NRL Report 7025

Cross-Sectional Plots of Plane Intersections

An Adaptation of the APT System

KINGSLEY P. THOMPSON

Information Systems Branch
Mathematical and Information Sciences Division

January 27, 1970

NAVAL RESEARCH LABORATORY
Washington, D.C.

UNCLASSIFIED

CONTENTS

ABSTRACT

The APT (Automatically Programmed Tooling) system provides for the software description of geometric forms, the delineation of a tool path, and, for output, discrete positioning information on punched paper tape for use by a numerically controlled machine. The CROSEC (Mod 1) program, described in this report, provides a means of extending the use of the canonical forms of the plane surfaces defined by the programmer in the part program by providing a plotting capability in which the lines in intersection, within specified limits, between a cross-sectional plane and all other defined planes are shown. A visualization of the initial plane framework on which the cutting is to be performed is thereby provided. The cross sectional plane and its dimensional limits are controlled by one plane definition and two point definitions. The plot is supplemented by printer output that aids in the interpretation of the plot. The program is written in the framework of the CDC 3800 APT 2.1 configuration. No additional program overlays or segments are necessary. This report contains a discussion of the method used, subroutine descriptions, listings and flowcharts, implementation aids, and a sample run.

PROBLEM STATUS

This is an interim report on a continuing problem.

AUTHORIZATION

NRL Problem 23Z0001

Manuscript submitted November 17, 1969.

CROSS-SECTIONAL PLOTS OF PLANE INTERSECTIONS

An Adaptation of the APT System

INTRODUCTION

This report contains a complete description of a plotting adaptation of the APT system in use at the Naval Research Laboratory (APT 2.1 on the CDC 3800). In the APT language, a plane can be defined in seven different ways; but after being processed they are all stored in a standard "canonical form." This program, called CROSEC (Mod 1), utilizes such plane canonical forms (of an APT Part Program) to obtain a plot of the intersection of one of the planes, designated as the cross-sectional plane, and all of the other planes that have been defined, processed, and stored at the end of APT Section 1. (See Ref. 1 for a detailed description of the APT system.)

The purpose of the program, primarily, is as an aid, an extra tool for the programmer in debugging his program. The defined surfaces as stored in their canonical forms provide a convenient starting point for geometric considerations. It is assumed that syntactic errors have already been discovered and corrected and that in using CROSEC the programmer wishes to verify that the surfaces he has defined do indeed describe the piece he wishes to have worked on by the tool. The hope is that verification can be accomplished easily if he can get a look at any cross section of his choosing through the conglomerate of the starting surfaces. He realizes that the plot might require some interpretation because defined surfaces intersecting together do not fully describe the finished piece. However, he accepts this limitation and looks upon the output as a working drawing, a picture of the output of Section 1. By means of this drawing and the accompanying identifying information from the printer, he should be able to make some significant debugging progress. Perhaps he will discover a section of surface that is defined improperly or over defined, or a combination of surfaces that could be redefined in a simpler manner. Also, he may discover a portion of surface that he has not yet defined and other events of this nature.

The report discusses plane equations, outlines the method of obtaining a coordinate system in the cross-sectional plane, describes the plot, and points out the limitations of the program. Also included are flowcharts, program descriptions, a complete listing, and details of implementation with APT 2.1. An example is introduced early and followed through the complete process in full detail.

DISCUSSION

The Plane

The *canonical* form for the plane, as defined in the APT system, is given by

$$AX + BY + CZ = D, \tag{1}$$

where*

*As quoted from pp. 80 and A-1 of Ref. 1.

A = X-Component of unit normal vector to plane
B = Y-Component of unit normal vector to plane
C = Z-Component of unit normal vector to plane
D = Normal distance from plane to origin.

Consider the *normal* form of the plane equation, namely

$$XCOS\alpha_x + YCOS\alpha_y + ZCOS\alpha_z = L, \tag{2}$$

where the coefficients for the X, Y, and Z coordinates in Eq. (2) are the direction cosines of the positive normal vector N from the origin to the plane. The direction cosines of the unit normal vector are identical with the direction cosines of N. The length, or absolute value, of this normal is L. Therefore, equating similar terms from Eqs. (1) and (2) yields

$$A = COS\alpha_x$$

$$B = COS\alpha_y$$

$$C = COS\alpha_z$$

$$D = L = |N| .$$

Thus, for example, A = 0, B = 1, C = 0, D = 1 is the plane passing through the point (0, 1, 0), parallel to the XZ coordinate plane, with direction angles of 90, 0, and 90 degrees, respectively, to the three axes.

Let us now take, for a more detailed example, the plane that passes through the points (1,0,0), (0,1,0), and (0,0,1) a unit distance out along each axis. To fit this approach to the definition of a plane, consider the *intercept* form for a plane equation,*

$$\frac{X}{X_1} + \frac{Y}{Y_2} + \frac{Z}{Z_3} = 1, \tag{3}$$

where X_1, Y_2, and Z_3 are the intercepts, i.e. the point (X_1, 0, 0) is the intersection of the X axis with the plane. Similarly with (0, Y_2, 0) and (0, 0, Z_3).

Using now the three unit axis points, already defined, in this intercept form, Eq. (3) leads to the simple and interesting equation

$$X + Y + Z = 1. \tag{4}$$

The correctness of Eq. (4) as truly representing the plane that passes through the three points is easily determined by setting any two of the variables equal to zero, and the remaining variable will be equal to 1. Equation (4) is illustrative of another form of an equation used to describe a plane, the *general* form, where the coefficients of the X, Y, and Z terms are considered to be *direction numbers* of the positive normal to the plane.

To go from the general form to the normal form, it is necessary to compute the direction cosines by dividing each of the coordinate coefficients in turn by the square root of the sum of the squares of all three coefficients. The length of the normal is obtained in a similar fashion by dividing the constant term by the same square-root quantity. Symbolically, if the general form is

$$PX + QY + RZ = S, \tag{5}$$

*All forms of equations for a plane can be found in Section 3, p. 2-1 of Ref. 2.

then

$$A = \text{COS}\alpha_x = \frac{P}{U}$$

$$B = \text{COS}\alpha_y = \frac{Q}{U}$$

$$C = \text{COS}\alpha_z = \frac{R}{U}$$

$$D = L = \frac{S}{U},$$

where

$$U = \sqrt{P*P + Q*Q + R*R.}$$

The normal form for the illustrative plane is therefore

$$\frac{X}{\sqrt{3}} + \frac{Y}{\sqrt{3}} + \frac{Z}{\sqrt{3}} = \frac{1}{\sqrt{3}}. \tag{6}$$

It is easily verified that the sum of the squares of the direction cosines is 1.

It is important in our development to know the coordinates of the point represented by the intersection of the normal N and the plane to which it is perpendicular. These coordinates are obtained by multiplying each of the direction cosines by the length of the normal. Symbolically, (D*A, D*B, D*C). For the illustrative plane the result is (1/3, 1/3, 1/3).

One final consideration regarding these plane equations—our illustrative case has a convenient set of intercept points; however, it is possible to determine the intercepts from the normal form. They are

$$(X_1, Y_1, Z_1) = \left(\frac{D}{A}, 0, 0\right)$$

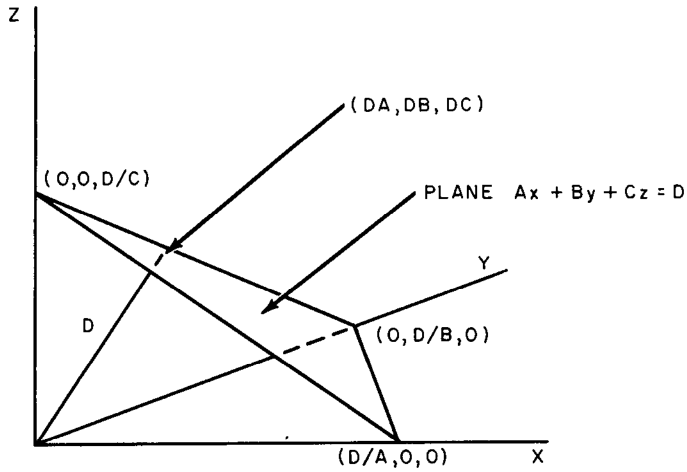
$$(X_2, Y_2, Z_2) = \left(0, \frac{D}{B}, 0\right)$$

$$(X_3, Y_3, Z_3) = \left(0, 0, \frac{D}{C}\right).$$

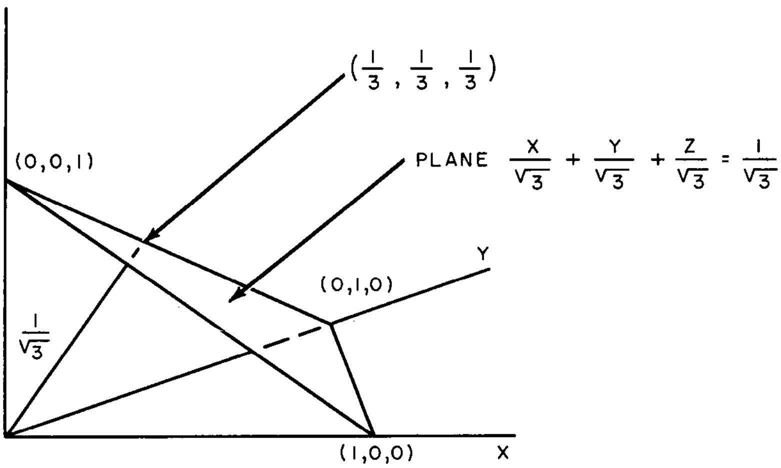
Figure 1 summarizes this initial development using the symbols for the APT canonical form of Eq. (1) in part a of the figure and the actual values of our illustrative plane in part b.

The Cross-Sectional Plane and Its Coordinate System

The plot of Section 1 output uses a coordinate system in the cross-sectional plane, particularized as the HOPE plane. Let us call the coordinate system in the HOPE plane the prime system (i.e., X′, Y′, Z′), in contrast to the original system established by the part programmer known as the X, Y, Z system. The origin of the prime system is the intersection of the normal with the plane, and the Z′ axis is the extension of the normal.



(a) Normal form of plane equation



(b) Equiangular plane in normal form

Fig. 1 - A general and a particular plane

The X′ axis is selected in one of three ways.

1. The "general solution" is the case where the HOPE plane intersects the X axis, and the positive X′ axis is the line passing from the prime origin through the intercept point and lies in the HOPE plane. In terms of the *normal* form the prime origin is (D*A, D*B, D*C) and the intercept point is (D/A, 0, 0) with direction cosines

$$\left[\frac{\left(\frac{D}{A} - D*A\right)}{U}, \frac{-D*B}{U}, \frac{-D*C}{U}\right],$$

where

$$U = \sqrt{\left(\frac{D}{A} - D*A\right)^2 + (D*B)^2 + (D*C)^2}.$$

For our illustrative plane X + Y + Z = 1, the X′ axis for the general solution would be positively directed from the origin with coordinates (1/3, 1/3, 1/3) to the X intercept with coordinates (1, 0, 0). The computation gives the X′ axis, in this case, direction cosines of

$$(\sqrt{2/3},\ -\sqrt{1/6},\ -\sqrt{1/6})$$

2. If the HOPE plane does not intersect the X axis and is parallel to it, then the X′ axis is that line lying in the HOPE plane, parallel to the X axis, with direction cosines (1, 0, 0) commencing at the prime origin.

3. If the HOPE plane is perpendicular to the X axis and parallel to the YZ plane such that the normal is the X axis, then the positive X′ axis is that line lying in the HOPE plane which starts at the prime origin and is parallel to the Z axis with direction cosines (0, 0, 1).

The Y′ axis is defined as the cross product of the Z′ and X′ axes, duly preserving right-handed concepts.

If the direction cosines of the X′ axis are T_{11}, T_{21}, T_{31}, and those of the Y′ axis T_{12}, T_{22}, T_{32} and those of the Z′ axis T_{13}, T_{23}, T_{33}, then

$$T_{12} = T_{23}*T_{31} - T_{21}*T_{33} \tag{7}$$

$$T_{22} = T_{11}*T_{33} - T_{13}*T_{31} \tag{8}$$

$$T_{32} = T_{13}*T_{21} - T_{11}*T_{23}. \tag{9}$$

Thus, for our illustrative plane, the direction cosines of Z′ are $(1/\sqrt{3},\ 1/\sqrt{3},\ 1/\sqrt{3})$ and of X′ are $(\sqrt{2/3},\ -\sqrt{1/6},\ -\sqrt{1/6})$. Using these equations, we find that the directions cosines for Y′ are $(0,\ \sqrt{1/2},\ -\sqrt{1/2})$.

The equations required to convert any point in space from the original coordinates to the prime coordinates are

$$X' = T_{11}\,(X - X_0) + T_{21}\,(Y - Y_0) + T_{31}\,(Z - Z_0),$$

$$Y' = T_{12}\,(X - X_0) + T_{22}\,(Y - Y_0) + T_{32}\,(Z - Z_0),$$

and

$$Z' = T_{13}\,(X - X_0) + T_{23}\,(Y - Y_0) + T_{33}\,(Z - Z_0),$$

where (X_0, Y_0, Z_0) is the prime origin defined in terms of X, Y, Z. Expanding yields

$$X' = T_{11}*X + T_{21}*Y + T_{31}*Z + C_1, \tag{10}$$

$$Y' = T_{12}*X + T_{22}*Y + T_{32}*Z + C_2, \tag{11}$$

and

$$Z' = T_{13}*X + T_{23}*Y + T_{33}*Z + C_3, \tag{12}$$

where

$$C_1 = -(T_{11}*X_0 + T_{21}*Y_0 + T_{31}*Z_0),$$

$$C_2 = -(T_{12}*X_0 + T_{22}*Y_0 + T_{32}*Z_0),$$

$$C_3 = -(T_{13}*X_0 + T_{23}*Y_0 + T_{33}*Z_0).$$

For the illustrative case where $(X_0, Y_0, Z_0) = (D*A, D*B, D*C) = (1/3, 1/3, 1/3)$, the matrix corresponding to the coefficients of Eqs. (10), (11), and (12), is

$$\begin{bmatrix} \sqrt{\frac{2}{3}} & -\sqrt{\frac{1}{6}} & -\sqrt{\frac{1}{6}} & 0 \\ 0 & \sqrt{\frac{1}{2}} & -\sqrt{\frac{1}{2}} & 0 \\ \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} & -\sqrt{\frac{1}{3}} \end{bmatrix}.$$

Figure 2 illustrates the definition of the prime axes in the HOPE plane. In this instance the HOPE plane is our illustrative plane X + Y + Z = 1.

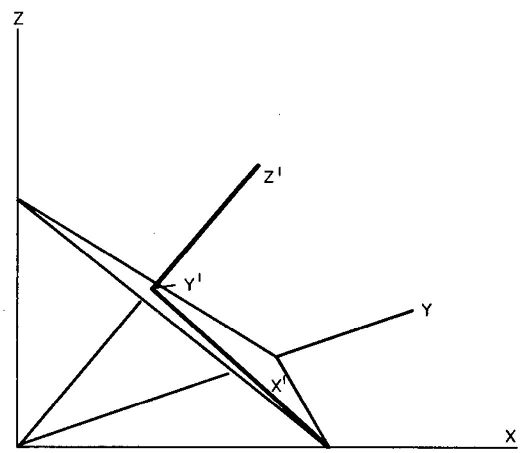


Fig. 2 - Prime axes in the equiangular HOPE plane

Points and Lines in the Cross-Sectional Plane

So far we have established an understanding of the various forms of the equations that describe a plane and have described an algorithm for defining a translation-rotation matrix for converting points from the part programmer's coordinate system to a coordinate system in the HOPE or cross-sectional plane. We can now ask, How is it determined whether any APT-defined point in general lies in the HOPE plane? Simply by substituting the X, Y, and Z values of the point into the equation for the plane to determine if an equality exists; e.g., Does the point (1/3, 1/3, 1/3) lie in the plane X + Y + Z = 1? Obviously yes, since 1/3 + 1/3 + 1/3 = 1.

If the equations for the three planes are solved simultaneously, a point that lies in all three planes is the result. If one of these planes is the HOPE plane, then there is no doubt that the point is in the cross section.

Consider now a situation in which the HOPE plane and a plane A are consecutively solved with two other planes B and C (defined such that no two planes of HOPE, A, B and HOPE, A, C are parallel), resulting in two points P_1 and P_2. Both of these points are simultaneously in both the HOPE plane and in plane A; in fact, the line segment joining P_1 and P_2 is a portion of the line of intersection between HOPE and A. Such a procedure if carried out with all the defined planes, will result in an entire network of lines of inter section. Each point as obtained is put through the matrix to obtain its definition in terms of the HOPE coordinate system to enable it to be plotted. If a minimum and a maximum value for each coordinate is specified, then many superfluous points of intersection can be eliminated.

Let us now consider what happens when the equiangular plane $X + Y + Z = 1$ intersects some planes defined in a part program, an actual situation. (Refer to sample run with "PARTNO TESTING" on p. 63.) Figure 3 depicts a simple part in three views. It is to be noted that there are eight defined planes. Figure 4 shows the cross-section network of intersecting lines obtained with the plane $X + Y + Z = 1$ as the HOPE plane intersecting "PARTNO TESTING." Using only the information provided in Figs. 3 and 4, can you distinguish between the proper intersection outline and those lines that are extraneous? It is an intersecting exercise, well worth spending a few minutes on.

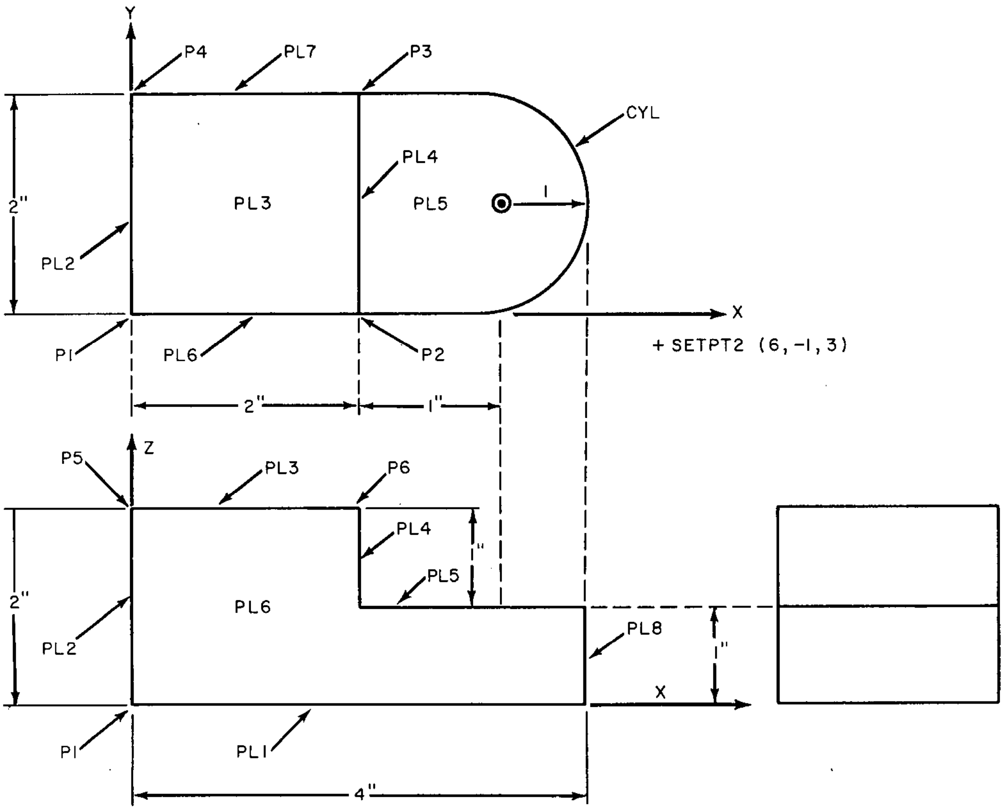


Fig. 3 - "PARTNO TESTING" with eight defined planes

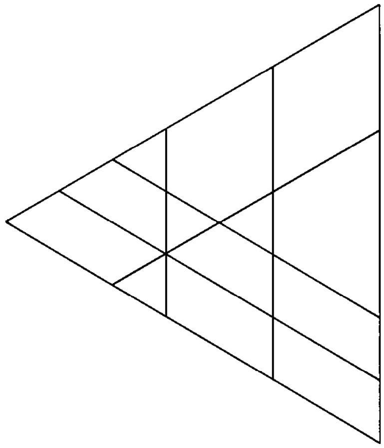


Fig. 4 - Cross section obtained from PARTNO TESTING and equiangular plane

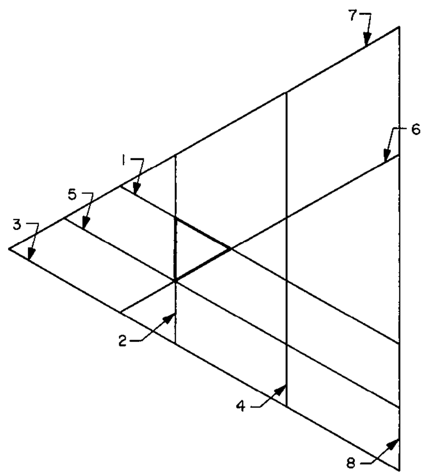


Fig. 5 - Lines of intersection identified by plane number

The solution to this question will now be presented.

The accompanying printout identifies points of intersection from which the lines of intersection were drawn. The coefficients of these points are given in the X, Y, Z system, in the X′, Y′, Z′ system, and in plotter coordinates along with the plane numbers associated with each point. There are 19 such points in Fig. 4. These point data constitute background information, available when needed. However, rather than clutter up the plot with point symbols and identifying numbers, only the lines of intersection have been drawn using a minimum point and a maximum point. From these two points the slope and the prime-system intercept have been computed. This information is also printed along with the plane name associated with the line. The lines have been named in Fig. 5. There are eight planes identified in the sketch of the part in Fig. 3 and there are eight lines in the intersection plot.

Verification of the correctness of the numbering of these lines of intersection can be achieved by noting that parallel *lines* in the plot correspond to parallel *planes* on the part. Sure enough, plane 1 representing the bottom of the piece is parallel to plane 3, the top, and is also parallel to plane 5 which represents the step surface. In Fig. 5 the lines labeled 1, 5, and 3 are parallel lines.

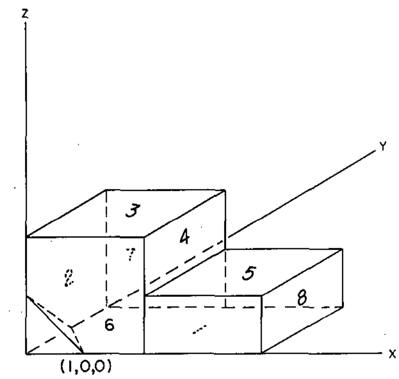


Fig. 6 - Perspective view of PARTNO TESTING (without the cylinder) with eight planes and the intersecting equiangular HOPE plane

In a similar fashion the lines of intersection for planes 2, 4, and 8 confirm a parallelism between these planes, all three of them being vertical sides in the part shown in Fig. 3. Finally, plane 6 (the front) is parallel to plane 7 (back), and the lines with these numbers are parallel in Fig. 5.

Figure 6, which shows a perspective view of the planes of the part and the equiangular HOPE plane, demonstrates the fact that the lines of intersection on the surface of the piece lie in planes 6 (front), 1 (bottom), and 2 (left side). The triangle formed by the lines of intersection of these three planes is accented in Fig. 5 and is the correct answer.

It is an interesting consequence of the situation that since plane X + Y + Z = 1 is equiangular to each of the three coordinate planes, and that since all eight of the defined planes in the simple part are either identical with or parallel to a coordinate plane, that HOPE is also equiangular to these eight planes, namely 57.4 degrees.

There is one other interesting observation to make about Fig. 5. Because of the equiangular characteristic of the cross-sectional plane, a reverse type of statement can be made. Any equilateral triangle in this figure corresponds to three mutually perpendicular planes. Besides the set already discussed {6, 1, 2}, eleven others can be identified. These are {1, 2, 7}, {1, 4, 6}, {1, 6, 8}, {2, 3, 6}, {3, 7, 8}, {1, 7, 8}, {5, 7, 8}, {2, 3, 7}, {3, 4, 7}, {3, 6, 8} and {3, 7, 8}. The last one, which is the outermost triangle, corresponds to the planes which are perpendicular to the maximum dimensions X = 4, Y = 2, and Z = 2, respectively.

There is a lot of information to be gleaned from the cross-sectional plot once it is understood and assimilated.

When the restrictions of XMIN, YMIN, ZMIN = (0, 0, 0) and XMAX, YMAX, ZMAX = (4, 2, 2) are placed on the cross-section points, then the network is reduced to that shown in Fig. 7, the true triangle of intersection with the part.

CROSEC Limitations

CROSEC in its present form (Mod 1) has some definite limitations. Some of the initially obvious ones are as follows:

1. There is no identification of defined points that might lie in the HOPE plane.

2. The plot does not contain any intersections of the HOPE plane with quadrix surfaces.

3. No tool motion information is present.

4. The plot resulting from the lines of intersection of defined planes with HOPE can be initially confusing, since some of the lines might have no immediate relation to the finished part, and the prime coordinate system might not be simply oriented with respect to the X, Y, Z system. However, a consideration of the original part program along with the printout can soon make the plot understandable.

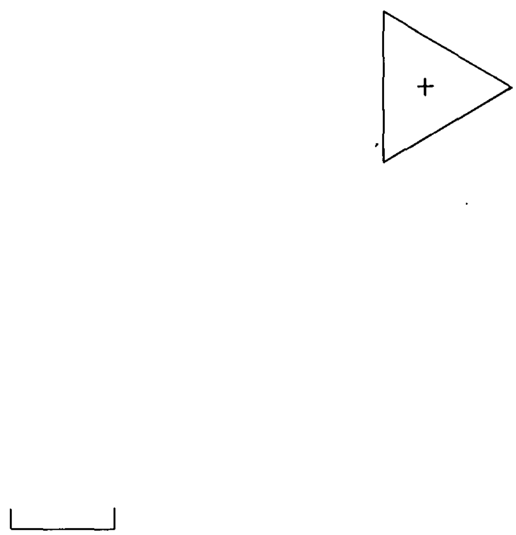


Fig. 7 - CROSEC (Mod 1) output plot for sample run with PARTNO TESTING

CONCLUSION

CROSEC (Mod 1) is a first step. In its present form, it can be helpful in a limited fashion. The intersection with curved surfaces is missing, as well as the path of the cutting tool in the cross-sectional plane. It is planned to incorporate these features in further work on this project.

ACKNOWLEDGMENT

The author expresses his appreciation to the Engineering Services Division for their cooperation in supplying information on the APT system in operation. In particular Lloyd Murphy, Nicholas Maddage, Donald Woods, and Jay Williams, Numerical Control Programming personnel of that Division, were particularly helpful. Also helpful, in more computer-related aspects, were Gary Flenner and Anna Byrd Mays of the NRL Research Computation Center, who worked on the initial implementation of APT at NRL.

REFERENCES

1. "APT Reference Manual for CDC 3400 and 3600 Computer Systems," Publication 60132800, Control Data Corporation, Palo Alto, Calif., Aug. 1965

2. Korn, G.A., and Korn, T.M., "Mathematical Handbook for Scientists and Engineers," New York:McGraw-Hill, 1961

Appendix A

SUBROUTINE DESCRIPTIONS WITH FLOWCHARTS AND LISTINGS

SUBROUTINE CROSEC

The CROSEC subroutine (Fig. A1) controls the Section 1 plotting of the intersections of defined planes with the HOPE cross-sectional plane. For convenience it incorporates all of the common area of Section 0 and Section 1 into itself without attempting to discard portions it does not need. The APT system subroutines called on are CANGET, SIMEQ and STDUNPK. The in-house library routines called include PLOTS, SYMBOL, PLOT, STOPPLOT, SQRTF, ACOSF, QNSINGL, THEND, STH, ENC, Q1Q10100, Q8QSTOPS, and Q8QDICT. Additional subroutines called by CROSEC and considered a part of the Mod 1 package are TESTHOPE, DELINE, and MM. There is also a function, ISITOK. The flowchart of Fig. A2 shows the relationship between these subroutines.

The CROSEC subroutine assumes that the defined symbol table (D.S.T.) is stored in the JTABL array of numbered common 2 between ITAB11 and ITAB12. It further assumes that the D.S.T. entries consist of pairs of words, the first of which is an eight-symbol Hollerith name, left justified, and, second, an APT "standard word" which includes an integer pointer giving the relative address in JTABL (extended beyond D.S.T.) where the canonical form is stored. These assumptions are standard procedure for the CDC APT. (See the Section 1 description starting on p. 01-1 of Ref. A1.) Accordingly, after executing a top of form, the D.S.T. is searched for the name HOPE, and if this name is not found the subroutine is exited and there is no plot obtained. If HOPE is found, its pointer is stored in KANSURF prior to calling CANGET which fetches and stores the canonical form in the DEFSTO array of the SECT1LOG. After identifying A, B, C, and D from the HOPE canonical form, the TESTHOPE subroutine is called.

TESTHOPE will either stop the run or return with a conversion matrix stored in XMAT9. This matrix will permit the conversion from the X, Y, Z coordinate system to the prime system whose origin is in the HOPE plane, thereby facilitating a two-dimensional plot.

Another top of form is executed.

Since the cosine of the angle between two planes is equal to the sum of the product of their corresponding direction cosines, it is a logical next step to take advantage of this fact and compute the angle between each defined plane and the HOPE plane.

Special care is needed in picking up canonical pointers from the D.S.T. prior to making the cosine computation. Undefined words or incorrectly defined words must be avoided as well as the synonym register which appears at the head of the D.S.T. The standard word must be unpacked by calling Subroutine STDUNPK in order to determine if BYTA contains a 4, representing a canonical form. After the canonical form has been recovered by a call to CANGET, it is necessary to examine the four most right-hand bits of the first word in the set. A 3 in this position identifies the canonical of a plane as opposed to other possible canonical forms such as points, cylinders, etc. In this manner the direction cosines are obtained, and the computation for the angle between the planes can then be performed.

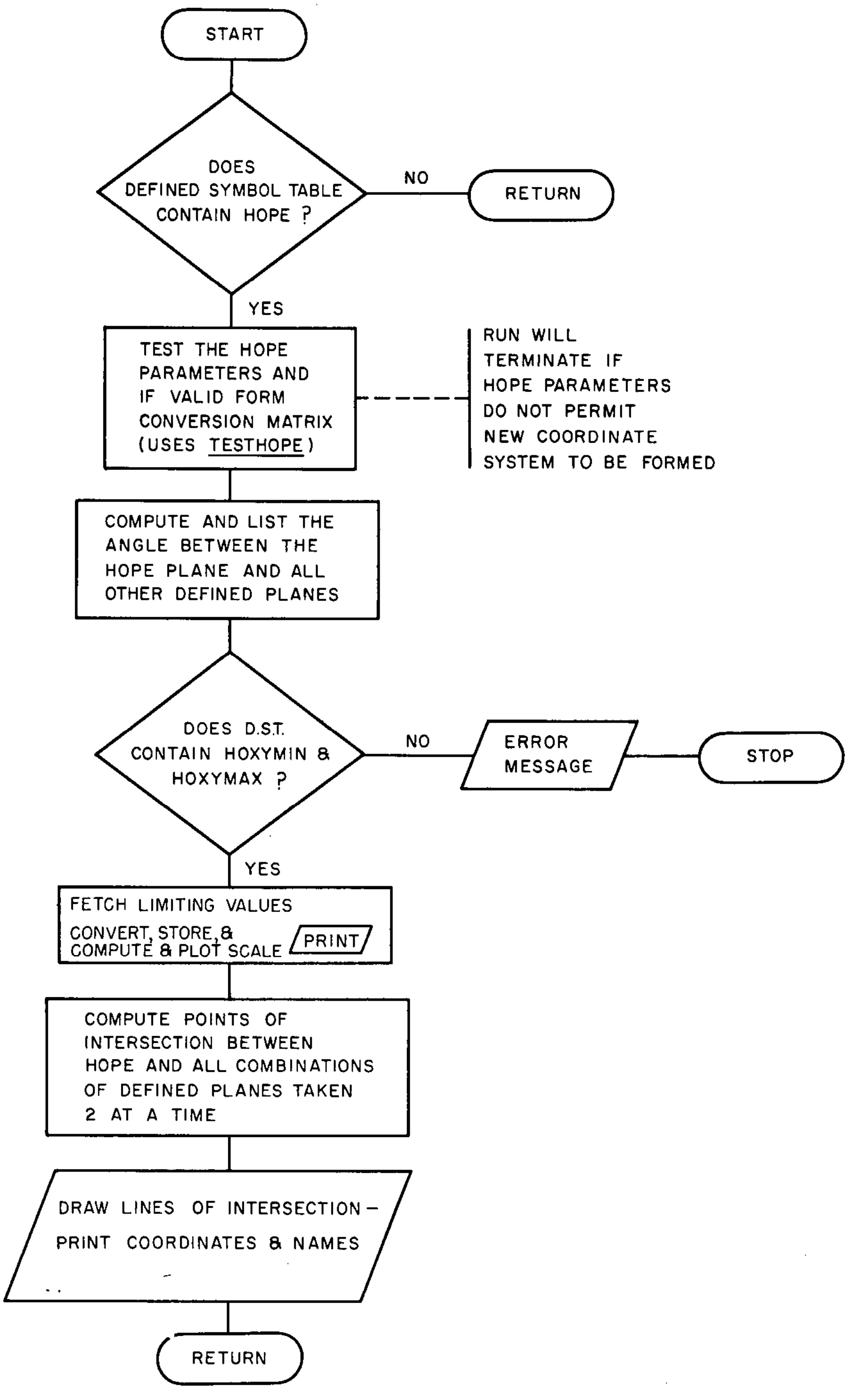
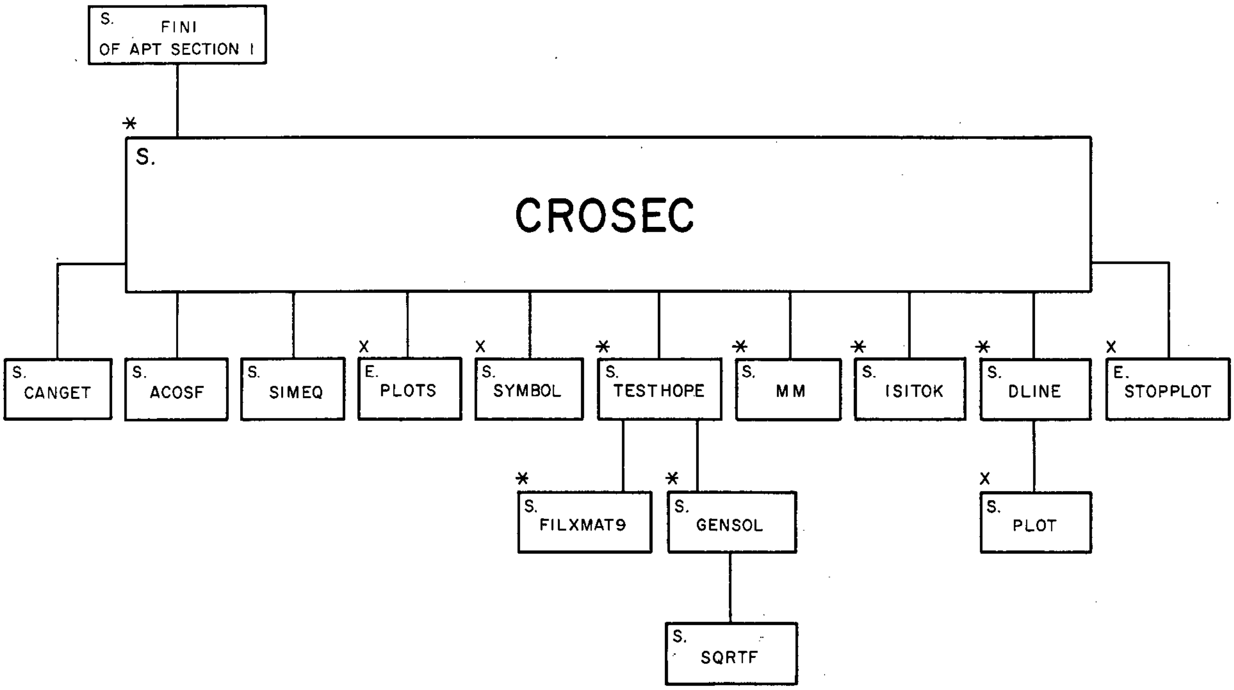


Fig. A1 - Flowchart for CROSEC subroutine



SUBPROGRAM DEPENDENCE FOR CROSEC (MOD I) USING
CDC 3800 CONFIGURATION WITH CALCOMP PLOT PACKAGE

S. = SUBROUTINE, E. = ENTRY, * = FLOW CHART IN THIS REPORT, X = PART OF CALCOMP PLOT PACKAGE

Fig. A2 - Relationship between subroutines called by CROSEC

After making special tests for perpendicular planes and parallel planes, the arc cosine function ACOSF is called and the resulting angle printed out in degrees. This angle has a range of 0 to 180 degrees. All information regarding this analysis of the angles between the planes is outputed through the line printer and is not retained, as it has no consequence to the plot.

Next, the D.S.T. is searched for the names HOXYMIN and HOXYMAX, which must be stored as adjacent pairs in order to be discovered. In the absence of these names, an error message is printed and the run is terminated. The message is "HOXYMIN AND HOXYMAX NOT FOUND." If these names are present, the corresponding values are fetched via CANGET and stored in correspondingly descriptive name locations, such as RAW X MIN.

These raw limiting values are passed through the matrix by calling subroutine MM, thus obtaining their corresponding values in the prime system, (XPMIN, YPMIN, ZPMIN) and (XPMAX, YPMAX, ZPMAX). A printout is now initiated that lists the raw limiting values.

The magnitudes of the minimum and maximum vectors are computed thusly,

$$\text{MINVECTOR} = \sqrt{(\text{XPMIN})^2 + (\text{YPMIN})^2 + (\text{ZPMIN})^2}$$

$$\text{MAXVECTOR} = \sqrt{(\text{XPMAX})^2 + (\text{YPMAX})^2 + (\text{ZPMAX})^2}.$$

The CALCOMP plot is restricted to a 10 × 10 in. space. When we allow for both negative and positive values of X′ and Y′ it follows that 2 times the largest vector must fit into 10 linear inches. The center of the plot is the origin of the prime coordinate system. To accomplish this, the largest of these two magnitudes is set equal to REACH and then divided into 5.0 to obtain the CALCOMP plot factor which is called SCALE. SCALE is therefore the number of inches of plot length that is equivalent to one inch of part length. SCLFAC is the reciprocal of SCALE and is the scale factor needed to obtain a 1:1 Gerber plot. (The details of obtaining a paper tape output for use on the Gerber plotter (Model 875) are discussed in Ref. A2.) Output messages regarding SCALE and SCLFAC are printed, followed by another top of form.

The plot string is initiated.

A + symbol is plotted at (5.0, 5.0), the plot coordinates for the HOPE prime system origin, and a scale mark is placed at the lower left-hand corner of the plot. A somewhat involved double loop is now entered at the heart of which is a call to SIMEQ which solves three plane equations simultaneously, resulting in the output of a common point of intersection (two planes intersect in a line, three planes intersect in a point).

Since the A, B, C, D of the HOPE plane are constant throughout the process, they are stored once and for all in positions 108 through 111 of the DEFTAB array (used by SIMEQ), before the entrance into the looping procedure.

Also initialized at this time are

1. IONCE, a flag which controls the call to subroutine DLINE,

2. IPTNO, the point number counter,

3. IP, the counter for coordinate pairs stored in the KR array. The KR array stores only the points related to the current line about to be drawn, and

4. IS, the counter for coordinate pairs stored in the KS array. (The KS array stores *all* points created by CROSEC.)

The outermost of the two loops commences at statement 1001.

DLINE is called only if IONCE is zero.

The canonical form for a plane is fetched under the same restraints as already described above. If a plane is indicated, its name is saved as NAME1 and its A, B, C, D are stored in DEFTAB 104 through 107. The inner loop commences at statement 1004. By a similar process, the values for the third plane are stored in DEFTAB 100 through 103, and the creation of NAME2 takes place. Then comes the call for SIMEQ followed by the test on JSUBER for proper execution, thereby allowing for the situation of two of the planes being identical.

If a bad situation resulting in no point has occurred, a transfer is made to the end of the inner loop. If a proper point had been obtained, its X, Y, Z coordinates are picked up from DEFTAB 112 through 114.

These numbers in turn are used to obtain the corresponding prime values and plotting values. The prime values are obtained by use of the conversion matrix via a call on subroutine MM. The plotting values are obtained by means of the formulas

$$X = XP * SCALE + 5.0$$

and

$$Y = YP * SCALE + 5.0, \tag{10}$$

where XP and YP are the prime coordinates.

The ISITOK function now works with an elaborate set of parameters to avoid duplication and to assure operation within the allowable limits. If the answer from the function is YES, then the point is valid and can be used. The rest of the YES followup stores the plotting values in the KS array and prints out the names of the planes, the point number, and the point coordinates in terms of the X, Y, Z system, the prime system, and the plotter system. After exiting from these two loops, DLINE is called once more if IONCE is zero; otherwise, the plot is terminated and CROSEC makes its return.

10/22/69

```
      SUBROUTINE CROSEC                                                 CRS
      COMMON /TIMETEST/       KSETIME1, KSETIME2, KSETIME,  KSETADR     CRS
C                                                                       CRS
C,,,     COMMON AREA FOR CDC APT 3 SECTION 0                            CRS
C                                                                       CRS
C******          ******          ******          ******          ****** CRS
      COMMON /SYSTEMZ/          SYSTEM(4),KAPTCN,   KAPTTR,   KAPTIO,   CRS
     A     KAPTID,   KFLAGS(10),K0,       K1,       K2,       K3,       CRS
     B     K4,       K5,       K6,       K7,       K8,       K9,        CRS
     C     IFILL1,   IFILL2,   IFILL3,   IFILL4,                        CRS
     D     KFLAG0,   KFLAG1,   KFLAG2,   KFLAG3,   KFLAG4,              CRS
     E     IFILL5,   IWAVEN,   IPTNLY,   NOPOST,   IFILL6,   KAUTOP,    CRS
     F     ICLPRT,   INDEXX,   IPLOTR,   IFILL7,   NOPLOT,   KDYNFG,    CRS
     G     LOCJPT,   LOCBEG,   KSECIN,   NCLREC,   LOCMAC,   KPOCKET,   CRS
     H     IFILL8,   IFILL9,   IFILL10,  IPOSTP(1),NUMPST,   IPOSTFL(18),CRS
     I     TAPETB(1),CANTAP,   CLTAPE,   POCTAP,   PLOTAP,   SRFTAP,    CRS
     J     LIBTAP,   CRDTAP,   IFILL11,  CORTAP,   TAPES1,   TAPES2,    CRS
     K     TAPES3,   TAPES4,   FORTIN,   INTAPE,   IOUTAP,   PUNTAP,    CRS
     L     LSTFLG,   LTVFLG,   KONVTCL,  KINTRUPT,                      CRS
     M     PI,       PIO2,     DGTRD,    RDTDG,    ONE,                 CRS
     N     EXTRA0(20)                                                   CRS
      EQUIVALENCE   (PROTAP,  TAPETB)                                   CRS
C******          ******          ******          ******          ****** CRS
C                                                                       CRS
C,,,     COMMON AREA FOR CDC APT 3 SECTION 1                            CRS
C                                                                       CRS
C******          ******          ******          ******          ****** CRS
      COMMON /SECT1LOG/         ITAB1,    ITAB2,    ITAB3,              CRS
     A     ITAB4,    ITAB5,    ISNAM,    ITAB11,   ITAB12,   ITAB13,    CRS
     B     JENDPTPP, JENDCAN,  JENDSTOR, JSTRTCAN, JENDSYM,             CRS
     C     JCANTEMP, JRPTAB,   JLPTAB,   MAXNST,                        CRS
     D     JINWD,    JCHAR,    IWDERR,   JBUFL,    NUPERP,   NUPUN,     CRS
     E     JSTYPE,   JVARS2,   SCHERR,   NMACV,    MACASN(25),          CRS
     F     INDXPT,   IPTP,     IXPT,     MODE,     EOCFLG,   LPNDFL,    CRS
     G     TRMFLG,   INTRUPT,  JUMPFL,   ICDERR,   DEBUG,               CRS
     H     MACMODE,  NESTFL,   NRESULT,  IPTLIM,   JEXEC,               CRS
     I     KTYPE,    MACTYP,                                            CRS
     J     IPARTERR, FINIS,    IOFLG,    MACDEL,   JSUBER,   NUMBERR,   CRS
     K     DEFSTO(85),    DEFTAB(1000),  ZSUR(30),                      CRS
     L     XMAT4(16),XMAT3(16),XMAT2(16),XMAT1(16),TMATX(16),           CRS
     M     ISTDMODE, ISTDLIT,  ISTDTBL,  ISTDINDX, ISTDTYPE, ISTDWD,    CRS
     N     JPTIND,   KPTCODE,  KPTNAME,  KPTTYPE,  KPTNUM,   KPTINDX,   CRS
     O     KPTSUB,   KOMFLG,   KOMPOP,   NOSURS,   KANFLG,   KRFSYS,    CRS
     P     KANREC,   KANCNT,   INAME,    KANSURF,  KANINDX,             CRS
     Q     JPRELEN,  NEWCARD,  JGORIT,   NUMSTID,  NUMCSEQ,             CRS
     R     IRECIX,   IRECNO,   JTLPOS,   ITITLE(9),LSRECN,              CRS
     S     NNODEFX,  NNODEFI,  NIDUM,    ISLASH,   IEQUAL,   IBLANK,    CRS
     T     IDUMMY,   N10000,   N7777,    MASKU,    MASKL,    IDIV,      CRS
     U     MACREL,   MACLOC,   MACBEGN,  MACLAST,  MACLEVEL,            CRS
     V     MACNAME(3), MACINDX(3), NMV,  JRESTOR,  MACPSH(3,25),        CRS
     W     JTEMP1,   JTEMP2,   JTEMP3,   JTEMP4,   JTEMP5,   JTEMP6,    CRS
     X     JTEMP7,   JTEMP8,   JTEMP9,   EXTRA1(20)                     CRS
      EQUIVALENCE (DEFANS(1),IDEFSTO(4),DEFSTO(4)),(LSTYPE, KTYPE),     CRS
     1     (PTNUM, KPTNUM)                                              CRS
      DIMENSION IDEFSTO(85),   DEFANS(26),   IDEFTAB(1000)   ,          CRS
     1     ILPTAB(200), IRPTAB(200), ITNTAB(200), JPROTP(100),          CRS
```

10/22/69

```
     2    IRECSV(200), MACVAR(25),  MACNOR(25),  INWORD(10), IBUUP(2),
     3    PISTO(6), IPISTO(6), IDREC(4)
      EQUIVALENCE (INWORD(14), IBUUP(16), ILPTAB(990), IRPTAB(790),
     1    ITNTAB(290), PISTO(390), IPISTO(390), IDREC(384),
     2    JPROTP(380), IRECSV(280),  MACVAR(75), MACNOR(50),
     3    IDEFTAB(1000), DEFTAB(1000))
C
      COMMON/VOCABTBL/ KOM(1100)
C
      COMMON /2/ JTABNUM,JTABL(12120)
C******           ******           ******           *******          ****
C
C
C         *        *        *        *        *        *        *
C         *        *      C O M M E N T       *        *        *
C         *        *        *        *        *        *        *
C         *        *        *        *        *        *        *
C
C     CROSEC (MOD 1) PROVIDES A MEANS OF EXTENDING THE USE OF THE PLANE
C     SURFACES(DEFINED BY THE PART PROGRAMMER IN THE PART PROGRAM) BY
C     PROVIDING A PLOTTING CAPABILITY IN WHICH THE LINES OF
C     INTERSECTION, WITHIN SPECIFIED LIMITS, BETWEEN A CROSS-SECTIONAL
C     PLANE AND ALL OTHER DEFINED PLANES ARE SHOWN.
C
C     THE CROSS SECTIONAL PLANE FOR THE PLOT AND ITS DIMENSIONAL LIMITS
C     ARE CONTROLLED BY MEANS OF ONE PLANE DEFINITION (NAMED ≠≠HOPE≠≠)
C     AND TWO POINT DEFINITIONS (NAMED ≠≠HOXYMIN≠≠ AND ≠≠HOXYMAX≠≠)
C     ADDED TO THE PART PROGRAM.
C
C     THE PLOT IS SUPPLEMENTED BY PRINTER OUTPUT THAT IDENTIFIES POINTS
C     OF INTERSECTION THAT HAVE BEEN NUMBERED ON THE PLOT.
C
C     THE PROGRAM IS WRITTEN IN THE FRAMEWORK OF THE CDC 3800 APT 2.1
C     CONFIGURATION AND NO ADDITIONAL OVERLAY OR SEGMENT MANIPULATION
C     IS NECESSARY.
C
C     CROSEC IS CALLED FROM SUBROUTINE FINI, AND THIS CALL IS THE ONLY
C     MODIFICATION TO APT 2.1 PROGRAMMING BEYOND THE ADDITION OF CROSEC
C     AND ITS FAMILY OF SUBROUTINES.
C
C
C         *        *        *        *        *        *        *
C
C     CROSEC DIMENSIONING BEGINS HERE
C
C     ≠HOPE≠ IS THE NAME GIVEN TO THE PLANE OF THE CROSS SECTION
C
C      NOTE--THE HOPE SYSTEM IS CALLED THE PRIME SYSTEM, THE TERMS ARE
C     USED INTERCHANGABLY AND IS DENOTED BY THE SYMBOL≠.
C     THIS SYMBOL IS USUALLY USED AS A SUFFIX, SUCH AS X≠  .
C
C
C     A,B,C,D, ARE THE HOPE PLANE CONSTANTS
C     OBTAINED FROM THE CANONICAL FORM
C
C     ISEND IS THE LENGTH OF THE DEFINED SYMBOL TABLE
```

10/22/69

```
C
C
C     THE XMAT9 ARRAY WILL HOLD THE MATRIX OF COEFFICIENTS FOR
C     COORDINATE  CONVERSION (TRANSLATION AND/OR ROTATION)
C     FROM THE MAJOR SYSTEM TO THE HOPE SYSTEM
C
C     FROM THE EQUATIONS
C
C      X≠ = T11(X-X0) + T21(Y-Y0) + T31(Z-Z0)
C
C      Y≠ = T12(X-X0) + T22(Y-Y0) + T32(Z-Z0)
C
C      Z≠ = T13(X-X0) + T32(Y-Y0) + T33(Z-Z0)
C
C
C     WHERE-
C
C         THE X≠ AXIS HAS DIRECTION COSINES- T11, T21, T31
C
C         THE Y≠ AXIS HAS DIRECTION COSINES- T12, T22, T32
C
C         THE Z≠ AXIS HAS DIRECTION COSINES- T13, T23, T33
C
C         THE PRIME SYSTEM ORIGIN IS AT (X0,Y0,Z0)
C
C     IT IS FORMED AS FOLLOWS-
C
C             I                                                I
C             I                                                I
C             I                                                I
C             I      T11         T21         T31       C1      I
C             I                                                I
C             I      T12         T22         T32       C2      I
C             I                                                I
C             I      T13         T23         T33       C3      I
C             I                                                I
C             I                                                I
C             I
C
C     WHERE-
C         C1 = -(T11*X0   +   T21*Y0   +   T31*Z0)
C
C         C2 = -(T12*X0   +   T22*Y0   +   T32*Z0)
C
C         C3 = -(T13*X0   +   T23*Y0   +   T33*Z0)
C
C
C
C
      COMMON/12/A,B,C,D,IDSEND,XMAT9(16)
C
C
C     THE KR ARRAY HOLDS THE (X≠Y≠) INFORMATION OF ALL POINTS OBTAINED
C     ON ONE COMPLETE PASS THROUGH THE INNER LOOP COMMENCING AT
C     STATEMENT 1004.
C
```

10/22/69

```
C     ≠≠IP≠≠ COUNTS THE NUMBER OF PAIRS
C
       COMMON/4/IP,KR(100)
       REAL KR
C
C
C     THE KS ARRAY HOLDS THE (X≠Y≠) INFORMATION OF ALL POINTS OBTAINED
C     BY PASSES THROUGH BOTH LOOPS, I.E. THE OUTER LOOP COMMENCING AT
C     STATEMENT 1001 AND THE INNER LOOP THAT STARTS AT 1004,
C     ≠≠IS≠≠ COUNTS THE NUMBER OF PAIRS
C
       COMMON/5/IS,KS(500)
       REAL KS
C
C     ≠≠IONCE≠≠ IS A FLAG THAT CONTROLS THE CALLING OF SUBROUTINE DLINE
C
       COMMON/6/IONCE
C
C     ≠≠ARRAY≠≠ IS FOR THE FORMATION OF THE PLOT STRING,
C
       DIMENSION ARRAY (254)
C
C     THE ≠≠IPOISET≠≠ ARRAY IS USED AS A PARAMETER IN THE ENCODE COMMAND
C     TO COMPOSE BCD FOR THE PLOT.
C
       DIMENSION IPOISET(9)
C
C
C        *         *         *         *         *         *         *
C     THE PROGRAMMING ACTION STARTS HERE
C
C     ≠≠YES≠≠ AND ≠≠NO≠≠ ARE THE TWO POSSIBLE ANSWERS FOR
C     FUNCTION ISITOK,
C
C
       YES=1
       NO=0
C
C
C
C
C     A TOP OF FORM ACTION TO SEPARATE CROSEC OUTPUT FROM EARLIER
C     APT OUTPUT,
C
       PRINT 40
   40 FORMAT (1H1)
C
C
C LOCATING THE NAME HOPE IN THE DEFINED SYMBOL TABLE
C DEFINED SYMBOL TABLE BEGINS AT ITAB11 AND ENDS AT ITAB12
C     ITS TOTAL LENGTH IS THEREFORE (ITAB12-ITAB11)+1
C
C  ≠HOPE≠ IS THE NAME GIVEN TO THE REFERENCE PLANE
C
C
       IDSEND=ITAB12-ITAB11
```

10/22/69

```
      IDSEND=IDSEND+1
      DO 900 I=1,IDSEND,2
      ISAVE=I
      I=I-1
C     INFORMATION IN THE DST APPEARS IN PAIRS AND
C     THE FIRST WORD OF THE FIRST PAIR IS AT LOCATION ≠≠ITAB11≠≠ IN
C     ARRAY ≠≠JTABL≠≠
C     THE FIRST WORDS OF THESE PAIRS ARE LEFT-JUSTIFIED BCD NAMES
C     TO A MAXIMUM OF SIX LETTERS AND BLANK FILLED
C
C
C                                H O P E
      IF (JTABL(ITAB11+I),EQ,30464725606060608)  800,900
C
C     ≠≠PLOTPLNO≠≠ IS THE INCREMENTAL INDEX TO THE SECOND WORD IN THE
C     ≠≠HOPE≠≠ PAIR.
C
  800 PLOTPLNO=I+1
      GO TO 600
  900  I=ISAVE
C
C     IF NO HOPE CARD HAS BEEN IDENTIFIED RETURN TO SOURCE OF CALL
C
  700  PRINT 701
  701  FORMAT (1X,*NO HOPE CARD*)
       GO TO 1401
C        *         *         *         *         *         *         *
C        *         *        C O M M E N T        *         *         *
C        *         *         *         *         *         *         *
C        *         *         *         *         *         *         *
C     DEFINING THE CANONICAL FORM OF THE PLANE AS
C        X * COS ALPHA  + Y * COS BETA + Z * COS GAMMA  = P
C     WHERE THE COSINE TERMS ARE DIRECTION COSINES OF X, Y, AND Z
C     RESPECTFULLY,
C     THE CANONICAL FORM OF THE ≠HOPE≠ PLANE IS CALLED AND STORED IN
C     THE REFERENCE ARRAY, WHERE
C             ≠RCOSA≠ CONTAINS COSINE ALPHA
C             ≠RCOSB≠ CONTAINS COSINE BETA
C             ≠RCOSG≠ CONTAINS COSINE GAMMA ,AND
C             ≠R HOPE P≠ CONTAINS THE ≠P≠ OR CONSTANT VALUE
C        *         *         *         *         *         *         *
C        *         *         *         *         *         *         *
C
C     ≠≠KANSURF≠≠ IS DESIGNED TO HOLD POINTERS TO CANONICAL FORMS
C     IN STATEMENT 600 IT IS HOLDING THE POINTER TO THE ≠≠HOPE≠≠PLANE
C     CANONICAL FORM
C
  600  KANSURF = JTABL (ITAB11 + PLOTPLNO)
C
C     THE ≠≠CANGET≠≠ SUBROUTINE FETCHES THE CANONICAL FORM POINTED TO
C     BY ≠≠KANSURF≠≠ AND STORES ITS ELEMENTS IN THE ≠≠DEFSTO≠≠ ARRAY,
C
       CALL CANGET
       RCOSA = DEFSTO(4)
       RCOSB=DEFSTO(5)
       RCOSG = DEFSTO(6)
```

10/22/69

```
      R HOPE P  = DEFSTO (7)
C
C     USING THE SHORTHAND SYMBOLS FOR THE PLANE CONSTANTS-A,B,C,D
C
      A=RCOSA
      B=RCOSB
      C=RCOSG
      D=RHOPEP
C
C     SUBROUTINE TESTHOPE FILLS IN THE ≠≠XMAT9≠≠ ARRAY
C
      CALL TEST HOPE (A,B,C,D,XMAT9)
C
C     ANOTHER TOP-OF-FORM
C
      PRINT 40
C        *          *          C O M M E N T          *          *          *
C        *          *          *          *          *          *          *
C     SINCE THE COSINE OF THE ANGLE BETWEEN TWO PLANES IS EQUAL TO THE
CSUM OF THE PRODUCT OF CORRESPONDING DIRECTION COSINES, THIS FACT IS
CTAKEN ADVANTAGE OF BY COMPUTING THE ANGLE BETWEEN EACH DEFINED PLANE
CAND THE REFERENCE PLANE.  SPECIAL ATTENTION IS CALLED TO THOSE PLANES
CTHAT ARE EITHER PARALLEL OR PERPENDICULAR TO THE REFERENCE PLANE.
C        *          *          *          *          *          *          *
C        *          *          *          *          *          *          *
C        *          *          *          *          *          *          *
      PRINT 41
   41 FORMAT(30X,*ANGLES BETWEEN HOPE AND THE DEFINED PLANES*,/////)
      DO 920 J=1,IDSEND,2
      I=ITAB11+J-1
C
C
C     A MORE DETAILED EXAMINATION OF THE SECOND WORD IN THE DST PAIRS
C     IS NEEDED TO AVOID PICKING UP ITEMS THAT ARE NOT PLANES.
C     ≠≠IONE≠≠ PICKS UP THE FIRST WORD OF THE PAIR
C     ≠≠ITWO≠≠ PICKS UP THE SECOND WORD OF THE PAIR
C
C
      IONE=JTABL(I)
      ITWO=JTABL(I+1)
C
C     IF A NAME IS UNDEFINED THE SECOND WORD CONTAINS THE BCD WORD
C     ≠≠NODEFX S≠≠.
C
C     IF A NAME IS INCORRECTLY DEFINED
C     THE SECOND WORD CONTAINS THE BCD WORD ≠≠NODEFI S≠≠.
C     THIS TEST PREVENTS OPERATING ON THESE SITUATIONS.
C
      IF(ITWO.EQ.NNODEFX.OR.ITWO.EQ.NNODEFI)920,921
C
C     ≠≠ISTDWD≠≠IS SET EQUAL TO THE SECOND WORD BECAUSE
C     IT IS NOW ASSUMMED TO BE A STANDARD WORD BY APT DEFINITIONS.
C
  921 ISTDWD=ITWO
C
C     SUBROUTINE STUNPK IS CALLED UPON TO UNPACK THE STANDARD WORD
```

```
                                                      10/22/69
C
      CALL STDUNPK
C
C     ≠≠ISTDTBL≠≠ RECEIVES BYTA AND A ≠≠4≠≠ REPRESENTS A CANØNICAL FØRM
C
      IF(ISTDTBL,EQ,4)50,920
C
C     IT IS NØW SAFE TØ PRØCEED AND EXAMINE THE CANØNICAL FØRM ITSELF
C
   50 KANSURF=ITWØ
      CALL CANGET
C
C     ≠≠ISAM≠≠ PICKS UP THE 4 RIGHTMØST BITS ØF THE FIRST CANØNICAL WØRD
C
            ISAM=(DEFSTØ(1),AND,17B)
C
C     IF ≠≠ISAM≠≠ IS EQUAL TØ A ≠≠3≠≠ THEN IT IS A PLANE CANØNICAL FØRM
C
      IF (ISAM,EQ,3) 930,920
C
C     THE SUM ØF PRØDUCTS ØF CØRRESPØNDING DIRECTIØN CØSINES
C     ØF TWØ PLANES,
C
  930 CØSSAM =RCØSA*DEFSTØ(4)+RCØSB*DEFSTØ(5)+RCØSG*DEFSTØ(6)
C
C     IF ≠≠CØSSAM≠≠ IS NEGATIVE, CHECK IT FURTHER AT STATEMENT 938
C
C     IF ≠≠CØSSAM≠≠IS ZERØ THEN THE PLANES ARE PERPENDICULAR
C
      IF (CØSSAM)  938,931,932
  931 PRINT 933, IØNE
  933 FØRMAT(1X,A8,1X,32HPERPENDICULAR TØ REFERENCE PLANE,///)
        GØ TØ 920
C
C     IF ≠≠CØSSAM≠≠IS ≠≠1≠≠ THEN THE PLANES ARE PARALLEL
C
  932 IF (CØSSAM,EQ.1) 935,937
  935 PRINT 934,IØNE
  934 FØRMAT (1X,A8,1X,27HPARALLEL TØ REFERENCE PLANE,///)
      GØ TØ 920
C
C     IF ≠≠CØSSAM≠≠ BETWEEN -1 AND 0
C     PRØCEED TØ TAKE THE ARC CØSINE
C     ØTHERWISE GØ TØ END ØF LØØP
C
  938 IF(CØSSAM,GT.(-1.0))937,920
C
C     IF ≠≠CØSSAM≠≠ IS NØN-ZERØ, AND NØT = 1,  TAKE THE ARC CØSINE
C     AND CØNVERT TØ DEGREES
C
  937 ANGSAM =ACØS(CØSSAM)*57.32
      PRINT 936,IØNE,ANGSAM
  936 FØRMAT (1X,A8,1X,2HAT,F10,1,1X, 26HDEGREES TØ REFERENCE PLANE,///
     1)
  920 CONTINUE
C
```

10/22/69

```
C      NOTHING IS DONE WITH THIS INFORMATION ON ANGLES BETWEEN THE PLANES
C      OTHER THAN PRINTING IT OUT FOR ITS USEFULNESS TO THE PROGRAMMER
C
C
        PRINT 40
C         *        *        *        *        *        *        *
C         *        *        *        *        *        *        *
C         *        *        *        *        *        *        *
C         *        *       C O M M E N T      *        *        *
C         *        *        *        *        *        *        *
C         *        *        *        *        *        *        *
C      THE MINIMUM AND MAXIMUM VALUES OF THE MAJOR COORDINATE SYSTEM
C      VARIABLES (I,E, X,Y,Z) ARE INTRODUCED THROUGH A SEQUENTIAL PAIR
C      OF SPECIALLY NAMED POINTS - ≠≠HOXYMIN≠≠ AND ≠≠HOXYMAX≠≠,
C
C      THESE VALUES ARE PICKED UP VIA THE DEFINED SYMBOL TABLE
C
C      THE COORESPONDING VALUES IN THE PRIME SYSTEM ARE COMPUTED,
C      (E,G.X≠MIN= F(XMIN,YMIN,ZMIN)
C
C      A MINIMUM AND A MAXIMUM VECTOR MAGNITUDE IN THE PRIME SYSTEM ARE
C      COMPUTED
C
C      (E,G. VMIN =SQRT(X≠MIN**2 + Y≠MIN**2 + Z≠MIN**2)
C
C      THE LARGEST OF THESE VECTORS IS SCALED TO COVER 5 INCHES OF PLOT
C      LENGTH,
C
C      THE POINTS OF INTERSECTION ARE COMPUTED
C                FIRST IN X,Y,Z  COORDINATES
C                SECOND IN X≠,Y≠,Z≠COORDINATES
C
C      THE Z≠ VALUES ARE IGNORED AND SHOULD BE ZERO
C
C      THE PLOTTING COORDINATES (XP,YP) ARE COMPUTED USING THE SCALE
C      FACTOR AND AN ADDITIVE BIAS OF 5,0 IN BOTH DIRECTIONS,
C
C
C      IF THE RESULTING VALUES OF XP AND YP ARE NEGATIVE OR EXCEED 10,0
C      THEY ARE REJECTED AS EXCEEDING THE BOUNDS OF THE LIMITING VECTOR
C
C
C
C         *        *        *        *        *        *        *
C         *        *        *        *        *        *        *
C         *        *        *        *        *        *        *
C         *        *        *        *        *        *        *
C
C      THE DST TABLE IS SEARCHED FOR THE NAMES
C      ≠≠HOXYMIN≠≠ AND ≠≠HOXYMAX≠≠,
C
        DO 950 K=1,IDSEND,2
        KSAVE=K
        K=K-1
C                            H O X Y M I N
        IF(JTABL(I_AB11+K),EQ,3046677044314560B.AND.
```

```
                                                              10/22/69
C                                  H O X Y M A X
    2     JTABL(ITAB11+K+2).EQ.30466770442167608) 951,950
C
C      INCREMENTAL INDECIES TO SECOND WORD LOCATIONS IN DST
C
  951  IMINNO =K+1
       IMAXNO=K+3
       GO TO 952
  950  K=KSAVE
       PRINT 954
  954  FORMAT (1X,*HOXYMIN AND HOXYMAX NOT FOUND*)
      STOP
C
C      THE MINIMUM VALUES ARE EXTRACTED FROM THE HOXYMIN CANONICAL FORM
C
  952  KANSURF=JTABL(ITAB11+IMINNO)
       CALL CANGET
       RAWXMIN=DEFSTO(4)
       RAWYMIN=DEFSTO(5)
       RAWZMIN=DEFSTO(6)
C
C      THE MAXIMUM VALUES ARE EXTRACTED FROM THE HOXYMAX CANONICAL FORM
C
       KANSURF=JTABL(ITAB11+IMAXNO)
       CALL CANGET
       RAWXMAX=DEFSTO(4)
       RAWYMAX=DEFSTO(5)
       RAWZMAX=DEFSTO(6)
       PRINT 957
  957 FORMAT (1X,*IN ORIGINAL COORDINATE SYSTEM*,/,/,/)
       PRINT 955,RAWXMIN,RAWYMIN,RAWZMIN,RAWXMAX,RAWYMAX,RAWZMAX
  955  FORMAT(5X,*MINIMUM VALUES*,/,/,3(10X,F10.5,/),/,
    1  5X,*MAXIMUM VALUES*,/,/,3(10X,F10,5,/))
C
C      THE COORESPONDING MINIMUM VALUES IN THE HOPE PLANE,
C      XPMIN,YPMIN,ZPMIN
C      ARE COMPUTED IN SUBROUTINE MM USING THE XMAT9 MATRIX
C
       CALL MM (XPMIN,YPMIN,ZPMIN,RAWXMIN,RAWYMIN,RAWZMIN,XMAT9)
C
C      THE SAME WITH THE MAXIMUM VALUES, XPMAX, YPMAX, ZPMAX
C
       CALL MM(XPMAX,YPMAX,ZPMAX,RAWXMAX,RAWYMAX,RAWZMAX,XMAT9)
C
C      THE MINIMUM AND MAXIMUM VECTOR MAGNITUDES ARE COMPUTED
C
       MINVCTOR=SQRT(XPMIN**2+YPMIN**2+ZPMIN**2)
       MAXVCTOR=SQRT(XPMAX**2+YPMAX**2+ZPMAX**2)
C
C      THE GREATEST OF THESE IN MAGNITUDE IS SET EQUAL TO ≠≠REACH≠≠
C
       IF (MINVCTOR.GT.MAXVCTOR)960,970
  960  REACH =MINVCTOR
       GO TO 980
  970  REACH = MAXVCTOR
       GO TO 980
```

UNCLASSIFIED

10/22/69

```
C
C     THE SCALE FACTOR- REACH MUST FIT INTO 5,0 INCHES OF PLOT LENGTH
C
  980  SCALE=5,0/REACH
C
C     PRINT OUT THE CAL-COMP PLOT SCALE
C
      PRINT 956
  956  FORMAT (/,/,1X,*IN HOPE PLANE*,/,/)
       PRINT 958,SCALE
  958 FORMAT(1X,*SCALE,..,*,F5,2,*  INCHES OF PLOT LENGTH = 1 INCH OF PAR
     1T LENGTH*)
C
C
C     CALCULATE AND PRINT OUT THE GERBER SCALE FACTOR NEEDED TO
C     PRODUCE A 1:1 PLOT, THAT IS, A PLOT ON WHICH ONE UNIT OF LENGTH
C     IS EQUAL TO ONE UNIT OF LENGTH ON THE ORIGINAL PART DRAWING,
C
       SCLFAC=1/SCALE
       PRINT 959,SCLFAC
  959 FORMAT (/,/,1X,*TO OBTAIN A 1 TO 1 PLOT ON THE GERBER USE A SCALE*
     1,* FACTOR IN BOTH X AND Y OF *,F5,2)
       PRINT 40
       CALL PLOTS (ARRAY,254,10)
C
C     PLACE A RIGHT ANGLE INTERSECTION AT THE ORIGIN (5,0,5,0)
C
       CALL SYMBOL (5,0,5,0,0,20,3,0,-1)
C
C
C     PLACE A SCALE MARK IN THE LOWER LEFT HAND CORNER
C     SHOWING THE CAL-COMP DISTANCE COORESPONDING TO ONE INCH OF
C     PART LENGTH,
C
       CALL PLOT (0,0,3)
       CALL PLOT (0,0,2)
       CALL PLOT (0,-0,25,2)
       CALL PLOT (SCALE,-0,25,2)
       CALL PLOT  (SCALE,0,2)
       CALL PLOT (SCALE,0,3)
C
C        *         *         *         *         *         *         *
C        *         *         *         *         *         *         *
C        *         *       C O M M E N T         *         *         *
C        *         *         *         *         *         *         *
C        *         *         *         *         *         *         *
C           THE SUBROUTINE ≠SIMEQ≠ THAT SOLVES THREE PLANE EQUATIONS
C     SIMULTANEOUSLY IS UTILIZED TO OBTAIN POINTS OF INTERSECTION
C     ON THE ≠≠HOPE≠≠ PLANE,  A LINE OF INTERSECTION IS IDENTIFIED
C     WHEN A PLANE IS INVOLVED IN MORE THAN ONE POINT,  FOR THIS
C     REASON AN OUTER LOOP IS CYCLED ≠≠IDSEND≠≠ TIMES, AND AN INNER
C     LOOP CYCLED ≠≠IDSEND * IDSEND≠≠ TIMES SO THAT EACH PLANE IS
C     HELD CONSTANT (ALONG WITH ≠≠HOPE≠≠ OF COURSE) AS ALL OTHER
C     PLANES ARE BROUGHT IN FOR TESTING,
C        *         *         *         *         *         *         *
C        *         *         *         *         *         *         *
```

NRL Report 7025

Cross-Sectional Plots of Plane Intersections

An Adaptation of the APT System

KINGSLEY P. THOMPSON

Information Systems Branch
Mathematical and Information Sciences Division

January 27, 1970

NAVAL RESEARCH LABORATORY
Washington, D.C.

UNCLASSIFIED

CONTENTS

ABSTRACT

The APT (Automatically Programmed Tooling) system provides for the software description of geometric forms, the delineation of a tool path, and, for output, discrete positioning information on punched paper tape for use by a numerically controlled machine. The CROSEC (Mod 1) program, described in this report, provides a means of extending the use of the canonical forms of the plane surfaces defined by the programmer in the part program by providing a plotting capability in which the lines in intersection, within specified limits, between a cross-sectional plane and all other defined planes are shown. A visualization of the initial plane framework on which the cutting is to be performed is thereby provided. The cross sectional plane and its dimensional limits are controlled by one plane definition and two point definitions. The plot is supplemented by printer output that aids in the interpretation of the plot. The program is written in the framework of the CDC 3800 APT 2.1 configuration. No additional program overlays or segments are necessary. This report contains a discussion of the method used, subroutine descriptions, listings and flowcharts, implementation aids, and a sample run.

PROBLEM STATUS

This is an interim report on a continuing problem.

AUTHORIZATION

NRL Problem 23Z0001

Manuscript submitted November 17, 1969.

CROSS-SECTIONAL PLOTS OF PLANE INTERSECTIONS

An Adaptation of the APT System

INTRODUCTION

This report contains a complete description of a plotting adaptation of the APT system in use at the Naval Research Laboratory (APT 2.1 on the CDC 3800). In the APT language, a plane can be defined in seven different ways; but after being processed they are all stored in a standard "canonical form." This program, called CROSEC (Mod 1), utilizes such plane canonical forms (of an APT Part Program) to obtain a plot of the intersection of one of the planes, designated as the cross-sectional plane, and all of the other planes that have been defined, processed, and stored at the end of APT Section 1. (See Ref. 1 for a detailed description of the APT system.)

The purpose of the program, primarily, is as an aid, an extra tool for the programmer in debugging his program. The defined surfaces as stored in their canonical forms provide a convenient starting point for geometric considerations. It is assumed that syntactic errors have already been discovered and corrected and that in using CROSEC the programmer wishes to verify that the surfaces he has defined do indeed describe the piece he wishes to have worked on by the tool. The hope is that verification can be accomplished easily if he can get a look at any cross section of his choosing through the conglomerate of the starting surfaces. He realizes that the plot might require some interpretation because defined surfaces intersecting together do not fully describe the finished piece. However, he accepts this limitation and looks upon the output as a working drawing, a picture of the output of Section 1. By means of this drawing and the accompanying identifying information from the printer, he should be able to make some significant debugging progress. Perhaps he will discover a section of surface that is defined improperly or over defined, or a combination of surfaces that could be redefined in a simpler manner. Also, he may discover a portion of surface that he has not yet defined and other events of this nature.

The report discusses plane equations, outlines the method of obtaining a coordinate system in the cross-sectional plane, describes the plot, and points out the limitations of the program. Also included are flowcharts, program descriptions, a complete listing, and details of implementation with APT 2.1. An example is introduced early and followed through the complete process in full detail.

DISCUSSION

The Plane

The *canonical* form for the plane, as defined in the APT system, is given by

$$AX + BY + CZ = D, \tag{1}$$

where*

*As quoted from pp. 80 and A-1 of Ref. 1.

A = X-Component of unit normal vector to plane
B = Y-Component of unit normal vector to plane
C = Z-Component of unit normal vector to plane
D = Normal distance from plane to origin.

Consider the *normal* form of the plane equation, namely

$$XCOS\alpha_x + YCOS\alpha_y + ZCOS\alpha_z = L, \tag{2}$$

where the coefficients for the X, Y, and Z coordinates in Eq. (2) are the direction cosines of the positive normal vector N from the origin to the plane. The direction cosines of the unit normal vector are identical with the direction cosines of N. The length, or absolute value, of this normal is L. Therefore, equating similar terms from Eqs. (1) and (2) yields

$$A = COS\alpha_x$$

$$B = COS\alpha_y$$

$$C = COS\alpha_z$$

$$D = L = |N| .$$

Thus, for example, A = 0, B = 1, C = 0, D = 1 is the plane passing through the point (0, 1, 0), parallel to the XZ coordinate plane, with direction angles of 90, 0, and 90 degrees, respectively, to the three axes.

Let us now take, for a more detailed example, the plane that passes through the points (1,0,0), (0,1,0), and (0,0,1) a unit distance out along each axis. To fit this approach to the definition of a plane, consider the *intercept* form for a plane equation,*

$$\frac{X}{X_1} + \frac{Y}{Y_2} + \frac{Z}{Z_3} = 1, \tag{3}$$

where X_1, Y_2, and Z_3 are the intercepts, i.e. the point (X_1, 0, 0) is the intersection of the X axis with the plane. Similarly with (0, Y_2, 0) and (0, 0, Z_3).

Using now the three unit axis points, already defined, in this intercept form, Eq. (3) leads to the simple and interesting equation

$$X + Y + Z = 1. \tag{4}$$

The correctness of Eq. (4) as truly representing the plane that passes through the three points is easily determined by setting any two of the variables equal to zero, and the remaining variable will be equal to 1. Equation (4) is illustrative of another form of an equation used to describe a plane, the *general* form, where the coefficients of the X, Y, and Z terms are considered to be *direction numbers* of the positive normal to the plane.

To go from the general form to the normal form, it is necessary to compute the direction cosines by dividing each of the coordinate coefficients in turn by the square root of the sum of the squares of all three coefficients. The length of the normal is obtained in a similar fashion by dividing the constant term by the same square-root quantity. Symbolically, if the general form is

$$PX + QY + RZ = S, \tag{5}$$

*All forms of equations for a plane can be found in Section 3, p. 2-1 of Ref. 2.

then

$$A = \text{COS}\alpha_x = \frac{P}{U}$$

$$B = \text{COS}\alpha_y = \frac{Q}{U}$$

$$C = \text{COS}\alpha_z = \frac{R}{U}$$

$$D = L = \frac{S}{U},$$

where

$$U = \sqrt{P*P + Q*Q + R*R.}$$

The normal form for the illustrative plane is therefore

$$\frac{X}{\sqrt{3}} + \frac{Y}{\sqrt{3}} + \frac{Z}{\sqrt{3}} = \frac{1}{\sqrt{3}}. \tag{6}$$

It is easily verified that the sum of the squares of the direction cosines is 1.

It is important in our development to know the coordinates of the point represented by the intersection of the normal N and the plane to which it is perpendicular. These coordinates are obtained by multiplying each of the direction cosines by the length of the normal. Symbolically, (D*A, D*B, D*C). For the illustrative plane the result is (1/3, 1/3, 1/3).

One final consideration regarding these plane equations—our illustrative case has a convenient set of intercept points; however, it is possible to determine the intercepts from the normal form. They are

$$(X_1, Y_1, Z_1) = \left(\frac{D}{A}, 0, 0\right)$$

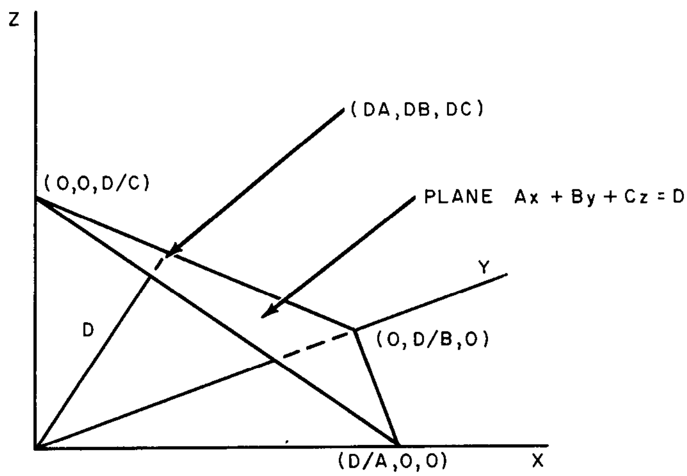
$$(X_2, Y_2, Z_2) = \left(0, \frac{D}{B}, 0\right)$$

$$(X_3, Y_3, Z_3) = \left(0, 0, \frac{D}{C}\right).$$

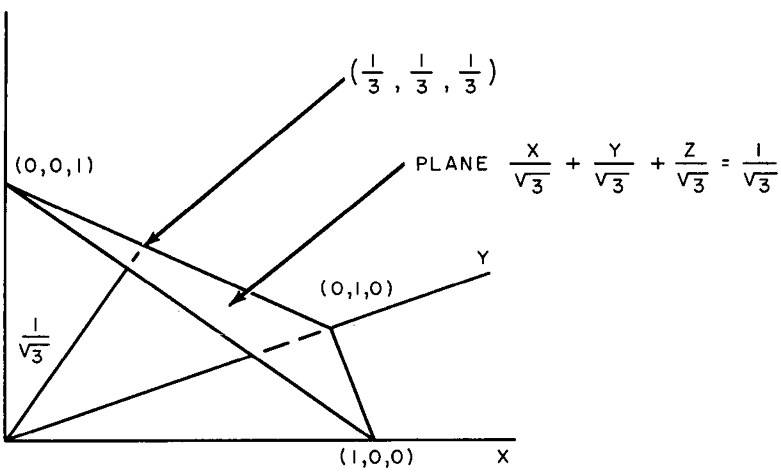
Figure 1 summarizes this initial development using the symbols for the APT canonical form of Eq. (1) in part a of the figure and the actual values of our illustrative plane in part b.

The Cross-Sectional Plane and Its Coordinate System

The plot of Section 1 output uses a coordinate system in the cross-sectional plane, particularized as the HOPE plane. Let us call the coordinate system in the HOPE plane the prime system (i.e., X′, Y′, Z′), in contrast to the original system established by the part programmer known as the X, Y, Z system. The origin of the prime system is the intersection of the normal with the plane, and the Z′ axis is the extension of the normal.



(a) Normal form of plane equation



(b) Equiangular plane in normal form

Fig. 1 - A general and a particular plane

The X′ axis is selected in one of three ways.

1. The "general solution" is the case where the HOPE plane intersects the X axis, and the positive X′ axis is the line passing from the prime origin through the intercept point and lies in the HOPE plane. In terms of the *normal* form the prime origin is (D*A, D*B, D*C) and the intercept point is (D/A, 0, 0) with direction cosines

$$\left[\frac{\left(\frac{D}{A} - D*A\right)}{U}, \frac{-D*B}{U}, \frac{-D*C}{U}\right],$$

where

$$U = \sqrt{\left(\frac{D}{A} - D*A\right)^2 + (D*B)^2 + (D*C)^2}.$$

For our illustrative plane X + Y + Z = 1, the X′ axis for the general solution would be positively directed from the origin with coordinates (1/3, 1/3, 1/3) to the X intercept with coordinates (1, 0, 0). The computation gives the X′ axis, in this case, direction cosines of

$$(\sqrt{2/3},\ -\sqrt{1/6},\ -\sqrt{1/6})$$

2. If the HOPE plane does not intersect the X axis and is parallel to it, then the X′ axis is that line lying in the HOPE plane, parallel to the X axis, with direction cosines (1, 0, 0) commencing at the prime origin.

3. If the HOPE plane is perpendicular to the X axis and parallel to the YZ plane such that the normal is the X axis, then the positive X′ axis is that line lying in the HOPE plane which starts at the prime origin and is parallel to the Z axis with direction cosines (0, 0, 1).

The Y′ axis is defined as the cross product of the Z′ and X′ axes, duly preserving right-handed concepts.

If the direction cosines of the X′ axis are T_{11}, T_{21}, T_{31}, and those of the Y′ axis T_{12}, T_{22}, T_{32} and those of the Z′ axis T_{13}, T_{23}, T_{33}, then

$$T_{12} = T_{23}*T_{31} - T_{21}*T_{33} \tag{7}$$

$$T_{22} = T_{11}*T_{33} - T_{13}*T_{31} \tag{8}$$

$$T_{32} = T_{13}*T_{21} - T_{11}*T_{23}. \tag{9}$$

Thus, for our illustrative plane, the direction cosines of Z′ are $(1/\sqrt{3},\ 1/\sqrt{3},\ 1/\sqrt{3})$ and of X′ are $(\sqrt{2/3},\ -\sqrt{1/6},\ -\sqrt{1/6})$. Using these equations, we find that the directions cosines for Y′ are $(0,\ \sqrt{1/2},\ -\sqrt{1/2})$.

The equations required to convert any point in space from the original coordinates to the prime coordinates are

$$X' = T_{11}\ (X - X_0) + T_{21}\ (Y - Y_0) + T_{31}\ (Z - Z_0),$$

$$Y' = T_{12}\ (X - X_0) + T_{22}\ (Y - Y_0) + T_{32}\ (Z - Z_0),$$

and

$$Z' = T_{13}\ (X - X_0) + T_{23}\ (Y - Y_0) + T_{33}\ (Z - Z_0),$$

where (X_0, Y_0, Z_0) is the prime origin defined in terms of X, Y, Z. Expanding yields

$$X' = T_{11}*X + T_{21}*Y + T_{31}*Z + C_1, \tag{10}$$

$$Y' = T_{12}*X + T_{22}*Y + T_{32}*Z + C_2, \tag{11}$$

and

$$Z' = T_{13}*X + T_{23}*Y + T_{33}*Z + C_3, \tag{12}$$

where

$$C_1 = -(T_{11}*X_0 + T_{21}*Y_0 + T_{31}*Z_0),$$

$$C_2 = -(T_{12}*X_0 + T_{22}*Y_0 + T_{32}*Z_0),$$

$$C_3 = -(T_{13}*X_0 + T_{23}*Y_0 + T_{33}*Z_0).$$

For the illustrative case where $(X_0, Y_0, Z_0) = (D*A, D*B, D*C) = (1/3, 1/3, 1/3)$, the matrix corresponding to the coefficients of Eqs. (10), (11), and (12), is

$$\begin{bmatrix} \sqrt{\frac{2}{3}} & -\sqrt{\frac{1}{6}} & -\sqrt{\frac{1}{6}} & 0 \\ 0 & \sqrt{\frac{1}{2}} & -\sqrt{\frac{1}{2}} & 0 \\ \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} & -\sqrt{\frac{1}{3}} \end{bmatrix}.$$

Figure 2 illustrates the definition of the prime axes in the HOPE plane. In this instance the HOPE plane is our illustrative plane $X + Y + Z = 1$.

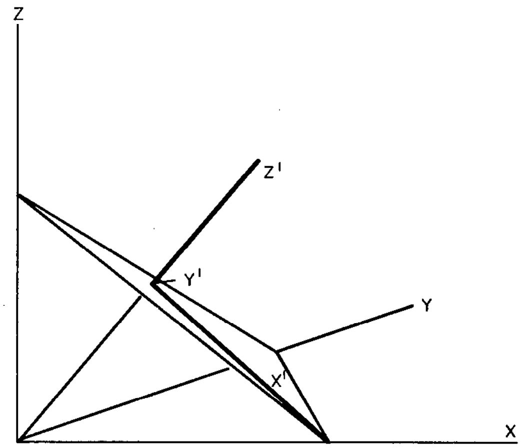


Fig. 2 - Prime axes in the equiangular HOPE plane

Points and Lines in the Cross-Sectional Plane

So far we have established an understanding of the various forms of the equations that describe a plane and have described an algorithm for defining a translation-rotation matrix for converting points from the part programmer's coordinate system to a coordinate system in the HOPE or cross-sectional plane. We can now ask, How is it determined whether any APT-defined point in general lies in the HOPE plane? Simply by substituting the X, Y, and Z values of the point into the equation for the plane to determine if an equality exists; e.g., Does the point (1/3, 1/3, 1/3) lie in the plane $X + Y + Z = 1$? Obviously yes, since $1/3 + 1/3 + 1/3 = 1$.

If the equations for the three planes are solved simultaneously, a point that lies in all three planes is the result. If one of these planes is the HOPE plane, then there is no doubt that the point is in the cross section.

Consider now a situation in which the HOPE plane and a plane A are consecutively solved with two other planes B and C (defined such that no two planes of HOPE, A, B and HOPE, A, C are parallel), resulting in two points P_1 and P_2. Both of these points are simultaneously in both the HOPE plane and in plane A; in fact, the line segment joining P_1 and P_2 is a portion of the line of intersection between HOPE and A. Such a procedure if carried out with all the defined planes, will result in an entire network of lines of inter section. Each point as obtained is put through the matrix to obtain its definition in terms of the HOPE coordinate system to enable it to be plotted. If a minimum and a maximum value for each coordinate is specified, then many superfluous points of intersection can be eliminated.

Let us now consider what happens when the equiangular plane X + Y + Z = 1 intersects some planes defined in a part program, an actual situation. (Refer to sample run with "PARTNO TESTING" on p. 63.) Figure 3 depicts a simple part in three views. It is to be noted that there are eight defined planes. Figure 4 shows the cross-section network of intersecting lines obtained with the plane X + Y + Z = 1 as the HOPE plane intersecting "PARTNO TESTING." Using only the information provided in Figs. 3 and 4, can you distinguish between the proper intersection outline and those lines that are extraneous? It is an intersecting exercise, well worth spending a few minutes on.

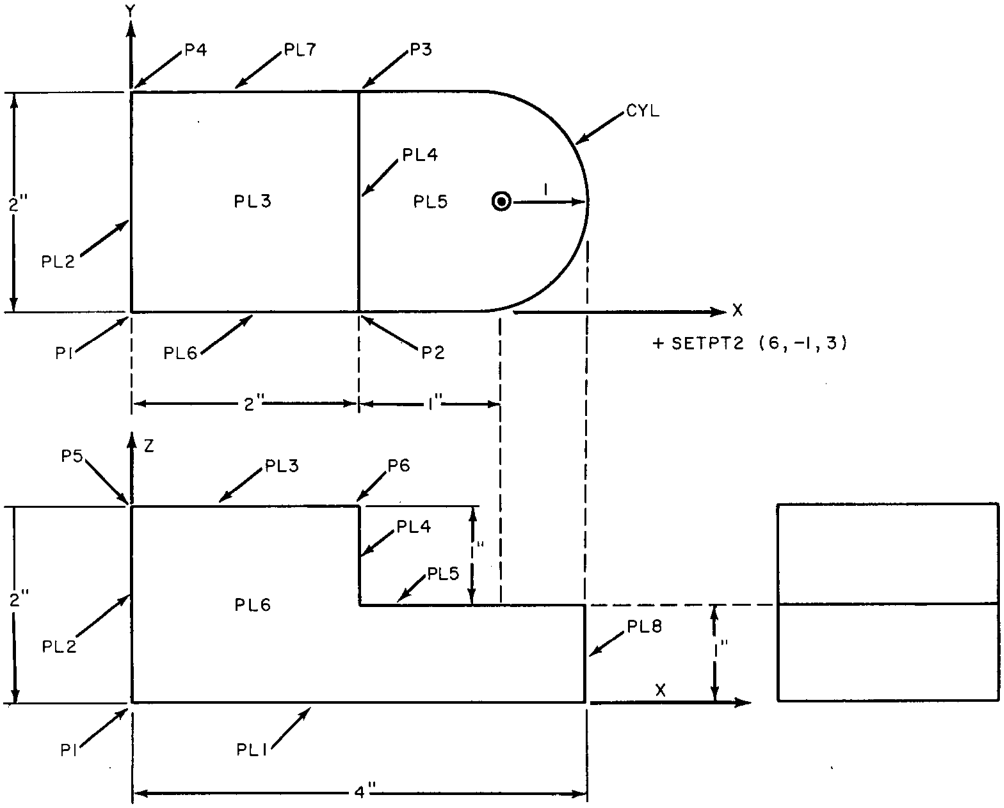


Fig. 3 - "PARTNO TESTING" with eight defined planes

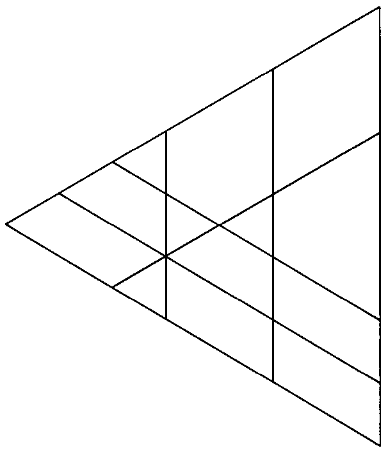


Fig. 4 - Cross section obtained from PARTNO TESTING and equiangular plane

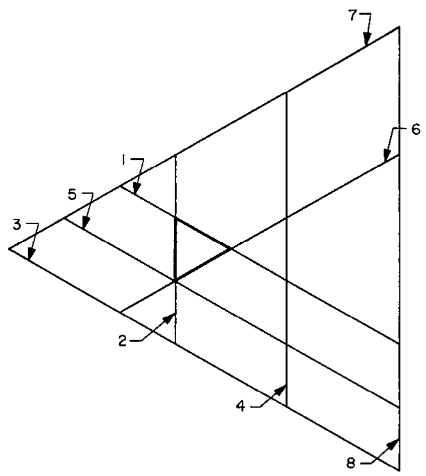


Fig. 5 - Lines of intersection identified by plane number

The solution to this question will now be presented.

The accompanying printout identifies points of intersection from which the lines of intersection were drawn. The coefficients of these points are given in the X, Y, Z system, in the X′, Y′, Z′ system, and in plotter coordinates along with the plane numbers associated with each point. There are 19 such points in Fig. 4. These point data constitute background information, available when needed. However, rather than clutter up the plot with point symbols and identifying numbers, only the lines of intersection have been drawn using a minimum point and a maximum point. From these two points the slope and the prime-system intercept have been computed. This information is also printed along with the plane name associated with the line. The lines have been named in Fig. 5. There are eight planes identified in the sketch of the part in Fig. 3 and there are eight lines in the intersection plot.

Verification of the correctness of the numbering of these lines of intersection can be achieved by noting that parallel *lines* in the plot correspond to parallel *planes* on the part. Sure enough, plane 1 representing the bottom of the piece is parallel to plane 3, the top, and is also parallel to plane 5 which represents the step surface. In Fig. 5 the lines labeled 1, 5, and 3 are parallel lines.

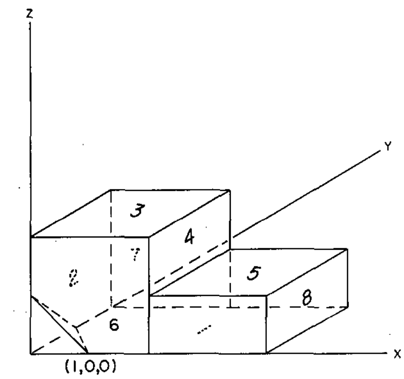


Fig. 6 - Perspective view of PARTNO TESTING (without the cylinder) with eight planes and the intersecting equiangular HOPE plane

In a similar fashion the lines of intersection for planes 2, 4, and 8 confirm a parallelism between these planes, all three of them being vertical sides in the part shown in Fig. 3. Finally, plane 6 (the front) is parallel to plane 7 (back), and the lines with these numbers are parallel in Fig. 5.

Figure 6, which shows a perspective view of the planes of the part and the equiangular HOPE plane, demonstrates the fact that the lines of intersection on the surface of the piece lie in planes 6 (front), 1 (bottom), and 2 (left side). The triangle formed by the lines of intersection of these three planes is accented in Fig. 5 and is the correct answer.

It is an interesting consequence of the situation that since plane X + Y + Z = 1 is equiangular to each of the three coordinate planes, and that since all eight of the defined planes in the simple part are either identical with or parallel to a coordinate plane, that HOPE is also equiangular to these eight planes, namely 57.4 degrees.

There is one other interesting observation to make about Fig. 5. Because of the equiangular characteristic of the cross-sectional plane, a reverse type of statement can be made. Any equilateral triangle in this figure corresponds to three mutually perpendicular planes. Besides the set already discussed $\{6, 1, 2\}$, eleven others can be identified. These are $\{1, 2, 7\}$, $\{1, 4, 6\}$, $\{1, 6, 8\}$, $\{2, 3, 6\}$, $\{3, 7, 8\}$, $\{1, 7, 8\}$, $\{5, 7, 8\}$, $\{2, 3, 7\}$, $\{3, 4, 7\}$, $\{3, 6, 8\}$ and $\{3, 7, 8\}$. The last one, which is the outermost triangle, corresponds to the planes which are perpendicular to the maximum dimensions X = 4, Y = 2, and Z = 2, respectively.

There is a lot of information to be gleaned from the cross-sectional plot once it is understood and assimilated.

When the restrictions of XMIN, YMIN, ZMIN = (0, 0, 0) and XMAX, YMAX, ZMAX = (4, 2, 2) are placed on the cross-section points, then the network is reduced to that shown in Fig. 7, the true triangle of intersection with the part.

CROSEC Limitations

CROSEC in its present form (Mod 1) has some definite limitations. Some of the initially obvious ones are as follows:

1. There is no identification of defined points that might lie in the HOPE plane.

2. The plot does not contain any intersections of the HOPE plane with quadrix surfaces.

3. No tool motion information is present.

4. The plot resulting from the lines of intersection of defined planes with HOPE can be initially confusing, since some of the lines might have no immediate relation to the finished part, and the prime coordinate system might not be simply oriented with respect to the X, Y, Z system. However, a consideration of the original part program along with the printout can soon make the plot understandable.

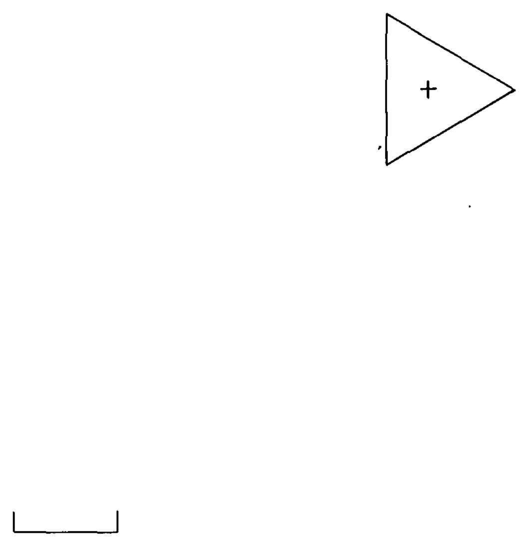


Fig. 7 - CROSEC (Mod 1) output plot for sample run with PARTNO TESTING

CONCLUSION

CROSEC (Mod 1) is a first step. In its present form, it can be helpful in a limited fashion. The intersection with curved surfaces is missing, as well as the path of the cutting tool in the cross-sectional plane. It is planned to incorporate these features in further work on this project.

ACKNOWLEDGMENT

The author expresses his appreciation to the Engineering Services Division for their cooperation in supplying information on the APT system in operation. In particular Lloyd Murphy, Nicholas Maddage, Donald Woods, and Jay Williams, Numerical Control Programming personnel of that Division, were particularly helpful. Also helpful, in more computer-related aspects, were Gary Flenner and Anna Byrd Mays of the NRL Research Computation Center, who worked on the initial implementation of APT at NRL.

REFERENCES

1. "APT Reference Manual for CDC 3400 and 3600 Computer Systems," Publication 60132800, Control Data Corporation, Palo Alto, Calif., Aug. 1965

2. Korn, G.A., and Korn, T.M., "Mathematical Handbook for Scientists and Engineers," New York:McGraw-Hill, 1961

Appendix A

SUBROUTINE DESCRIPTIONS WITH FLOWCHARTS AND LISTINGS

SUBROUTINE CROSEC

The CROSEC subroutine (Fig. A1) controls the Section 1 plotting of the intersections of defined planes with the HOPE cross-sectional plane. For convenience it incorporates all of the common area of Section 0 and Section 1 into itself without attempting to discard portions it does not need. The APT system subroutines called on are CANGET, SIMEQ and STDUNPK. The in-house library routines called include PLOTS, SYMBOL, PLOT, STOPPLOT, SQRTF, ACOSF, QNSINGL, THEND, STH, ENC, Q1Q10100, Q8QSTOPS, and Q8QDICT. Additional subroutines called by CROSEC and considered a part of the Mod 1 package are TESTHOPE, DELINE, and MM. There is also a function, ISITOK. The flowchart of Fig. A2 shows the relationship between these subroutines.

The CROSEC subroutine assumes that the defined symbol table (D.S.T.) is stored in the JTABL array of numbered common 2 between ITAB11 and ITAB12. It further assumes that the D.S.T. entries consist of pairs of words, the first of which is an eight-symbol Hollerith name, left justified, and, second, an APT "standard word" which includes an integer pointer giving the relative address in JTABL (extended beyond D.S.T.) where the canonical form is stored. These assumptions are standard procedure for the CDC APT. (See the Section 1 description starting on p. 01-1 of Ref. A1.) Accordingly, after executing a top of form, the D.S.T. is searched for the name HOPE, and if this name is not found the subroutine is exited and there is no plot obtained. If HOPE is found, its pointer is stored in KANSURF prior to calling CANGET which fetches and stores the canonical form in the DEFSTO array of the SECT1LOG. After identifying A, B, C, and D from the HOPE canonical form, the TESTHOPE subroutine is called.

TESTHOPE will either stop the run or return with a conversion matrix stored in XMAT9. This matrix will permit the conversion from the X, Y, Z coordinate system to the prime system whose origin is in the HOPE plane, thereby facilitating a two-dimensional plot.

Another top of form is executed.

Since the cosine of the angle between two planes is equal to the sum of the product of their corresponding direction cosines, it is a logical next step to take advantage of this fact and compute the angle between each defined plane and the HOPE plane.

Special care is needed in picking up canonical pointers from the D.S.T. prior to making the cosine computation. Undefined words or incorrectly defined words must be avoided as well as the synonym register which appears at the head of the D.S.T. The standard word must be unpacked by calling Subroutine STDUNPK in order to determine if BYTA contains a 4, representing a canonical form. After the canonical form has been recovered by a call to CANGET, it is necessary to examine the four most right-hand bits of the first word in the set. A 3 in this position identifies the canonical of a plane as opposed to other possible canonical forms such as points, cylinders, etc. In this manner the direction cosines are obtained, and the computation for the angle between the planes can then be performed.

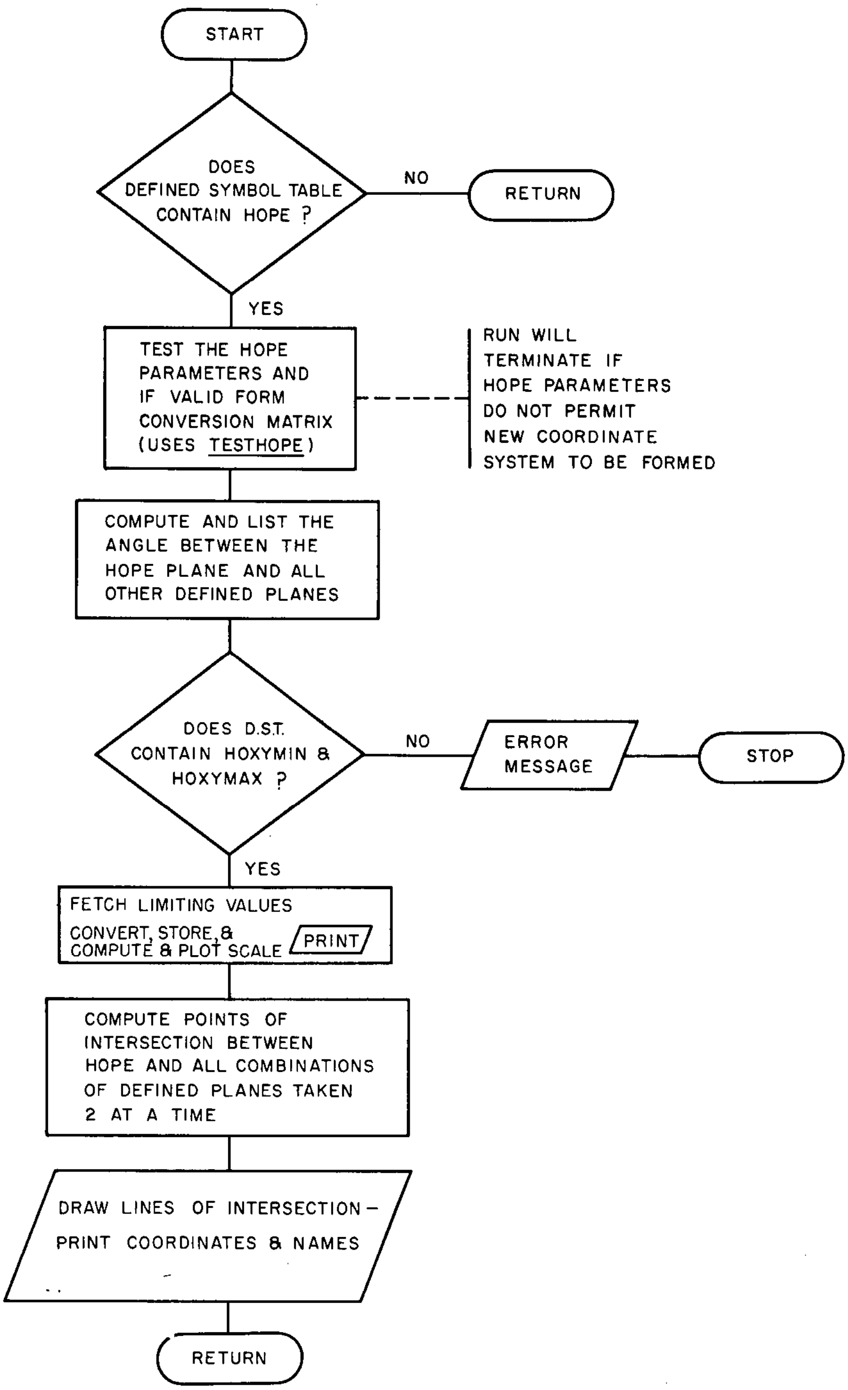


Fig. A1 - Flowchart for CROSEC subroutine

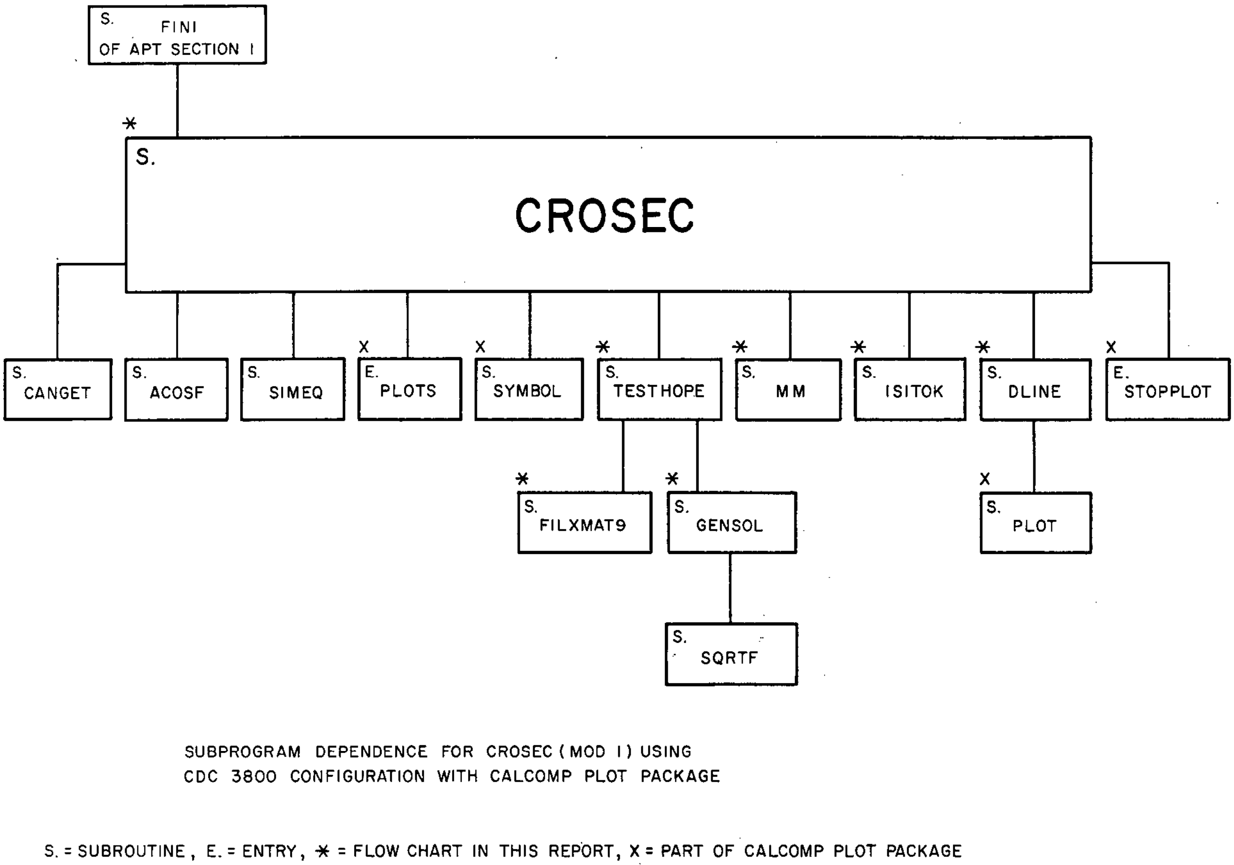


Fig. A2 - Relationship between subroutines called by CROSEC

After making special tests for perpendicular planes and parallel planes, the arc cosine function ACOSF is called and the resulting angle printed out in degrees. This angle has a range of 0 to 180 degrees. All information regarding this analysis of the angles between the planes is outputed through the line printer and is not retained, as it has no consequence to the plot.

Next, the D.S.T. is searched for the names HOXYMIN and HOXYMAX, which must be stored as adjacent pairs in order to be discovered. In the absence of these names, an error message is printed and the run is terminated. The message is "HOXYMIN AND HOXYMAX NOT FOUND." If these names are present, the corresponding values are fetched via CANGET and stored in correspondingly descriptive name locations, such as RAW X MIN.

These raw limiting values are passed through the matrix by calling subroutine MM, thus obtaining their corresponding values in the prime system, (XPMIN, YPMIN, ZPMIN) and (XPMAX, YPMAX, ZPMAX). A printout is now initiated that lists the raw limiting values.

The magnitudes of the minimum and maximum vectors are computed thusly,

$$\text{MINVECTOR} = \sqrt{(\text{XPMIN})^2 + (\text{YPMIN})^2 + (\text{ZPMIN})^2}$$

$$\text{MAXVECTOR} = \sqrt{(\text{XPMAX})^2 + (\text{YPMAX})^2 + (\text{ZPMAX})^2}.$$

The CALCOMP plot is restricted to a 10 × 10 in. space. When we allow for both negative and positive values of X′ and Y′ it follows that 2 times the largest vector must fit into 10 linear inches. The center of the plot is the origin of the prime coordinate system. To accomplish this, the largest of these two magnitudes is set equal to REACH and then divided into 5.0 to obtain the CALCOMP plot factor which is called SCALE. SCALE is therefore the number of inches of plot length that is equivalent to one inch of part length. SCLFAC is the reciprocal of SCALE and is the scale factor needed to obtain a 1:1 Gerber plot. (The details of obtaining a paper tape output for use on the Gerber plotter (Model 875) are discussed in Ref. A2.) Output messages regarding SCALE and SCLFAC are printed, followed by another top of form.

The plot string is initiated.

A + symbol is plotted at (5.0, 5.0), the plot coordinates for the HOPE prime system origin, and a scale mark is placed at the lower left-hand corner of the plot. A somewhat involved double loop is now entered at the heart of which is a call to SIMEQ which solves three plane equations simultaneously, resulting in the output of a common point of intersection (two planes intersect in a line, three planes intersect in a point).

Since the A, B, C, D of the HOPE plane are constant throughout the process, they are stored once and for all in positions 108 through 111 of the DEFTAB array (used by SIMEQ), before the entrance into the looping procedure.

Also initialized at this time are

1. IONCE, a flag which controls the call to subroutine DLINE,

2. IPTNO, the point number counter,

3. IP, the counter for coordinate pairs stored in the KR array. The KR array stores only the points related to the current line about to be drawn, and

4. IS, the counter for coordinate pairs stored in the KS array. (The KS array stores *all* points created by CROSEC.)

The outermost of the two loops commences at statement 1001.

DLINE is called only if IONCE is zero.

The canonical form for a plane is fetched under the same restraints as already described above. If a plane is indicated, its name is saved as NAME1 and its A, B, C, D are stored in DEFTAB 104 through 107. The inner loop commences at statement 1004. By a similar process, the values for the third plane are stored in DEFTAB 100 through 103, and the creation of NAME2 takes place. Then comes the call for SIMEQ followed by the test on JSUBER for proper execution, thereby allowing for the situation of two of the planes being identical.

If a bad situation resulting in no point has occurreu, a transfer is made to the end of the inner loop. If a proper point had been obtained, its X, Y, Z coordinates are picked up from DEFTAB 112 through 114.

These numbers in turn are used to obtain the corresponding prime values and plotting values. The prime values are obtained by use of the conversion matrix via a call on subroutine MM. The plotting values are obtained by means of the formulas

$$X = XP * SCALE + 5.0$$

and

$$Y = YP * SCALE + 5.0, \tag{10}$$

where XP and YP are the prime coordinates.

The ISITOK function now works with an elaborate set of parameters to avoid duplication and to assure operation within the allowable limits. If the answer from the function is YES, then the point is valid and can be used. The rest of the YES followup stores the plotting values in the KS array and prints out the names of the planes, the point number, and the point coordinates in terms of the X, Y, Z system, the prime system, and the plotter system. After exiting from these two loops, DLINE is called once more if IONCE is zero; otherwise, the plot is terminated and CROSEC makes its return.

10/22/69

```
      SUBROUTINE CROSEC                                                 CRS
      COMMON /TIMETEST/        KSETIME1, KSETIME2, KSETIME,  KSETADR    CRS
C                                                                       CRS
C...      COMMON AREA FOR CDC APT 3 SECTION 0                           CRS
C                                                                       CRS
C******         ******         ******         ******         ******    CRS
      COMMON /SYSTEMZ/         SYSTEM(4),KAPTCN,   KAPTTR,   KAPTIO,    CRS
     A     KAPTID,   KFLAGS(10),KO,      K1,       K2,       K3,        CRS
     B     K4,       K5,       K6,       K7,       K8,       K9,        CRS
     C     IFILL1,   IFILL2,   IFILL3,   IFILL4,                        CRS
     D     KFLAGO,   KFLAG1,   KFLAG2,   KFLAG3,   KFLAG4,              CRS
     E     IFILL5,   IWAVEN,   IPTNLY,   NOPOST,   IFILL6,   KAUTOP,    CRS
     F     ICLPRT,   INDEXX,   IPLOTR,   IFILL7,   NOPLOT,   KDYNFG,    CRS
     G     LOCJPT,   LOCBEG,   KSECIN,   NCLREC,   LOCMAC,   KPOCKET,   CRS
     H     IFILL8,   IFILL9,   IFILL10,  IPOSTP(1),NUMPST,   IPOSTFL(18),CRS
     I     TAPETB(1),CANTAP,   CLTAPE,   POCTAP,   PLOTAP,   SRFTAP,    CRS
     J     LIBTAP,   CRDTAP,   IFILL11,  CORTAP,   TAPES1,   TAPES2,    CRS
     K     TAPES3,   TAPES4,   FORTIN,   INTAPE,   IOUTAP,   PUNTAP,    CRS
     L     LSTFLG,   LTVFLG,   KONVTCL,  KINTRUPT,                      CRS
     M     PI,       PIO2,     DGTRD,    RDTDG,    ONE,                 CRS
     N     EXTRAO(20)                                                   CRS
      EQUIVALENCE   (PROTAP,   TAPETB)                                  CRS
C******         ******         ******         ******         ******    CRS
C                                                                       CRS
C...      COMMON AREA FOR CDC APT 3 SECTION 1                           CRS
C                                                                       CRS
C******         ******         ******         ******         ******    CRS
      COMMON /SECT1LOG/        ITAB1,    ITAB2,    ITAB3,               CRS
     A     ITAB4,    ITAB5,    ISNAM,    ITAB11,   ITAB12,   ITAB13,    CRS
     B     JENDPTPP, JENDCAN,  JENDSTOR, JSTRTCAN, JENDSYM,             CRS
     C     JCANTEMP, JRPTAB,   JLPTAB,   MAXNST,                        CRS
     D     JINWD,    JCHAR,    IWDERR,   JBUFL,    NUPERP,   NUPUN,     CRS
     E     JSTYPE,   JVARS2,   SCHERR,   NMACV,    MACASN(25),          CRS
     F     INDXPT,   IPTP,     IXPT,     MODE,     EOCFLG,   LPNDFL,    CRS
     G     TRMFLG,   INTRUPT,  JUMPFL,   ICDERR,   DEBUG,               CRS
     H     MACMODE,  NESTFL,   NRESULT,  IPTLIM,   JEXEC,               CRS
     I     KTYPE,    MACTYP,                                            CRS
     J     IPARTERR, FINIS,    IOFLG,    MACDEL,   JSUBER,   NUMBERR,   CRS
     K     DEFSTO(85),    DEFTAB(1000),  ZSUR(30),                      CRS
     L     XMAT4(16),XMAT3(16),XMAT2(16),XMAT1(16),TMATX(16),           CRS
     M     ISTDMODE, ISTDLIT,  ISTDTBL,  ISTDINDX, ISTDTYPE, ISTDWD,    CRS
     N     JPTIND,   KPTCODE,  KPTNAME,  KPTTYPE,  KPTNUM,   KPTINDX,   CRS
     O     KPTSUB,   KOMFLG,   KOMPOP,   NOSURS,   KANFLG,   KRFSYS,    CRS
     P     KANREC,   KANCNT,   INAME,    KANSURF,  KANINDX,             CRS
     Q     JPRELEN,  NEWCARD,  JGORIT,   NUMSTID,  NUMCSEQ,             CRS
     R     IRECIX,   IRECNO,   JTLPOS,   ITITLE(9),LSRECN,              CRS
     S     NNODEFX,  NNODEFI,  NIDUM,    ISLASH,   IEQUAL,   IBLANK,    CRS
     T     IDUMMY,   N10000,   N7777,    MASKU,    MASKL,    IDIV,      CRS
     U     MACREL,   MACLOC,   MACBEGN,  MACLAST,  MACLEVEL,            CRS
     V     MACNAME(3), MACINDX(3), NMV,  JRESTOR,  MACPSH(3,25),        CRS
     W     JTEMP1,   JTEMP2,   JTEMP3,   JTEMP4,   JTEMP5,   JTEMP6,    CRS
     X     JTEMP7,   JTEMP8,   JTEMP9,   EXTRA1(20)                     CRS
      EQUIVALENCE (DEFANS(1),IDEFSTO(4),DEFSTO(4)),(LSTYPE, KTYPE),     CRS
     1     (PTNUM, KPTNUM)                                              CRS
      DIMENSION IDEFSTO(85),   DEFANS(26),   IDEFTAB(1000)   ,          CRS
     1     ILPTAB(200), IRPTAB(200), ITNTAB(200), JPROTP(100),          CRS
```

10/22/69

```
     2    IRECSV(200), MACVAR(25),  MACNOR(25),  INWORD(10), IBUUP(2),
     3    PISTO(6), IPISTO(6), IDREC(4)
      EQUIVALENCE (INWORD(14), IBUUP(16), ILPTAB(990), IRPTAB(790),
     1    ITNTAB(290), PISTO(390), IPISTO(390), IDREC(384),
     2    JPROTP(380), IRECSV(280),  MACVAR(75), MACNOR(50),
     3    IDEFTAB(1000), DEFTAB(1000))
C
      COMMON/VOCABTBL/ KOM(1100)
C
      COMMON /2/ JTABNUM,JTABL(12120)
C******          ******          ******          *******          ****
C
C
C      *      *      *      *      *      *      *
C      *      *    C O M M E N T     *      *      *
C      *      *      *      *      *      *      *
C      *      *      *      *      *      *      *
C
C     CROSEC (MOD 1) PROVIDES A MEANS OF EXTENDING THE USE OF THE PLANE
C     SURFACES(DEFINED BY THE PART PROGRAMMER IN THE PART PROGRAM) BY
C     PROVIDING A PLOTTING CAPABILITY IN WHICH THE LINES OF
C     INTERSECTION, WITHIN SPECIFIED LIMITS, BETWEEN A CROSS-SECTIONAL
C     PLANE AND ALL OTHER DEFINED PLANES ARE SHOWN.
C
C     THE CROSS SECTIONAL PLANE FOR THE PLOT AND ITS DIMENSIONAL LIMITS
C     ARE CONTROLLED BY MEANS OF ONE PLANE DEFINITION (NAMED ≠≠HOPE≠≠)
C     AND TWO POINT DEFINITIONS (NAMED ≠≠HOXYMIN≠≠ AND ≠≠HOXYMAX≠≠)
C     ADDED TO THE PART PROGRAM.
C
C     THE PLOT IS SUPPLEMENTED BY PRINTER OUTPUT THAT IDENTIFIES POINTS
C     OF INTERSECTION THAT HAVE BEEN NUMBERED ON THE PLOT.
C
C     THE PROGRAM IS WRITTEN IN THE FRAMEWORK OF THE CDC 3800 APT 2.1
C     CONFIGURATION AND NO ADDITIONAL OVERLAY OR SEGMENT MANIPULATION
C     IS NECESSARY.
C
C     CROSEC IS CALLED FROM SUBROUTINE FINI, AND THIS CALL IS THE ONLY
C     MODIFICATION TO APT 2.1 PROGRAMMING BEYOND THE ADDITION OF CROSEC
C     AND ITS FAMILY OF SUBROUTINES.
C
C
C      *      *      *      *      *      *      *
C
C     CROSEC DIMENSIONING BEGINS HERE
C
C     ≠HOPE≠ IS THE NAME GIVEN TO THE PLANE OF THE CROSS SECTION
C
C      NOTE--THE HOPE SYSTEM IS CALLED THE PRIME SYSTEM, THE TERMS ARE
C     USED INTERCHANGABLY AND IS DENOTED BY THE SYMBOL≠.
C     THIS SYMBOL IS USUALLY USED AS A SUFFIX, SUCH AS X≠  .
C
C
C     A,B,C,D, ARE THE HOPE PLANE CONSTANTS
C     OBTAINED FROM THE CANONICAL FORM
C
C     ISEND IS THE LENGTH OF THE DEFINED SYMBOL TABLE
```

10/22/69

```
C
C
C     THE XMAT9 ARRAY WILL HØLD THE MATRIX ØF CØEFFICIENTS FØR
C     CØØRDINATE  CØNVERSIØN (TRANSLATIØN AND/ØR RØTATIØN)
C     FRØM THE MAJØR SYSTEM TØ THE HØPE SYSTEM
C
C     FRØM THE EQUATIØNS
C
C      X≠ = T11(X-X0) + T21(Y-Y0) + T31(Z-Z0)
C
C      Y≠ = T12(X-X0) + T22(Y-Y0) + T32(Z-Z0)
C
C      Z≠ = T13(X-X0) + T32(Y-Y0) + T33(Z-Z0)
C
C
C     WHERE-
C
C         THE X≠ AXIS HAS DIRECTIØN CØSINES- T11, T21, T31
C
C         THE Y≠ AXIS HAS DIRECTIØN CØSINES- T12, T22, T32
C
C         THE Z≠ AXIS HAS DIRECTIØN CØSINES- T13, T23, T33
C
C         THE PRIME SYSTEM ØRIGIN IS AT (X0,Y0,Z0)
C
C     IT IS FØRMED AS FØLLØWS-
C
C            I                                                I
C            I                                                I
C            I                                                I
C            I      T11        T21        T31       C1        I
C            I                                                I
C            I      T12        T22        T32       C2        I
C            I                                                I
C            I      T13        T23        T33       C3        I
C            I                                                I
C            I                                                I
C            I
C
C     WHERE-
C         C1 = -(T11*X0   +   T21*Y0   +   T31*Z0)
C
C         C2 = -(T12*X0   +   T22*Y0   +   T32*Z0)
C
C         C3 = -(T13*X0   +   T23*Y0   +   T33*Z0)
C
C
C
C
      CØMMØN/12/A,B,C,D,IDSEND,XMAT9(16)
C
C
C     THE KR ARRAY HØLDS THE (X≠Y≠) INFØRMATIØN ØF ALL PØINTS ØBTAINED
C     ØN ØNE CØMPLETE PASS THRØUGH THE INNER LØØP CØMMENCING AT
C     STATEMENT 1004.
C
```

UNCLASSIFIED

10/22/69

```
C     ≠≠IP≠≠ COUNTS THE NUMBER OF PAIRS                                 CRS 1690
C                                                                       CRS 1700
       COMMON/4/IP,KR(100)                                              CRS 1710
       REAL KR                                                          CRS 1720
C                                                                       CRS 1730
C                                                                       CRS 1740
C     THE KS ARRAY HOLDS THE (X≠Y≠) INFORMATION OF ALL POINTS OBTAINED  CRS 1750
C     BY PASSES THROUGH BOTH LOOPS, I.E. THE OUTER LOOP COMMENCING AT   CRS 1760
C     STATEMENT 1001 AND THE INNER LOOP THAT STARTS AT 1004,            CRS 1770
C     ≠≠IS≠≠ COUNTS THE NUMBER OF PAIRS                                 CRS 1780
C                                                                       CRS 1790
       COMMON/5/IS,KS(500)                                              CRS 1800
       REAL KS                                                          CRS 1810
C                                                                       CRS 1820
C     ≠≠IONCE≠≠ IS A FLAG THAT CONTROLS THE CALLING OF SUBROUTINE DLINE CRS 1830
C                                                                       CRS 1840
       COMMON/6/IONCE                                                   CRS 1850
C                                                                       CRS 1860
C     ≠≠ARRAY≠≠ IS FOR THE FORMATION OF THE PLOT STRING,                CRS 1870
C                                                                       CRS 1880
       DIMENSION ARRAY (254)                                            CRS 1890
C                                                                       CRS 1900
C     THE ≠≠IPOISET≠≠ ARRAY IS USED AS A PARAMETER IN THE ENCODE COMMANDCRS 1910
C     TO COMPOSE BCD FOR THE PLOT.                                      CRS 1920
C                                                                       CRS 1930
       DIMENSION IPOISET(9)                                             CRS 1940
C                                                                       CRS 1950
C                                                                       CRS 1960
C        *         *         *         *         *         *         *  CRS 1970
C     THE PROGRAMMING ACTION STARTS HERE                                CRS 1980
C                                                                       CRS 1990
C     ≠≠YES≠≠ AND ≠≠NO≠≠ ARE THE TWO POSSIBLE ANSWERS FOR               CRS 2000
C     FUNCTION ISITOK,                                                  CRS 2010
C                                                                       CRS 2020
C                                                                       CRS 2030
       YES=1                                                            CRS 2040
       NO=0                                                             CRS 2050
C                                                                       CRS 2060
C                                                                       CRS 2070
C                                                                       CRS 2080
C                                                                       CRS 2090
C     A TOP OF FORM ACTION TO SEPARATE CROSEC OUTPUT FROM EARLIER       CRS 2100
C     APT OUTPUT,                                                       CRS 2110
C                                                                       CRS 2120
       PRINT 40                                                         CRS 2130
    40 FORMAT (1H1)                                                     CRS 2140
C                                                                       CRS 2150
C                                                                       CRS 2160
C LOCATING THE NAME HOPE IN THE DEFINED SYMBOL TABLE                    CRS 2170
C DEFINED SYMBOL TABLE BEGINS AT ITAB11 AND ENDS AT ITAB12              CRS 2180
C     ITS TOTAL LENGTH IS THEREFORE (ITAB12-ITAB11)+1                   CRS 2190
C                                                                       CRS 2200
C  ≠HOPE≠ IS THE NAME GIVEN TO THE REFERENCE PLANE                      CRS 2210
C                                                                       CRS 2220
C                                                                       CRS 2230
       IDSEND=ITAB12-ITAB11                                             CRS 2240
```

10/22/69

```
      IDSEND=IDSEND+1
      DO 900 I=1,IDSEND,2
      ISAVE=I
      I=I-1
C     INFORMATION IN THE DST APPEARS IN PAIRS AND
C     THE FIRST WORD OF THE FIRST PAIR IS AT LOCATION ≠≠ITAB11≠≠ IN
C     ARRAY ≠≠JTABL≠≠
C     THE FIRST WORDS OF THESE PAIRS ARE LEFT-JUSTIFIED BCD NAMES
C     TO A MAXIMUM OF SIX LETTERS AND BLANK FILLED
C
C
C                                  H O P E
      IF (JTABL(ITAB11+I),EQ,30464725606060608)  800,900
C
C     ≠≠PLOTPLNO≠≠ IS THE INCREMENTAL INDEX TO THE SECOND WORD IN THE
C     ≠≠HOPE≠≠ PAIR,
C
  800 PLOTPLNO=I+1
      GO TO 600
  900  I=ISAVE
C
C     IF NO HOPE CARD HAS BEEN IDENTIFIED RETURN TO SOURCE OF CALL
C
  700  PRINT 701
  701  FORMAT (1X,*NO HOPE CARD*)
       GO TO 1401
C        *         *         *         *         *         *         *
C        *         *        C O M M E N T        *         *         *
C        *         *         *         *         *         *         *
C        *         *         *         *         *         *         *
C     DEFINING THE CANONICAL FORM OF THE PLANE AS
C         X * COS ALPHA  + Y * COS BETA + Z * COS GAMMA  = P
C     WHERE THE COSINE TERMS ARE DIRECTION COSINES OF X, Y, AND Z
C     RESPECTFULLY,
C     THE CANONICAL FORM OF THE ≠HOPE≠ PLANE IS CALLED AND STORED IN
C     THE REFERENCE ARRAY, WHERE
C            ≠RCOSA≠ CONTAINS COSINE ALPHA
C            ≠RCOSB≠ CONTAINS COSINE BETA
C            ≠RCOSG≠ CONTAINS COSINE GAMMA ,AND
C            ≠R HOPE P≠ CONTAINS THE ≠P≠ OR CONSTANT VALUE
C        *         *         *         *         *         *         *
C        *         *         *         *         *         *         *
C
C     ≠≠KANSURF≠≠ IS DESIGNED TO HOLD POINTERS TO CANONICAL FORMS
C     IN STATEMENT 600 IT IS HOLDING THE POINTER TO THE ≠≠HOPE≠≠PLANE
C     CANONICAL FORM
C
  600  KANSURF = JTABL (ITAB11 + PLOTPLNO)
C
C     THE ≠≠CANGET≠≠ SUBROUTINE FETCHES THE CANONICAL FORM POINTED TO
C     BY ≠≠KANSURF≠≠ AND STORES ITS ELEMENTS IN THE ≠≠DEFSTO≠≠ ARRAY,
C
       CALL CANGET
       RCOSA = DEFSTO(4)
       RCOSB=DEFSTO(5)
       RCOSG = DEFSTO(6)
```

NRL Report 7025

Cross-Sectional Plots of Plane Intersections

An Adaptation of the APT System

KINGSLEY P. THOMPSON

Information Systems Branch
Mathematical and Information Sciences Division

January 27, 1970

NAVAL RESEARCH LABORATORY
Washington, D.C.

CONTENTS

ABSTRACT

The APT (Automatically Programmed Tooling) system provides for the software description of geometric forms, the delineation of a tool path, and, for output, discrete positioning information on punched paper tape for use by a numerically controlled machine. The CROSEC (Mod 1) program, described in this report, provides a means of extending the use of the canonical forms of the plane surfaces defined by the programmer in the part program by providing a plotting capability in which the lines in intersection, within specified limits, between a cross-sectional plane and all other defined planes are shown. A visualization of the initial plane framework on which the cutting is to be performed is thereby provided. The cross sectional plane and its dimensional limits are controlled by one plane definition and two point definitions. The plot is supplemented by printer output that aids in the interpretation of the plot. The program is written in the framework of the CDC 3800 APT 2.1 configuration. No additional program overlays or segments are necessary. This report contains a discussion of the method used, subroutine descriptions, listings and flowcharts, implementation aids, and a sample run.

PROBLEM STATUS

This is an interim report on a continuing problem.

AUTHORIZATION

NRL Problem 23Z0001

Manuscript submitted November 17, 1969.

CROSS-SECTIONAL PLOTS OF PLANE INTERSECTIONS

An Adaptation of the APT System

INTRODUCTION

This report contains a complete description of a plotting adaptation of the APT system in use at the Naval Research Laboratory (APT 2.1 on the CDC 3800). In the APT language, a plane can be defined in seven different ways; but after being processed they are all stored in a standard "canonical form." This program, called CROSEC (Mod 1), utilizes such plane canonical forms (of an APT Part Program) to obtain a plot of the intersection of one of the planes, designated as the cross-sectional plane, and all of the other planes that have been defined, processed, and stored at the end of APT Section 1. (See Ref. 1 for a detailed description of the APT system.)

The purpose of the program, primarily, is as an aid, an extra tool for the programmer in debugging his program. The defined surfaces as stored in their canonical forms provide a convenient starting point for geometric considerations. It is assumed that syntactic errors have already been discovered and corrected and that in using CROSEC the programmer wishes to verify that the surfaces he has defined do indeed describe the piece he wishes to have worked on by the tool. The hope is that verification can be accomplished easily if he can get a look at any cross section of his choosing through the conglomerate of the starting surfaces. He realizes that the plot might require some interpretation because defined surfaces intersecting together do not fully describe the finished piece. However, he accepts this limitation and looks upon the output as a working drawing, a picture of the output of Section 1. By means of this drawing and the accompanying identifying information from the printer, he should be able to make some significant debugging progress. Perhaps he will discover a section of surface that is defined improperly or over defined, or a combination of surfaces that could be redefined in a simpler manner. Also, he may discover a portion of surface that he has not yet defined and other events of this nature.

The report discusses plane equations, outlines the method of obtaining a coordinate system in the cross-sectional plane, describes the plot, and points out the limitations of the program. Also included are flowcharts, program descriptions, a complete listing, and details of implementation with APT 2.1. An example is introduced early and followed through the complete process in full detail.

DISCUSSION

The Plane

The *canonical* form for the plane, as defined in the APT system, is given by

$$AX + BY + CZ = D, \tag{1}$$

where*

*As quoted from pp. 80 and A-1 of Ref. 1.

A = X-Component of unit normal vector to plane
B = Y-Component of unit normal vector to plane
C = Z-Component of unit normal vector to plane
D = Normal distance from plane to origin.

Consider the *normal* form of the plane equation, namely

$$XCOS\alpha_x + YCOS\alpha_y + ZCOS\alpha_z = L, \tag{2}$$

where the coefficients for the X, Y, and Z coordinates in Eq. (2) are the direction cosines of the positive normal vector N from the origin to the plane. The direction cosines of the unit normal vector are identical with the direction cosines of N. The length, or absolute value, of this normal is L. Therefore, equating similar terms from Eqs. (1) and (2) yields

$$A = COS\alpha_x$$

$$B = COS\alpha_y$$

$$C = COS\alpha_z$$

$$D = L = |N| .$$

Thus, for example, A = 0, B = 1, C = 0, D = 1 is the plane passing through the point (0, 1, 0), parallel to the XZ coordinate plane, with direction angles of 90, 0, and 90 degrees, respectively, to the three axes.

Let us now take, for a more detailed example, the plane that passes through the points (1,0,0), (0,1,0), and (0,0,1) a unit distance out along each axis. To fit this approach to the definition of a plane, consider the *intercept* form for a plane equation,*

$$\frac{X}{X_1} + \frac{Y}{Y_2} + \frac{Z}{Z_3} = 1, \tag{3}$$

where X_1, Y_2, and Z_3 are the intercepts, i.e. the point (X_1, 0, 0) is the intersection of the X axis with the plane. Similarly with (0, Y_2, 0) and (0, 0, Z_3).

Using now the three unit axis points, already defined, in this intercept form, Eq. (3) leads to the simple and interesting equation

$$X + Y + Z = 1. \tag{4}$$

The correctness of Eq. (4) as truly representing the plane that passes through the three points is easily determined by setting any two of the variables equal to zero, and the remaining variable will be equal to 1. Equation (4) is illustrative of another form of an equation used to describe a plane, the *general* form, where the coefficients of the X, Y, and Z terms are considered to be *direction numbers* of the positive normal to the plane.

To go from the general form to the normal form, it is necessary to compute the direction cosines by dividing each of the coordinate coefficients in turn by the square root of the sum of the squares of all three coefficients. The length of the normal is obtained in a similar fashion by dividing the constant term by the same square-root quantity. Symbolically, if the general form is

$$PX + QY + RZ = S, \tag{5}$$

*All forms of equations for a plane can be found in Section 3, p. 2-1 of Ref. 2.

then

$$A = \mathrm{COS}\alpha_x = \frac{P}{U}$$

$$B = \mathrm{COS}\alpha_y = \frac{Q}{U}$$

$$C = \mathrm{COS}\alpha_z = \frac{R}{U}$$

$$D = L = \frac{S}{U},$$

where

$$U = \sqrt{P*P + Q*Q + R*R.}$$

The normal form for the illustrative plane is therefore

$$\frac{X}{\sqrt{3}} + \frac{Y}{\sqrt{3}} + \frac{Z}{\sqrt{3}} = \frac{1}{\sqrt{3}}. \tag{6}$$

It is easily verified that the sum of the squares of the direction cosines is 1.

It is important in our development to know the coordinates of the point represented by the intersection of the normal N and the plane to which it is perpendicular. These coordinates are obtained by multiplying each of the direction cosines by the length of the normal. Symbolically, (D*A, D*B, D*C). For the illustrative plane the result is (1/3, 1/3, 1/3).

One final consideration regarding these plane equations—our illustrative case has a convenient set of intercept points; however, it is possible to determine the intercepts from the normal form. They are

$$(X_1, Y_1, Z_1) = \left(\frac{D}{A}, 0, 0\right)$$

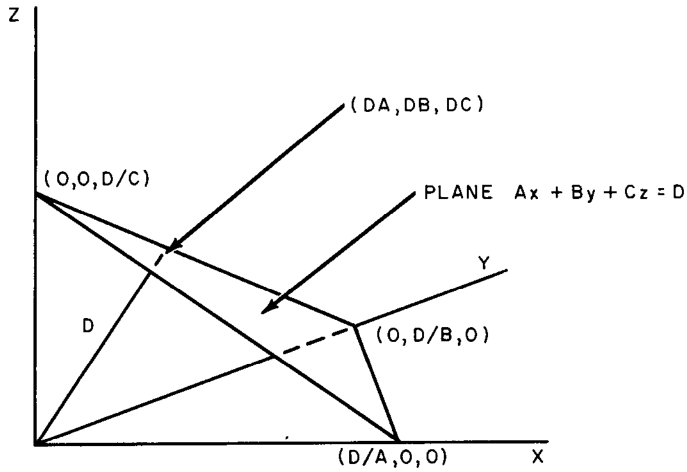
$$(X_2, Y_2, Z_2) = \left(0, \frac{D}{B}, 0\right)$$

$$(X_3, Y_3, Z_3) = \left(0, 0, \frac{D}{C}\right).$$

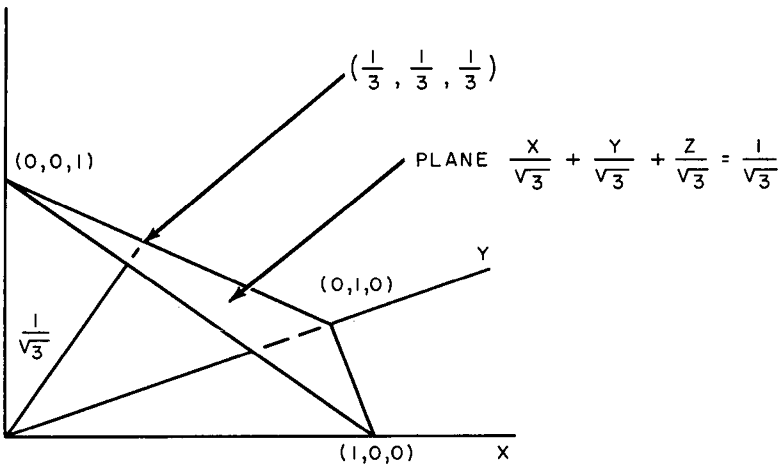
Figure 1 summarizes this initial development using the symbols for the APT canonical form of Eq. (1) in part a of the figure and the actual values of our illustrative plane in part b.

The Cross-Sectional Plane and Its Coordinate System

The plot of Section 1 output uses a coordinate system in the cross-sectional plane, particularized as the HOPE plane. Let us call the coordinate system in the HOPE plane the prime system (i.e., X′, Y′, Z′), in contrast to the original system established by the part programmer known as the X, Y, Z system. The origin of the prime system is the intersection of the normal with the plane, and the Z′ axis is the extension of the normal.



(a) Normal form of plane equation



(b) Equiangular plane in normal form

Fig. 1 - A general and a particular plane

The X′ axis is selected in one of three ways.

1. The "general solution" is the case where the HOPE plane intersects the X axis, and the positive X′ axis is the line passing from the prime origin through the intercept point and lies in the HOPE plane. In terms of the *normal* form the prime origin is (D*A, D*B, D*C) and the intercept point is (D/A, 0, 0) with direction cosines

$$\left[\frac{\left(\frac{D}{A} - D*A\right)}{U}, \frac{-D*B}{U}, \frac{-D*C}{U}\right],$$

where

$$U = \sqrt{\left(\frac{D}{A} - D*A\right)^2 + (D*B)^2 + (D*C)^2}.$$

For our illustrative plane X + Y + Z = 1, the X′ axis for the general solution would be positively directed from the origin with coordinates (1/3, 1/3, 1/3) to the X intercept with coordinates (1, 0, 0). The computation gives the X′ axis, in this case, direction cosines of

$$(\sqrt{2/3},\ -\sqrt{1/6},\ -\sqrt{1/6})$$

2. If the HOPE plane does not intersect the X axis and is parallel to it, then the X′ axis is that line lying in the HOPE plane, parallel to the X axis, with direction cosines (1, 0, 0) commencing at the prime origin.

3. If the HOPE plane is perpendicular to the X axis and parallel to the YZ plane such that the normal is the X axis, then the positive X′ axis is that line lying in the HOPE plane which starts at the prime origin and is parallel to the Z axis with direction cosines (0, 0, 1).

The Y′ axis is defined as the cross product of the Z′ and X′ axes, duly preserving right-handed concepts.

If the direction cosines of the X′ axis are T_{11}, T_{21}, T_{31}, and those of the Y′ axis T_{12}, T_{22}, T_{32} and those of the Z′ axis T_{13}, T_{23}, T_{33}, then

$$T_{12} = T_{23}*T_{31} - T_{21}*T_{33} \tag{7}$$

$$T_{22} = T_{11}*T_{33} - T_{13}*T_{31} \tag{8}$$

$$T_{32} = T_{13}*T_{21} - T_{11}*T_{23}. \tag{9}$$

Thus, for our illustrative plane, the direction cosines of Z′ are $(1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$ and of X′ are $(\sqrt{2/3}, -\sqrt{1/6}, -\sqrt{1/6})$. Using these equations, we find that the directions cosines for Y′ are $(0, \sqrt{1/2}, -\sqrt{1/2})$.

The equations required to convert any point in space from the original coordinates to the prime coordinates are

$$X' = T_{11}\ (X - X_0) + T_{21}\ (Y - Y_0) + T_{31}\ (Z - Z_0),$$

$$Y' = T_{12}\ (X - X_0) + T_{22}\ (Y - Y_0) + T_{32}\ (Z - Z_0),$$

and

$$Z' = T_{13}\ (X - X_0) + T_{23}\ (Y - Y_0) + T_{33}\ (Z - Z_0),$$

where (X_0, Y_0, Z_0) is the prime origin defined in terms of X, Y, Z. Expanding yields

$$X' = T_{11}*X + T_{21}*Y + T_{31}*Z + C_1, \tag{10}$$

$$Y' = T_{12}*X + T_{22}*Y + T_{32}*Z + C_2, \tag{11}$$

and

$$Z' = T_{13}*X + T_{23}*Y + T_{33}*Z + C_3, \tag{12}$$

where

$$C_1 = -(T_{11}*X_0 + T_{21}*Y_0 + T_{31}*Z_0),$$

$$C_2 = -(T_{12}*X_0 + T_{22}*Y_0 + T_{32}*Z_0),$$

$$C_3 = -(T_{13}*X_0 + T_{23}*Y_0 + T_{33}*Z_0).$$

For the illustrative case where $(X_0, Y_0, Z_0) = (D*A, D*B, D*C) = (1/3, 1/3, 1/3)$, the matrix corresponding to the coefficients of Eqs. (10), (11), and (12), is

$$\begin{bmatrix} \sqrt{\frac{2}{3}} & -\sqrt{\frac{1}{6}} & -\sqrt{\frac{1}{6}} & 0 \\ 0 & \sqrt{\frac{1}{2}} & -\sqrt{\frac{1}{2}} & 0 \\ \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} & -\sqrt{\frac{1}{3}} \end{bmatrix}.$$

Figure 2 illustrates the definition of the prime axes in the HOPE plane. In this instance the HOPE plane is our illustrative plane X + Y + Z = 1.

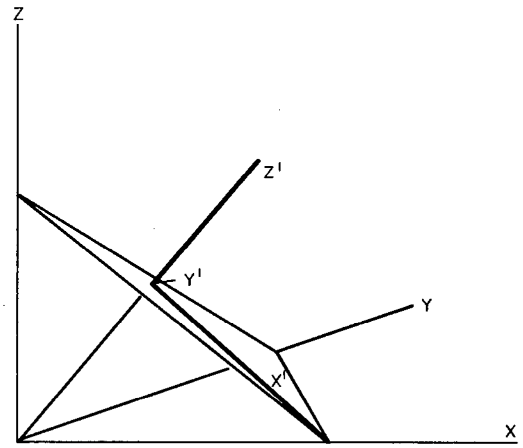


Fig. 2 - Prime axes in the equiangular HOPE plane

Points and Lines in the Cross-Sectional Plane

So far we have established an understanding of the various forms of the equations that describe a plane and have described an algorithm for defining a translation-rotation matrix for converting points from the part programmer's coordinate system to a coordinate system in the HOPE or cross-sectional plane. We can now ask, How is it determined whether any APT-defined point in general lies in the HOPE plane? Simply by substituting the X, Y, and Z values of the point into the equation for the plane to determine if an equality exists; e.g., Does the point (1/3, 1/3, 1/3) lie in the plane X + Y + Z = 1? Obviously yes, since 1/3 + 1/3 + 1/3 = 1.

If the equations for the three planes are solved simultaneously, a point that lies in all three planes is the result. If one of these planes is the HOPE plane, then there is no doubt that the point is in the cross section.

Consider now a situation in which the HOPE plane and a plane A are consecutively solved with two other planes B and C (defined such that no two planes of HOPE, A, B and HOPE, A, C are parallel), resulting in two points P_1 and P_2. Both of these points are simultaneously in both the HOPE plane and in plane A; in fact, the line segment joining P_1 and P_2 is a portion of the line of intersection between HOPE and A. Such a procedure if carried out with all the defined planes, will result in an entire network of lines of inter section. Each point as obtained is put through the matrix to obtain its definition in terms of the HOPE coordinate system to enable it to be plotted. If a minimum and a maximum value for each coordinate is specified, then many superfluous points of intersection can be eliminated.

Let us now consider what happens when the equiangular plane X + Y + Z = 1 intersects some planes defined in a part program, an actual situation. (Refer to sample run with "PARTNO TESTING" on p. 63.) Figure 3 depicts a simple part in three views. It is to be noted that there are eight defined planes. Figure 4 shows the cross-section network of intersecting lines obtained with the plane X + Y + Z = 1 as the HOPE plane intersecting "PARTNO TESTING." Using only the information provided in Figs. 3 and 4, can you distinguish between the proper intersection outline and those lines that are extraneous? It is an intersecting exercise, well worth spending a few minutes on.

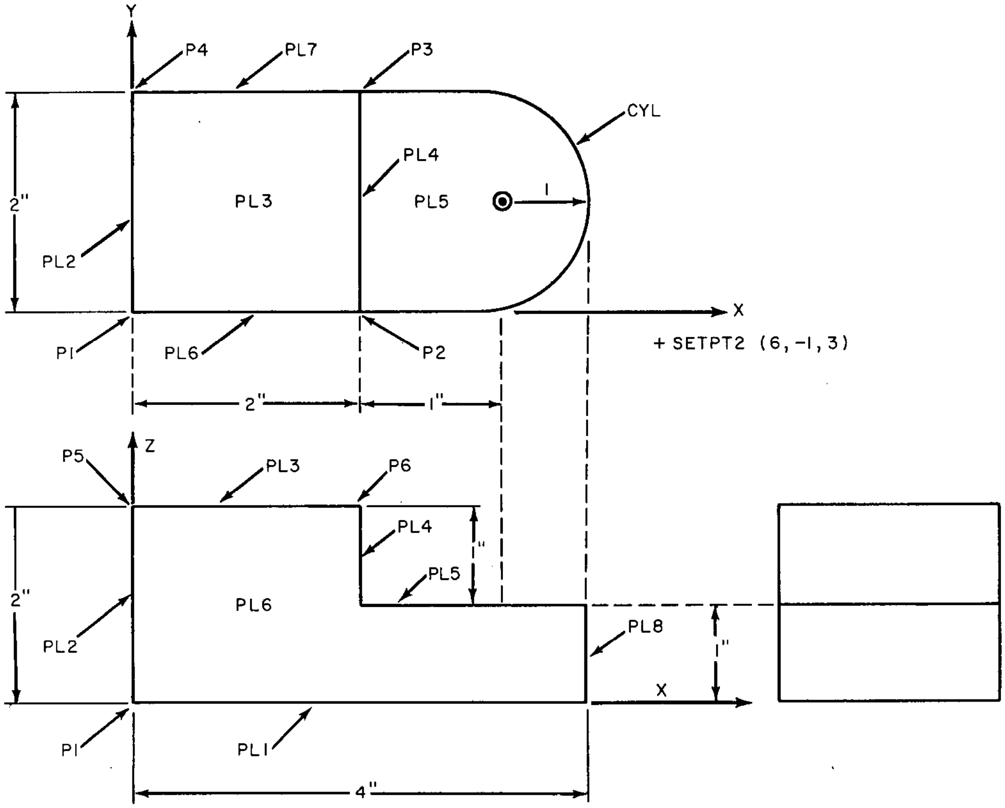


Fig. 3 - "PARTNO TESTING" with eight defined planes

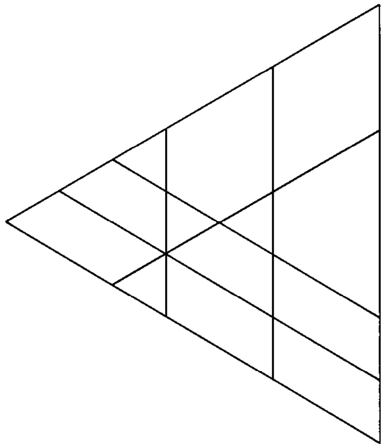
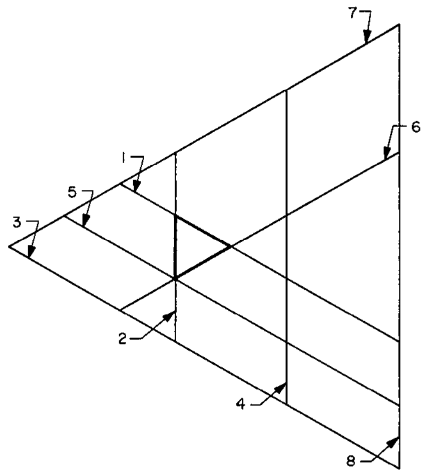


Fig. 4 - Cross section obtained from PARTNO TESTING and equiangular plane

Fig. 5 - Lines of intersection identified by plane number



The solution to this question will now be presented.

The accompanying printout identifies points of intersection from which the lines of intersection were drawn. The coefficients of these points are given in the X, Y, Z system, in the X′, Y′, Z′ system, and in plotter coordinates along with the plane numbers associated with each point. There are 19 such points in Fig. 4. These point data constitute background information, available when needed. However, rather than clutter up the plot with point symbols and identifying numbers, only the lines of intersection have been drawn using a minimum point and a maximum point. From these two points the slope and the prime-system intercept have been computed. This information is also printed along with the plane name associated with the line. The lines have been named in Fig. 5. There are eight planes identified in the sketch of the part in Fig. 3 and there are eight lines in the intersection plot.

Verification of the correctness of the numbering of these lines of intersection can be achieved by noting that parallel *lines* in the plot correspond to parallel *planes* on the part. Sure enough, plane 1 representing the bottom of the piece is parallel to plane 3, the top, and is also parallel to plane 5 which represents the step surface. In Fig. 5 the lines labeled 1, 5, and 3 are parallel lines.

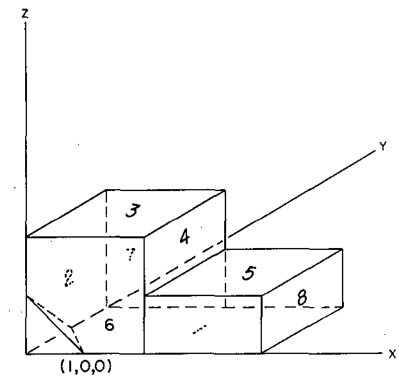


Fig. 6 - Perspective view of PARTNO TESTING (without the cylinder) with eight planes and the intersecting equiangular HOPE plane

In a similar fashion the lines of intersection for planes 2, 4, and 8 confirm a parallelism between these planes, all three of them being vertical sides in the part shown in Fig. 3. Finally, plane 6 (the front) is parallel to plane 7 (back), and the lines with these numbers are parallel in Fig. 5.

Figure 6, which shows a perspective view of the planes of the part and the equiangular HOPE plane, demonstrates the fact that the lines of intersection on the surface of the piece lie in planes 6 (front), 1 (bottom), and 2 (left side). The triangle formed by the lines of intersection of these three planes is accented in Fig. 5 and is the correct answer.

It is an interesting consequence of the situation that since plane X + Y + Z = 1 is equiangular to each of the three coordinate planes, and that since all eight of the defined planes in the simple part are either identical with or parallel to a coordinate plane, that HOPE is also equiangular to these eight planes, namely 57.4 degrees.

There is one other interesting observation to make about Fig. 5. Because of the equiangular characteristic of the cross-sectional plane, a reverse type of statement can be made. Any equilateral triangle in this figure corresponds to three mutually perpendicular planes. Besides the set already discussed {6, 1, 2}, eleven others can be identified. These are {1, 2, 7}, {1, 4, 6}, {1, 6, 8}, {2, 3, 6}, {3, 7, 8}, {1, 7, 8}, {5, 7, 8}, {2, 3, 7}, {3, 4, 7}, {3, 6, 8} and {3, 7, 8}. The last one, which is the outermost triangle, corresponds to the planes which are perpendicular to the maximum dimensions X = 4, Y = 2, and Z = 2, respectively.

There is a lot of information to be gleaned from the cross-sectional plot once it is understood and assimilated.

When the restrictions of XMIN, YMIN, ZMIN = (0, 0, 0) and XMAX, YMAX, ZMAX = (4, 2, 2) are placed on the cross-section points, then the network is reduced to that shown in Fig. 7, the true triangle of intersection with the part.

CROSEC Limitations

CROSEC in its present form (Mod 1) has some definite limitations. Some of the initially obvious ones are as follows:

1. There is no identification of defined points that might lie in the HOPE plane.

2. The plot does not contain any intersections of the HOPE plane with quadrix surfaces.

3. No tool motion information is present.

4. The plot resulting from the lines of intersection of defined planes with HOPE can be initially confusing, since some of the lines might have no immediate relation to the finished part, and the prime coordinate system might not be simply oriented with respect to the X, Y, Z system. However, a consideration of the original part program along with the printout can soon make the plot understandable.

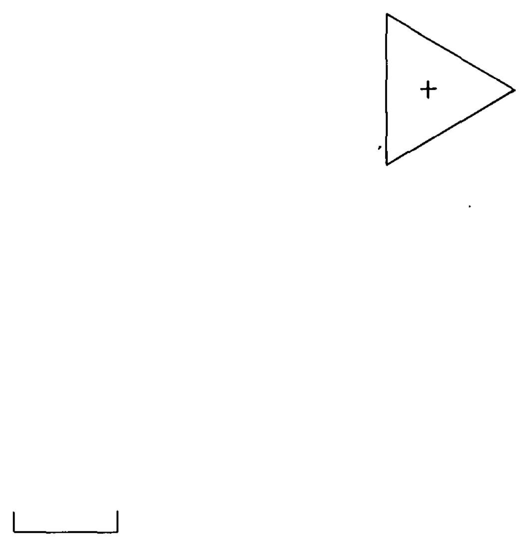


Fig. 7 - CROSEC (Mod 1) output plot for sample run with PARTNO TESTING

CONCLUSION

CROSEC (Mod 1) is a first step. In its present form, it can be helpful in a limited fashion. The intersection with curved surfaces is missing, as well as the path of the cutting tool in the cross-sectional plane. It is planned to incorporate these features in further work on this project.

ACKNOWLEDGMENT

The author expresses his appreciation to the Engineering Services Division for their cooperation in supplying information on the APT system in operation. In particular Lloyd Murphy, Nicholas Maddage, Donald Woods, and Jay Williams, Numerical Control Programming personnel of that Division, were particularly helpful. Also helpful, in more computer-related aspects, were Gary Flenner and Anna Byrd Mays of the NRL Research Computation Center, who worked on the initial implementation of APT at NRL.

REFERENCES

1. "APT Reference Manual for CDC 3400 and 3600 Computer Systems," Publication 60132800, Control Data Corporation, Palo Alto, Calif., Aug. 1965

2. Korn, G.A., and Korn, T.M., "Mathematical Handbook for Scientists and Engineers," New York:McGraw-Hill, 1961

Appendix A

SUBROUTINE DESCRIPTIONS WITH FLOWCHARTS AND LISTINGS

SUBROUTINE CROSEC

The CROSEC subroutine (Fig. A1) controls the Section 1 plotting of the intersections of defined planes with the HOPE cross-sectional plane. For convenience it incorporates all of the common area of Section 0 and Section 1 into itself without attempting to discard portions it does not need. The APT system subroutines called on are CANGET, SIMEQ and STDUNPK. The in-house library routines called include PLOTS, SYMBOL, PLOT, STOPPLOT, SQRTF, ACOSF, QNSINGL, THEND, STH, ENC, Q1Q10100, Q8QSTOPS, and Q8QDICT. Additional subroutines called by CROSEC and considered a part of the Mod 1 package are TESTHOPE, DELINE, and MM. There is also a function, ISITOK. The flowchart of Fig. A2 shows the relationship between these subroutines.

The CROSEC subroutine assumes that the defined symbol table (D.S.T.) is stored in the JTABL array of numbered common 2 between ITAB11 and ITAB12. It further assumes that the D.S.T. entries consist of pairs of words, the first of which is an eight-symbol Hollerith name, left justified, and, second, an APT "standard word" which includes an integer pointer giving the relative address in JTABL (extended beyond D.S.T.) where the canonical form is stored. These assumptions are standard procedure for the CDC APT. (See the Section 1 description starting on p. 01-1 of Ref. A1.) Accordingly, after executing a top of form, the D.S.T. is searched for the name HOPE, and if this name is not found the subroutine is exited and there is no plot obtained. If HOPE is found, its pointer is stored in KANSURF prior to calling CANGET which fetches and stores the canonical form in the DEFSTO array of the SECT1LOG. After identifying A, B, C, and D from the HOPE canonical form, the TESTHOPE subroutine is called.

TESTHOPE will either stop the run or return with a conversion matrix stored in XMAT9. This matrix will permit the conversion from the X, Y, Z coordinate system to the prime system whose origin is in the HOPE plane, thereby facilitating a two-dimensional plot.

Another top of form is executed.

Since the cosine of the angle between two planes is equal to the sum of the product of their corresponding direction cosines, it is a logical next step to take advantage of this fact and compute the angle between each defined plane and the HOPE plane.

Special care is needed in picking up canonical pointers from the D.S.T. prior to making the cosine computation. Undefined words or incorrectly defined words must be avoided as well as the synonym register which appears at the head of the D.S.T. The standard word must be unpacked by calling Subroutine STDUNPK in order to determine if BYTA contains a 4, representing a canonical form. After the canonical form has been recovered by a call to CANGET, it is necessary to examine the four most right-hand bits of the first word in the set. A 3 in this position identifies the canonical of a plane as opposed to other possible canonical forms such as points, cylinders, etc. In this manner the direction cosines are obtained, and the computation for the angle between the planes can then be performed.

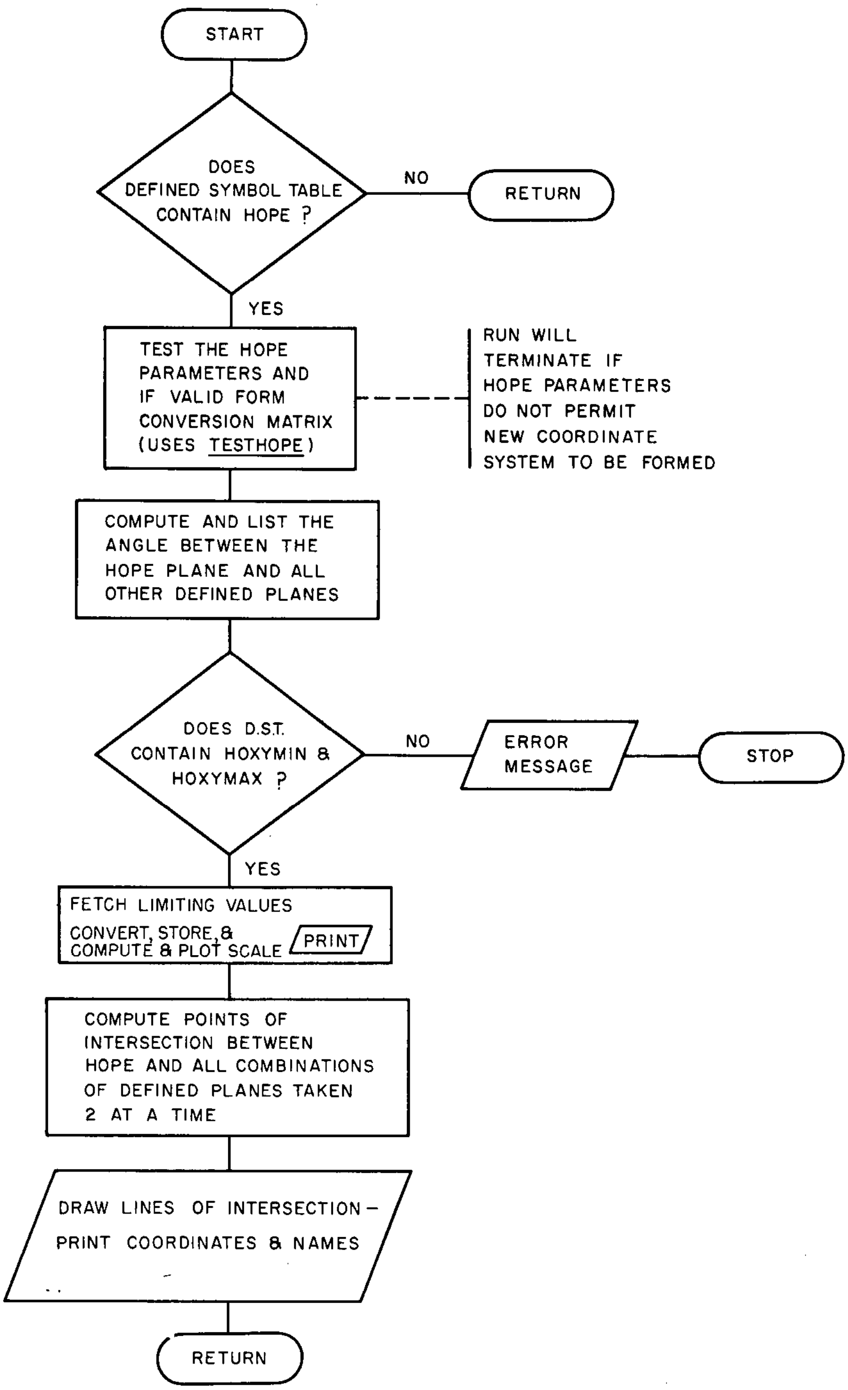


Fig. A1 - Flowchart for CROSEC subroutine

NRL Report 7025

Cross-Sectional Plots of Plane Intersections

An Adaptation of the APT System

KINGSLEY P. THOMPSON

Information Systems Branch
Mathematical and Information Sciences Division

January 27, 1970

NAVAL RESEARCH LABORATORY
Washington, D.C.

CONTENTS

ABSTRACT

The APT (Automatically Programmed Tooling) system provides for the software description of geometric forms, the delineation of a tool path, and, for output, discrete positioning information on punched paper tape for use by a numerically controlled machine. The CROSEC (Mod 1) program, described in this report, provides a means of extending the use of the canonical forms of the plane surfaces defined by the programmer in the part program by providing a plotting capability in which the lines in intersection, within specified limits, between a cross-sectional plane and all other defined planes are shown. A visualization of the initial plane framework on which the cutting is to be performed is thereby provided. The cross sectional plane and its dimensional limits are controlled by one plane definition and two point definitions. The plot is supplemented by printer output that aids in the interpretation of the plot. The program is written in the framework of the CDC 3800 APT 2.1 configuration. No additional program overlays or segments are necessary. This report contains a discussion of the method used, subroutine descriptions, listings and flowcharts, implementation aids, and a sample run.

PROBLEM STATUS

This is an interim report on a continuing problem.

AUTHORIZATION

NRL Problem 23Z0001

Manuscript submitted November 17, 1969.

CROSS-SECTIONAL PLOTS OF PLANE INTERSECTIONS

An Adaptation of the APT System

INTRODUCTION

This report contains a complete description of a plotting adaptation of the APT system in use at the Naval Research Laboratory (APT 2.1 on the CDC 3800). In the APT language, a plane can be defined in seven different ways; but after being processed they are all stored in a standard "canonical form." This program, called CROSEC (Mod 1), utilizes such plane canonical forms (of an APT Part Program) to obtain a plot of the intersection of one of the planes, designated as the cross-sectional plane, and all of the other planes that have been defined, processed, and stored at the end of APT Section 1. (See Ref. 1 for a detailed description of the APT system.)

The purpose of the program, primarily, is as an aid, an extra tool for the programmer in debugging his program. The defined surfaces as stored in their canonical forms provide a convenient starting point for geometric considerations. It is assumed that syntactic errors have already been discovered and corrected and that in using CROSEC the programmer wishes to verify that the surfaces he has defined do indeed describe the piece he wishes to have worked on by the tool. The hope is that verification can be accomplished easily if he can get a look at any cross section of his choosing through the conglomerate of the starting surfaces. He realizes that the plot might require some interpretation because defined surfaces intersecting together do not fully describe the finished piece. However, he accepts this limitation and looks upon the output as a working drawing, a picture of the output of Section 1. By means of this drawing and the accompanying identifying information from the printer, he should be able to make some significant debugging progress. Perhaps he will discover a section of surface that is defined improperly or over defined, or a combination of surfaces that could be redefined in a simpler manner. Also, he may discover a portion of surface that he has not yet defined and other events of this nature.

The report discusses plane equations, outlines the method of obtaining a coordinate system in the cross-sectional plane, describes the plot, and points out the limitations of the program. Also included are flowcharts, program descriptions, a complete listing, and details of implementation with APT 2.1. An example is introduced early and followed through the complete process in full detail.

DISCUSSION

The Plane

The *canonical* form for the plane, as defined in the APT system, is given by

$$AX + BY + CZ = D, \tag{1}$$

where*

*As quoted from pp. 80 and A-1 of Ref. 1.

A = X-Component of unit normal vector to plane
B = Y-Component of unit normal vector to plane
C = Z-Component of unit normal vector to plane
D = Normal distance from plane to origin.

Consider the *normal* form of the plane equation, namely

$$XCOS\alpha_x + YCOS\alpha_y + ZCOS\alpha_z = L, \tag{2}$$

where the coefficients for the X, Y, and Z coordinates in Eq. (2) are the direction cosines of the positive normal vector N from the origin to the plane. The direction cosines of the unit normal vector are identical with the direction cosines of N. The length, or absolute value, of this normal is L. Therefore, equating similar terms from Eqs. (1) and (2) yields

$$A = COS\alpha_x$$

$$B = COS\alpha_y$$

$$C = COS\alpha_z$$

$$D = L = |N| .$$

Thus, for example, A = 0, B = 1, C = 0, D = 1 is the plane passing through the point (0, 1, 0), parallel to the XZ coordinate plane, with direction angles of 90, 0, and 90 degrees, respectively, to the three axes.

Let us now take, for a more detailed example, the plane that passes through the points (1,0,0), (0,1,0), and (0,0,1) a unit distance out along each axis. To fit this approach to the definition of a plane, consider the *intercept* form for a plane equation,*

$$\frac{X}{X_1} + \frac{Y}{Y_2} + \frac{Z}{Z_3} = 1, \tag{3}$$

where X_1, Y_2, and Z_3 are the intercepts, i.e. the point (X_1, 0, 0) is the intersection of the X axis with the plane. Similarly with (0, Y_2, 0) and (0, 0, Z_3).

Using now the three unit axis points, already defined, in this intercept form, Eq. (3) leads to the simple and interesting equation

$$X + Y + Z = 1. \tag{4}$$

The correctness of Eq. (4) as truly representing the plane that passes through the three points is easily determined by setting any two of the variables equal to zero, and the remaining variable will be equal to 1. Equation (4) is illustrative of another form of an equation used to describe a plane, the *general* form, where the coefficients of the X, Y, and Z terms are considered to be *direction numbers* of the positive normal to the plane.

To go from the general form to the normal form, it is necessary to compute the direction cosines by dividing each of the coordinate coefficients in turn by the square root of the sum of the squares of all three coefficients. The length of the normal is obtained in a similar fashion by dividing the constant term by the same square-root quantity. Symbolically, if the general form is

$$PX + QY + RZ = S, \tag{5}$$

*All forms of equations for a plane can be found in Section 3, p. 2-1 of Ref. 2.