2,1,8)=ZMAPAX(I)	182400
12090	CONTINUE	182500
	RETURN	182600
	END	182700
C*****		182800
	SUBROUTINE BLCK	182900
C-----		183000
C	CHECK ALL BODY LINE COORDINATES FOR VALIDITY	183100
C	COMPUTE DEFAULT VALUES FOR ZSID AND ZMAPAX	183200
C-----		183300
	COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),M(10),	183400
1	CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),	183500
2	KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),	183600
3	KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)	183700
4	VB(10,3,2,8),CRVB(10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),	183800
5	BLB(8)	183900
	COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)	184000
	COMMON /RS/KI,MJ,KJ,MJ,MJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,	184100
1	YUCL,ZERO,FORE,AFFE,TUPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,MJ,	184200
2	KI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,MJ,RJ,A4,DX	184300
3	TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO	184400
	COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,	184500
1	I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3	184600
	COMMON /LA/SLOPE(10),IDEN(80),CARD(80)	184700
	COMMON /LS/TI,T	184800

Table 8. Continued

REAL K,M,KS1,KI,MI,KJ,MJ,NIJ,LINE,KV,KRV	184900
LOGICAL*1 SLOPE,IDEN,CARD,TI,T	185000
13000 CONTINUE	185100
C ZSID AND ZMAPAX DEFAULT	185200
DO 13060 L=7,8	185300
NB(L)=0	185400
DO 13010 I=1,10	185500
IF(VB(I,1,1,L).LT.RINF.AND.VB(I,2,1,L).LT.RINF)NB(L)=I	185600
13010 CONTINUE	185700
NB(L)=NB(L)-1	185800
IF(NB(L).GT.0)GO TO 13060	185900
IF(NB(L).EQ.-1)GO TO 13030	186000
WRITE(IW,13020)L	186100
13020 FORMAT(' COND CODE 3020: BODY LINE',I2,' REQUIRES X AND Z VALUES F	186200
OR TWO OR MORE POINTS')	186300
WRITE(IW,2035)	186400
2035 FORMAT(' EXECUTION HALTED')	186500
STOP 3020	186600
13030 IF(L.NE.7)GO TO 13040	186700
NB(7)=1	186800
VB(1,1,1,7)=0.	186900
VB(1,2,1,7)=0.	187000
VB(2,1,1,7)=XQ(NSEG01)	187100
VB(2,2,1,7)=0.	187200
GO TO 13060	187300
13040 I1=NB(7)+1	187400
NB(8)=NB(7)	187500
DO 13050 I=1,I1	187600
DO 13050 J=1,2	187700
VB(I,J,1,8)=VB(I,J,1,7)	187800
13050 CONTINUE	187900
13060 CONTINUE	188000
C VALIDITY CHECK	188100
A1=VB(NB(1)+1,1,1,1)	188200
DO 13080 L=2,8	188300
I1=NB(L)+1	188400
IF(ABS(VB(I1,1,1,L)-A1).LE.ERRMAX)GO TO 13080	188500
IF(VB(I1,1,1,L).GE.A1.AND.L.GE.4)GO TO 13080	188600
WRITE(IW,13070)ERRMAX,L	188700
13070 FORMAT(' COND CODE 3070: ALL BODY LINES MUST BE THE SAME LENGTH TO	188800
1 WITHIN ERRMAX=',1PE14.7,' BL',I2)	188900
WRITE(IW,2035)	189000
STOP 3070	189100
13080 CONTINUE	189200
DO 13100 L=1,8	189300
IF(ABS(VB(1,1,1,L)).LE.ERRMAX)GO TO 13100	189400
WRITE(IW,13090)ERRMAX,L,VB(1,1,1,L)	189500
13090 FORMAT(' COND CODE 3090: FIRST X OF EACH BODY LINE MUST BE ZERO (T	189600
10 WITHIN ERRMAX=',1PE14.7,') BL',I2,' X=',1PE14.7)	189700
WRITE(IW,2035)	189800
STOP 3090	189900
13100 CONTINUE	190000
C RESIGN ZLCL AND ZLSL	190100
DO 13110 L=2,5,3	190200
I1=NB(L)+1	190300
DO 13110 I=1,I1	190400

Table 8. Continued

```

DO 13110 J=2,3                                190500
DO 13110 N=1,2                                190600
IF(VB(I,J,N,L).GT.RINF)GO TO 13110            190700
VB(I,J,N,L)=-VB(I,J,N,L)                     190800
13110 CONTINUE                                190900
C SET UP ANGLES FOR ZMAPAX AND ZSID            191000
DO 14026 L=7,8                                191100
  I1=NB(L)                                    191200
  DO 14025 I=1,I1                             191300
    A1=ATAN((VB(I+1,2,1,L)-VB(I,2,1,L))/(VB(I+1,1,1,L)-VB(I,1,1,L))) 191400
    A1=A1/PI*180                               191500
    VB(I,3,1,L)=A1                             191600
    VB(I,3,2,L)=A1                             191700
  14025 CONTINUE                                191800
14026 CONTINUE                                191900
  RETURN                                       192000
  END                                         192100
C-----
SUBROUTINE QK3D                                192200
C-----
C SET UP QUICK MODEL FOR 3D CASE              192300
C-----
COMMON /RA/X(10),Z(10),TH(10),M(10),K(10),R(10),B(10),M(10), 192400
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10), 192500
2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2), 192600
3 KV(2),V(10,3,3),SL(3),R-OSL(10,3),CRVSL(10,3) 192700
4,VB(10,3,2,8),CRVB(10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10), 192800
5 BLB(8)                                       192900
COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8) 193000
COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERHMAX,OPDN,ELLX, 193100
1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,MI, 193200
2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,MJ,RJ,A4,DX 193300
3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO 193400
COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP, 193500
1 I,J,L,IP1,LMI,I1,NPTS,I2,ISL,LB,N,NMI,I3 193600
COMMON /LA/SLOPE(10),IDEN(80),CARD(80) 193700
COMMON /LS/TI,T 193800
REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV 193900
LOGICAL*1 SLOPE,IDEN,CARD,TI,T 194000
14000 CONTINUE                                194100
IF(IF.NE.IP)REWIND IF 194200
C CS MODELS 194300
WRITE(IF,90001)IDEN 194400
WRITE(IF,21700) 194500
I1=1 194600
WRITE(IF,90004)I1,I1,ZERO,VB(NB(I1)+1,1,1,1) 194700
C BL MODELS 194800
WRITE(IF,90005)YUCL 194900
WRITE(IF,90006)I1,KRV(1),KV(1) 195000
WRITE(IF,90007)ZERO,ZERO,VB(NB(I1)+1,1,1,1),ZERO,ZERO,ZERO 195100
WRITE(IF,90009) 195200
DO 14020 L=1,8 195300
  I1=NB(L) 195400
  WRITE(IF,90005)BLB(L) 195500
DO 14010 I=1,I1 195600
  IP1=I+1 195700

```

Table 8. Continued

```

      I2=1
      IF(CRVB(I,L).EQ.ELLX) I2=2
      WRITE(IF,90006) I,CRVB(I,L),KV(I2)
      WRITE(IF,90007) VB(I,1,1,L),VB(I,2,1,L),VB(IP1,1,1,L),VB(IP1,2,1,L),
      1,VB(I,3,1,L),VB(I,3,2,L)
14010 CONTINUE
      WRITE(IF,90009)
14020 CONTINUE
      WRITE(IF,21900)
      WRITE(IF,90011)
      IF(IPRINT.EQ.0) GO TO 14030
C SET UP CARDS TO MAKE QUICK EXERCISE MODEL
      DX=.001
      A1=DX
      A2=VB(NB(1)+1,1,1,1)-DX
      A3=(A2-A1)*.01
      A2=A2+2.*DX
      I1=1
      I2=2
      WRITE(IF,90012) I1,I2,A1,A2,A3
14030 WRITE(IF,90013)
      RETURN
90001 FORMAT(80A1)
90004 FORMAT(212,6X,2F10.5)
90005 FORMAT(A4)
90006 FORMAT(I2,1X,A4,' PIEC ',A4)
90007 FORMAT(3F10.5,2(F9.5,'A'),F10.5)
90009 FORMAT(' -1')
90011 FORMAT(' ')
90012 FORMAT(I2,I3,5X,6F10.5)
90013 FORMAT(' 0')
90014 FORMAT(3F10.5,F9.5)
21700 FORMAT(' 1',78X/
      1' 1 6 NONCIRCULAR CROSS SECTION
      2'LELL 1ELLI PIEC 8LCL SID LSCP
      3'UELL 2ELLI PIEC SID TUCL USCP
      4'LBSL 3LINE PIEC 8LSL LSLCP
      5LELL',11X/
      6'LSSL 4LINE PIEC LSLCP SSL
      7'USSL 5LINE PIEC SSL USLCP
      8UELL',11X/
      9'UTSL 6LINE PIEC USLCP TUSL
      A ' 1 MAP ')
      21900 FORMAT('YUSL YUCL',66X/
      1 'YLSL YUCL',66X/
      2 'YLCCL YUCL',66X/
      3 'YMAP YUCL',66X/
      4 'YUSCP YSID',66X/
      5 'YLSCP YSID',66X/
      6 'YUSLCP YSSL',66X/
      7 'YLSLCP YSSL',66X/
      8 'ZUSCP ZUCL',66X/
      9 'ZLSCP ZLCL',66X/
      A 'ZUSLCP ZUSL',66X/
      B 'ZLSLCP ZLSL',66X/
      C 'ZSSL ZSID',66X)

```

```

196100
196200
196300
196400
196500
196600
196700
196800
196900
197000
197100
197200
197300
197400
197500
197600
197700
197800
197900
198000
198100
198200
198300
198400
198500
198600
198700
198800
198900
199000
199100
199200
199300
199400
199500
199600
199700
199800
199900
200000
200100
200200
200300
200400
200500
200600
200700
200800
200900
201000
201100
201200
201300
201400
201500
201600

```

Table 8. Concluded

```

END
C*****
BLOCKDATA
COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
4 VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
5 BLB(8)
COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
COMMON /RS/KI,MJ,KJ,MJ,MJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
1 YUCL,ZERO,FORE,AFFE,TOPS,BOTS,OPUP,OGIV,P10180,XJ,ZJ,XI,ZI,MI,
2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THJ,SIJ,A1,MJ,RJ,A4,DX
3 TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEOP1,ISTOP,JSTOP,KSTOP,
1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
COMMON /LS/TI,T
REAL K,M,KSI,KI,MI,KJ,MJ,MJ,LINE,KV,KRV
LOGICAL*1 SLOPE,IDEN,CAND,TI,T
DATA KRV/'LINE','OGIV','ELLX','LINE','LINE','CL','ZUCL','ZLCL',/
1 KV/'KVS','KV0','SLOPE/10*'F',/T/'T'/
DATA RHO,X1,X2,Z1,Z2,T1,T2,KSI,ETA/90*1.E70/
DATA X,Z, R,M,K,M,B,TH/ 80*1.E70/
DATA CURVE/10*'','BLNK','',RINF/1.E69/,
1 ERRMAX/1.E-5,DIR/10*'OPDN','OPDN','OPDN','ELLX','ELLX',/
2 YUCL/'YUCL','ZERO/0./,FORE/'FORE','AFTE/'AFTE','TOPS/'TOPS',/
3 BOTS/'BOTS','OPDN','OPDN','OGIV','OGIV','OPUP','OPUP',/
DATA XQ,ZQ,TQ/44*1.E70/
DATA V/ 90*1.E70/SL/'ZUSL','ZLSL','YSSL'/
DATA RHOSL/30*1.E70/CRVSL/30*'LINE'/
DATA VB/480*1.E70/CRVB /80*'LINE',/TYPE/'SYMM',ASYM/'ASYM'/
DATA BLB/'ZUCL','ZLCL','YSID','ZUSL','ZLSL','YSSL','ZSID','ZMAP',/
DATA XMAPAX,ZMAPAX,XSID,ZSID/40*1.E70/
DATA UPFO , UPAF , UPTO , UPBO , DNFO , DNAF , DNTO , DNBO
1 /'UPFO','UPAF','UPTO','UPBO','DNFO','DNAF','DNTO','DNBO',/
DATA JCASE/582,774,834,526,624,006,010,012,136,590,
1 626,628,632,014,138,140,142,638,832,262,
2 838,
-9*0/
DATA NCASE/ 1. 2. 3. 4. 6,
1 7. 8 /
END

```

201700
201800
201900
202000
202100
202200
202300
202400
202500
202600
202700
202800
202900
203000
203100
203200
203300
203400
203500
203600
203700
203800
203900
204000
204100
204200
204300
204400
204500
204600
204700
204800
204900
205000
205100
205200
205300
205400
205500
205600
205700
205800
205900

Table 9. Subroutine Description

<u>Subroutine Name</u>	<u>Function</u>
Main Program	Overall program control
INIT	Initializes certain program variables
INPT	Reads and prints input data
CASE	Determines the case number ICASE(I) of the input option for each segment (see Table 2)
SEGE	Determines the equation of each segment and computes certain intersection information
INTR	Determines intersection points of successive segments for which only segment equations are known to that point in code
QARY	Assembles coordinate and slope data in convenient arrays for building QUICK input deck (QUICK)
PRNT	Prints the QUICK input arrays assembled by QARY
R ϕ FL	Determines the equations of rounds and fillets and computes intersections with body line segments, prints results
QUCK	Builds the QUICK input deck for the symmetrical case with asymmetrical slices
LIST	Prints the QUICK input deck (if unit IF is not the card punch) for checking by user
SLCE	Performs geometry computation for slicing planes, prints results
RFSL	Determines the equations of rounds and fillets and computes intersections with slicing plane segments, prints results
C ϕ PY	Assembles coordinate and slope data in convenient arrays for building QUICK input deck for asymmetrical case (QK3D)

Table 9. Concluded

BLCK	Computes default values of certain body line coordinates and checks validity of previously computed coordinates for asymmetrical case
QK3D	Builds the QUICK input deck for the asymmetrical case with slices
BLφCKDATA	Initializes certain program constants

Table 10. Variable Description

Variable Name	Type*	Description
B	RA	Z intercept of straight body line segment
H	RA	X coordinate of ogive center
I	IS	Intersection subscript
J	IS	Scratch integer
K	RA	Z coordinate of ogive center
L	IS	Scratch integer
M	RA	Slope of body line segment
N	IS	Scratch integer
R	RA	Radius of ogival segment
S	RS	Variable used in selection from multiple intersections, S=±1.
T	LS	= 'T'
V(I,J,L)	RA	Slice geometry array: J=1, X coordinate; J=2, Z (or Y) coordinate; J=3, slope angle (deg); L=1, upper slice; L=2, lower slice; L=3, side slice
X	RA	Input array for X coordinate of body lines
Z	RA	Input array for Z (or Y) coordinate of body lines
Ai	RS	Scratch variables i=1,2,...
BI	RS	Z intercept of Ith segment
BJ	RS	Z intercept of I-1st segment
CL	RA	Body line names used in symmetrical case to assemble QUICK input deck

Table 10. Continued

DX	RS	Round-off compensation variable
HI	RS	X coordinate of ogive center for Ith segment
HJ	RS	X coordinate of ogive center for I-1st segment
IF	IS	Unit number (data set reference number) of QUICK file written by QUICK and QK3D
IP	IS	Unit number of facility card punch
IR	IS	Unit number for input data
IW	IS	Unit number for facility printer
Ii	IS	Scratch integers, $i=1,2,\dots$
KI	RS	Z coordinate of ogive center for Ith segment
KJ	RS	Z coordinate of ogive center for I-1st segment
KV	RA	Segment type (line or ogive)
LB	IS	Overall $D\phi$ variable for asymmetrical case, LB=1, upper centerline; LB=2, lower centerline; LB=3, side (see Fig. 7)
MI	RS	Slope of Ith segment (straight line)
MJ	RS	Slope of I-1st segment (straight line)
NB(LB)	IA	Number of segments in the body line segments in asymmetrical case
NP	IA	NS+1
NS(L)	IA	Number of segments in the three slicing planes, L defined as for V above
RI	RS	Radius of ogive for Ith segment
RJ	RS	Radius of ogive for I-1st segment

Table 10. Continued

SL	RA	Body line names for slicing planes
TH	RA	Input array for local slopes of body lines at end of segment
TI	LS	=SL ϕ PE(I)
TQ(I,J)	RA	Local segment slope (degrees): J=1, at beginning of Ith segment; J=2 at end of Ith segment; input to subroutine QUICK
Ti	RA	Slope (degrees) of fillet or round: i=1, beginning of fillet; i=2, end of fillet
VB(I,J,N,L)	RA	Coordinates and slopes of the eight body lines of the asymmetrical case: J=1, X coordinate J=2, Z coordinate J=3, slope (deg); N=1, beginning of Ith segment, N=2 end of Ith segment (N=2 used only with J=3) L=1 Upper centerline (ZUCL) L=2 Lower centerline (ZLCL) L=3 Side (YSID) L=4 Upper slicing plane (ZUSL) L=5 Lower slicing plane (ZLSL) L=6 Side slicing plane (YSSL) L=7 Side (ZSID) L=8 Map axis (ZMAPAX) (see Fig. 8)
XI	RS	X coordinate of end of Ith segment
XJ	RS	X coordinate of end of I-1st segment
XQ	RA	X coordinate of beginning of Ith segment (I=1,NSEG) and end of last segment (I=NSEGP1), input to QUICK
Xi	RA	X coordinate at intersections of fillet or round with body lines, i as above in Ti
ZI	RS	Z coordinate at end of Ith segment
ZJ	RS	Z coordinate at end of I-1st segment

Table 10. Continued

ZQ	RA	Z coordinate of beginning of Ith segment (I=1,NSEG) and end of last segment (I=NSEGP1), input to QUICK
Zi	RA	Z coordinate at intersections of fillet or round with body lines, i as above in Ti
BIJ	RS	Z intercept
BLB(L)	RA	Body line names for QK3D, L as above for VB
DIR	RA	Selection indicator variable for multiple root intersections
ETA	RA	Z coordinate of round or fillet center at end of Ith segment
IP1	IS	Scratch integer, =I+1
ISL	IS	Slice indicator: ISL=0 no slices ISL=1 slices present
KRV	RA	Segment type, line or ogive
KSI	RA	X coordinate of round or fillet center at end of Ith segment
LM1	IS	scratch integer, =L-1
MIJ	RS	Tangent of straight line slope angle
NM1	IS	Scratch integer, =N-1
RH ϕ	RA	Input variable for radius of round or fillet at end of Ith segment
THI	RS	Slope (deg) at end of Ith segment
THJ	RS	Slope (deg) at end of I-1st segment
AFTE	RA	= 'AFTE', possible DIR value (aft solution)
ASYM	RA	= 'ASYM', possible TYPE value
BLNK	RS	= 'bbbb' (four blank characters)
B ϕ TS	RS	= 'B ϕ TS', possible DIR value (bottom solution)

Table 10. Continued

CARD(80)	LA	Card image array
CRVB(I,L)	RA	Curve type indicator for QUICK, CRVB ='LINE' or 'ELLX'
ELLX	RS	='ELLX'
FΦRE	RS	='FΦRE', possible DIR value (fore solution)
IDEN(80)	LA	Input variable for identification data
LINE	RS	='LINE', possible CURVE value
NPTS	IS	Number of stations at which QUICK will print out geometry data if IPRINT≠0
NSEG	IS	Number of segments for symmetrical case
ΦGIV	RS	='ΦGIV', possible CURVE value
ΦPDN	RS	='ΦPDN', possible DIR value (open down)
ΦPUP	RS	='ΦPUP', possible DIR value, (open up)
RINF	RS	Real infinity, = 1.E69
THIJ	RS	Straight-line slope (deg)
TLSSL } TSSL } TUSL }	RA	Input arrays for local slopes of the lower slice, side slice, and upper slice segments, respectively
XLSSL } XSSL } XUSL }	RA	Input arrays of X coordinate of end of lower, side, and upper slicing plane segments, respectively
ZLSSL } YSSL } ZUSL }	RA	Input arrays of Z coordinate of end of lower, side, and upper slicing plane segments, respectively
TΦPS	RS	='TΦPS', possible DIR value (top solution)
TYPE	RS	Type body: TYPE='SYMM', symmetrical ='ASYM', assymetrical
XSID	RA	X coordinate of side body line (Z)

Table 10. Continued

YUCL	RS	'YUCL'
ZER ϕ	RS	Real zero = 0.0
ZSID	RA	Z coordinate of side body line
CRVSL(I,L)	RA	Curve type ('LINE' or 'ELLX') for slicing planes, L as in V above
CURVE	RA	Curve type for Ith segment, ='LINE' or ' ϕ GIV'
ICASE	IA	Integer for identifying input option (see Table 2)
INPUT		Namelist name for input data
IST ϕ P	IS	Error indicator to halt execution when set different from zero
JCASE	IA	Integer array used to determine ICASE (see Table 2)
JST ϕ P	IS	Error indicator similar to IST ϕ P
KST ϕ P	IS	Error indicator similar to IST ϕ P
NCASE	IA	ICASE values which input options may not follow ICASE = 5 or 9
RH ϕ LS	RA	Input array for lower slice fillet/round radii
RH ϕ SL(I,L)	RA	Array of fillet/round radii, L as in V above
RH ϕ SS	RA	Input array for side slice fillet/round radii
RH ϕ US	RA	Input array for upper slice fillet/round radii
SL ϕ PE	LA	Variable to indicate Ith segment is to be tangent to end of I-1st segment
ERRMAX	RS	Computational error limit, = 5.E-5

Table 10. Concluded

IPRINT	IS	Print indicator: IPRINT $\neq$ 0 causes KWIKN ϕ SE to generate cards to make QUICK exercise the completed math model
NSEGP1	IS	=NSEG+1
PI ϕ 180	RS	$\pi/180$
XMAPAX	RA	X coordinate of map axis on input
ZMAPAX	RA	Z coordinate of map axis on input

*R=real, I=integer, L=logical, A=array, S=scalar