COMMONWEALTH OF AUSTRALIA

DEPARTMENT OF NATIONAL DEVELOPMENT

BUREAU OF MINERAL RESOURCES, GEOLOGY AND GEOPHYSICS

Record 1972/115

A COMPUTER PROGRAM FOR PLOTTING PERSPECTIVE DIAGRAMS OF GEOLOGICAL SECTION DATA

by

S. Henley

Record No. 1972/115

A COMPUTER PROGRAM FOR PLOTTING PERSPECTIVE DIAGRAMS OF GEOLOGICAL SECTION DATA

by

S. Henley

CONTENTS

SUMMARY

A computer program for plotting two-point perspective diagrams is described, and the method of computation of co-ordinates in the plane of the diagram is outlined. The program is applicable to three-dimensional data arranged in parallel sections or traverses, and the actual data points are plotted; the program does not attempt interpolation or smoothing.

INTRODUCTION

The program, BLOCK, accepts a string of x,y, and z data, which may be obtained by digitization of a set of stacked sections, or may represent a series of parallel traverses of any type, and plots a block diagram in two-point perspective, in any desired position relative to the observer.

Normally adjacent points are connected by straight lines, but an option in the program allows that successive points greater than a specified distance apart in x,y,z space are not connected.

Isometric projection may be simulated by using an observer-origin distance which is very large relative to distances within the co-ordinate system to be plotted; one-point perspective may be simulated by looking along directions close to the x or y axis; however, for most purposes an angle of azimuth between 30° and 60° from the axes and an angle of tilt between 20° and 35°, together with an observer-origin distance commensurate with the desired eye-diagram distance at the scale used, will be found to give the most useful and pleasing results.

TWO - POINT PERSPECTIVE

The construction of a two-point perspective block diagram is illustrated in Fig. 1, in which the points are labelled to correspond with variable names in the program.

The position of a co-ordinate system with origin O may be defined by the distance of O from the observer's eye, E', the angle of tilt, θ , that the line E'O makes with the horizontal, and the angle of azimuth, α , that E'O makes with the y-axis of the co-ordinate system.

The diagram is constructed such that -

$$H_1Z = s\ d\ \tan\alpha ,$$
$$H_rZ = s\ d\ \tan(\pi/2 - \alpha) = s\ d/\tan\alpha ,$$
and $$OZ = s\ d\ \tan\theta ,$$

where s is the scale of the diagram at the origin, and d is the distance E'O (i.e. the distance from observer to origin)

The two-dimensional co-ordinates X, Y, of any point, (x,y,z), in the diagram, relative to the origin, are derived below, with reference to construction lines in Fig. 1.

Distance OA is given by the expression:

$$OA = x.H_rO/(x + H_rO),$$

and OC, parallel to the horizon, is given by:

$$OC = OA.\ (H_rZ/H_rO).$$

Similarly, distance AC, parallel to OZ, is

$$AC = OA.\ (OZ/H_rO).$$

It is thus obvious that

$$GH_1 = H_1Z + OC$$
and $$GA = OZ - AC.$$
Thus $$H_1A = \sqrt{GH_1^2 + GA^2}$$

We can now calculate distance AE as follows -

$$AE = y.H_1A/(y + H_1A),$$

and AF, parallel to the horizon, is

$$AF = AE.\ (GH_1/H_1A).$$

Thus the two-dimensional co-ordinate (in the plane of the diagram) of point (x,y,z) is given by

$$X = OH = OC - AF$$

and Y can be found from the relation

$$Y = EH\left(= AC + EF\right) + q,$$

where $$EF = AE.\ (GA/H_1A),$$

and q is the value of the z co-ordinate transformed into the plane of the diagram, given by the following equation:

$$q = z.\ (OZ - AC - EF)\ /\ OZ.$$

PROGRAM OPERATION

The program reads scale and block-dimension data, and plots the frame of the block diagram. It then reads successive values of x,y, and z, computes the co-ordinates, and plots the points into the block, after checking (a) that the distance from the previous point is less than the critical value SEGMAX, above which no connecting line is drawn, and (b) that the values of x,y, and z are acceptable.

Input cards

Job deck structure is given below.

```
*JOB,chargecode,ident,timelimit
*EQUIP,1=PB500
*FTN,X,L
    Fortran deck inserted here
*LOAD
*RUN, timelimit, printlimit
      Number of runs (NRUN) card
      Co-ordinates card      (a)
      Scale card             (b)
      Distance limit card    (c)
      Data cards             (d)
7/8EOF
*EOD
```

Details of card formats:

Number of runs (NRUN) card. The number of different views desired is punched, right justified, in columns 1-2. The maximum number of runs allowed is 99. If more than one run is required, additional cards of types (a), (b), and (c) are inserted after the data cards (d) and the $^7_8$EOF card (i.e. between the $^7_8$EOF card and the *EOD card).

(a) Co-ordinates card.

Columns	1 - 10	Inches from left	Plotter position of origin
	11 - 20	Inches up from base	
	21 - 30	X_0	Co-ordinates of origin in input units
	31 - 40	Y_0	
	41 - 50	Z_0	
	51 - 60	X_{max}	Maximum values, in input units.
	61 - 70	Y_{max}	
	71 - 80	Z_{max}	

(b) Scale card.

Columns	1 - 10	X scale	in kilometres per input unit
	11 - 20	Y scale	
	21 - 30	Z scale	

31 - 40 P, general scale in plotter inches per kilometre: this scale is true only at the origin, since at points behind this the plotted scale is smaller, and at points in front the plotted scale is larger.

41 - 50 Angle of tilt of view, in degrees (generally between 20.0 and 35.0).

51 - 60 Azimuth angle in degrees. For viewing <u>from</u> the -y semi-circle, this angle must be between -90.0 and +90.0. To view from the +y sector, X scale and Y scale (columns 1 - 10 and 11 - 20) must be punched as negative numbers, and angles between - 90.0 and +90.0 again used.

61 - 70 Distance of origin from the eye, in kilometres.

(c) Distance limits card.

Columns 11 - 20 Maximum acceptable value of z, in input units.

21 - 30 Maximum acceptable horizontal distance between adjacent points for their join to be plotted; in input units.

(d) Data cards.

Columns	3 - 10	Identification field (alphanumeric)
	11 - 20	X value
	21 - 30	Y value
	31 - 40	Z value

<u>Output</u> A plotted two-point perspective diagram, on the 30-inch drum plotter, with the original section lines plotted to scale.

<u>Limitations</u> There is no limit to the number of data points which may be processed in any one job.

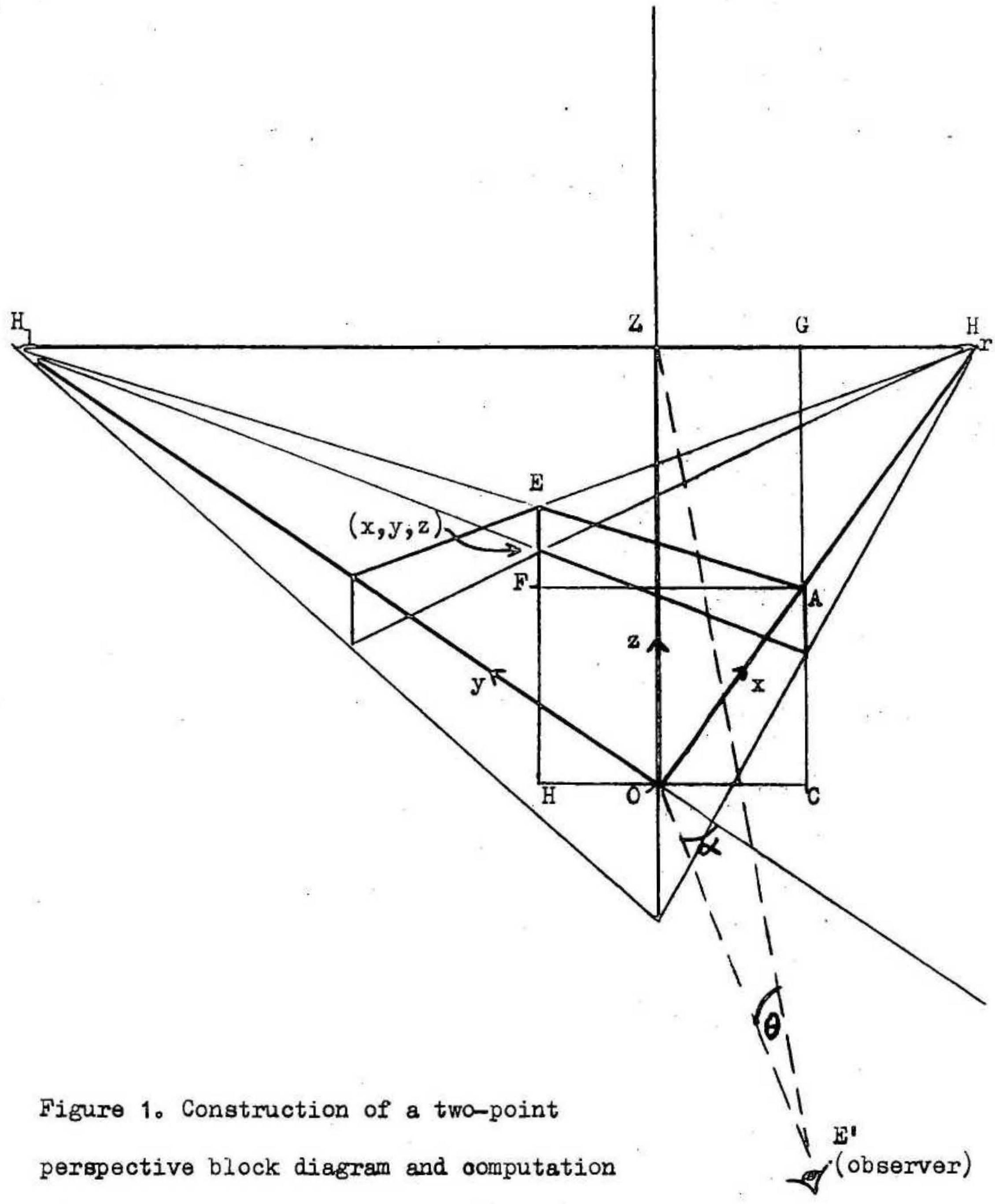

Figure 1. Construction of a two-point perspective block diagram and computation of the co-ordinates of a point (x,y,z).

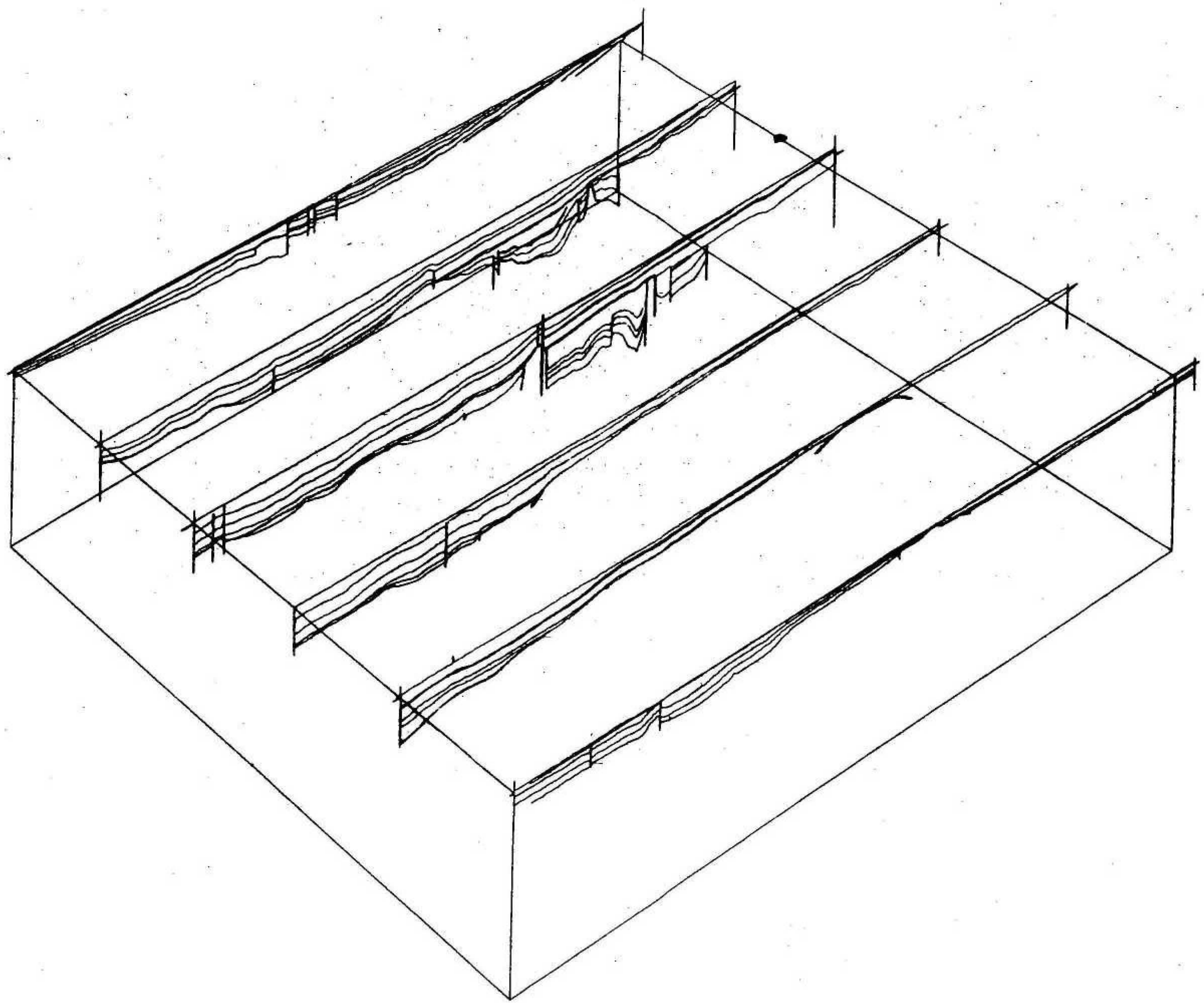

Fig. 2

```
      PROGRAM BLOCK
      DIMENSION C(10)
      COMMON HLZ,HRZ.OZ.HRO,INP,NPEN,LUN,NRUN,
     1XO,YO,XORG,YORG,ZORG,XMAX,YMAX,ZMAX
      COMMON/XYZPLOT/M
      DATA (C=23H*DRCOPDF,1,,BLOKPLOT,,,)
      NUMTIE=0
      READ 900, NUM
      IF (NUM,LE,0) NUM=NUMTIE=1
      READ 901,XO,YO.XORG,YORG,ZORG,XMAX,YMAX,ZMAX
      M=0
      LUN=1
      DO 50 NRUN=1,NUM
      CALL PLOT (0,0.0,0,1)
      CALL PLOT (1,0.1,0,2 )
      CALL PLOT (0,0.15,0,3)
   10 NPEN=3
      IF (NRUN,EQ,1) 20,30
   20 INP=2
      GO TO 40
   30 INP=10  $ REWIND 10
   40 CALL SCALEIN(XS,YS,ZS)
      CALL FRAME(XS,YS,ZS)
      PRINT 44
   44 FORMAT (1H0 ,*FRAME COMPLETED*)
      CALL XYZPLOT(XS,YS,ZS)
      PRINT 45
   45 FORMAT (1H0,*XYZPLOT COMPLETED*)
      IF (NUMTIE,EQ,1) GO TO 50
      REWIND 10
      CALL DISCDOCS(C,N)
      PRINT 46,N
   46 FORMAT (1H0,*BLOKPLOT ON DISC IF N IS 1,  N=*,I1)
   50 CONTINUE
  900 FORMAT (I2)
  901 FORMAT(8F10)
      CALL EXIT
      END
```

```
      SUBROUTINE SCALEIN(X,Y,Z)
      COMMON HLZ,HRZ,OZ,HRO,INP,NPEN,LUN,NRUN,
     1XO,YO,XORG,YORG,ZORG,XMAX,YMAX,ZMAX
      READ 10,X,Y,Z,P,TILT,ALPHA,DIST
   10 FORMAT (8F10)
      IF (ABS(ALPHA).GT.90.0) 15,16
   15 X=-X $ Y=-Y $ ALPHA=SIGN((ABS(ALPHA)-90.0),ALPHA)
   16 CONTINUE
      X=X*P $ Y=Y*P $ Z=Z*P
      Q=3.14159/180.0
      TILT=TILT*Q  $  ALPHA=ALPHA*Q
      DISTP=DIST*P
      HRZ=DISTP/TANF(ALPHA)
      HLZ=DISTP*TANF(ALPHA)
      OZ=DISTP*TANF(TILT)
      HRO=SQRT(HRZ**2+OZ**2)
      WRITE(61,20) X,Y,Z,OZ,HLZ,HRZ
   20 FORMAT(1H1,*SCALES ETC. -*/4H X= ,F7.3,*INCHES PER X INPUT UNIT*/
     14H Y= ,F7.3,* INCHES PER Y INPUT UNIT*/4H Z= ,F7.3* INCHES PER Z I
     2NPUT UNIT*/1H0,*OZ=*,F7.3,*  HLZ=*,F7.3,*  HRZ=*,F7.3)
      RETURN
      END
```

```
      SUBROUTINE FRAME(XS,YS,ZS)
      COMMON HLZ,HRZ,DZ,HRO,INP,NPEN,LUN,NRUN,
     1XO,YO,XORG,YORG,ZORG,XMAX,YMAX,ZMAX
      DIMENSION XX(8),YY(8),ZZ(8)
      PRINT 10
   10 FORMAT(1H1,*X-Y COORDINATES OF FRAME CORNERS*//)
      XX(1)=XORG $ YY(1)=YORG $ ZZ(1)=ZORG
      XX(2)=XMAX  $  YY(2)=YORG  $  ZZ(2)=ZORG
      XX(3)=XMAX  $  YY(3)=YMAX  $  ZZ(3)=ZORG
      XX(4)=XORG  $  YY(4)=YMAX  $  ZZ(4)=ZORG
      XX(5)=XORG  $  YY(5)=YMAX  $  ZZ(5)=ZMAX
      XX(6)=XMAX  $  YY(6)=YMAX  $  ZZ(6)=ZMAX
      XX(7)=XMAX  $  YY(7)=YORG  $  ZZ(7)=ZMAX
      XX(8)=XORG  $  YY(8)=YORG  $  ZZ(8)=ZMAX
      DO 30 I=1,8
      XX(I)=(XX(I)-XORG)*XS
      YY(I)=(YY(I)-YORG)*YS
      ZZ(I)=(ZZ(I)-ZORG)*ZS
      CALL COORDS (XX(I),YY(I),ZZ(I),XX(I),YY(I))
      PRINT 20,XX(I),YY(I)
   20 FORMAT(1H ,2F10.2)
   30 CONTINUE
      CALL PLOT     (XX(1),YY(1),3)
      CALL PLOT     (XX(2),YY(2),4)
      CALL PLOT     (XX(7),YY(7),4)
      CALL PLOT     (XX(2),YY(2),3)
      CALL PLOT     (XX(3),YY(3),4)
      CALL PLOT     (XX(6),YY(6),4)
      CALL PLOT     (XX(3),YY(3),3)
      CALL PLOT     (XX(4),YY(4),4)
      CALL PLOT     (XX(5),YY(5),4)
      CALL PLOT     (XX(6),YY(6),4)
      CALL PLOT     (XX(7),YY(7),4)
      CALL PLOT     (XX(8),YY(8),4)
      CALL PLOT     (XX(1),YY(1),4)
      CALL PLOT (XX(4),YY(4),4)
      CALL PLOT  (XX(5),YY(5),3)
      CALL PLOT  (XX(8),YY(8),4)
      RETURN
      END
```

```
      SUBROUTINE COORDS (X,Y,Z,XE,YEZ)
      COMMON HLZ,HRZ,OZ,HRO,INP,NPEN,LUN,NRUN,
     1XO,YO,XORG,YORG,ZORG,XMAX,YMAX,ZMAX
      OA=X*HRO/(X+HRO)
      OC=HRZ*OA/HRO
      AC=OZ*OA/HRO
      GHL=HLZ+OC
      GA=OZ-AC
      HLA=SQRT(GHL**2+GA**2)
      AE=Y*HLA/(Y+HLA)
      AF=GHL*AE/HLA
      EF=GA*AE/HLA
      ZE=Z*(OZ-AC-EF)/OZ
      XE=OC-AF
      YE=AC+EF
      YEZ=YE+ZE
      XE=XE+XO
      YEZ=YEZ+YO
C     PRINT 30,X,Y,Z,XE,YE,ZE,YEZ
C  30 FORMAT (1H ,3F10.3,10X,3F10.3,10X,F10.3)
      RETURN
      END
```

```
      SUBROUTINE XYZPLOT (XS,YS,ZS)
      COMMON HLZ,HRZ,CZ,HRO,INP,NPEN,LUN,NRUN,
     1XO,YO,XORG,YORG,ZORG,XMAX,YMAX,ZMAX
      COMMON/XYZPLOT/M
      COMMON/SEGMAX/SEGMAX
      I=0
      CALL BCDERSET(INP,1)
      IF (NRUN.GT.1) GO TO 40
      XMAX=(XMAX-XORG)*XS
      YMAX=(YMAX-YORG)*YS
      ZMAX=(ZMAX-ZORG)*ZS
      READ 10,DUM,ZMIN,SEGMAX
      SEGMAX=SEGMAX*XS
      YOLD=XOLD=10.0**10
      MM=1
    5 READ (INP,10) NAME,X,Y,Z
      IF (INBCDCKF(INP).EQ.1) GO TO 5
      IF (EOF,INP) 70,20
   10 FORMAT (2X,A8,3F10)
   20 X=(X-XORG)*XS $ Y=(Y-YORG)*YS $ Z=(Z-ZORG)*ZS
      IF (Z.GT.ZMIN) GO TO 75
      IF (ABS(Z).GT.ABS(ZMAX)) GO TO 75
      O=0.0
   25 I=I+1
      NPEN=4
      IF (HYPOTNUS(X,XOLD,Y,YOLD).GT.SEGMAX) 30,35
   30 I=0
      O=0.0  $  ANTONIO=10.0**10
      WRITE (10) O ,O , ANTONIO
      M=M+1
   35 IF (I.LT.1) NPEN=3
      CALL COORDS(X,Y,Z,XE,YEZ)
      CALL PLOT (XE,YEZ,NPEN,LUN)
      IF (NRUN.GT.1) GO TO 40
      XOLD=X $ YOLD=Y
      WRITE (10) X,Y,Z
      M=M+1
      IF (M/100*100.EQ.M) PRINT 36 ,M
   36 FORMAT(I12,*TH POINT PLOTTED*)
      IF (INP.EQ.2) GO TO 5
   40 IF (MM.EQ.M) RETURN
      READ (10) X,Y,Z
      MM=MM+1
      GO TO 25
   70 RETURN
   75 I=0
      PRINT 80,NAME,X,Y,Z,M
   80 FORMAT (1H0,*COORDINATE TOO LARGE OR SMALL - SAMPLE POINT *,A8/
     ,* XYZ*,3F10.4,I10,*TH POINT*)
      GO TO 5
      END
```

```
      FUNCTION HYPOTNUS(XA,XB,YA,YB)
      HYPOTNUS=SQRT((XA-XB)**2+(YA-YB)**2)
      RETURN
      END
```