UNITED STATES
DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

National Center for Earthquake Research
345 Middlefield Road
Menlo Park, California 94025

DOCUMENTATION OF PROGRAMS THAT COMPUTE

1) STATIC TILTS FOR A SPATIALLY VARIABLE SLIP DISTRIBUTION, AND 2) QUASI-STATIC TILTS PRODUCED BY AN EXPANDING DISLOCATION LOOP WITH A SPATIALLY VARIABLE SLIP DISTRIBUTION

Stuart McHugh

Open-File Report #76-578

ABSTRACT

The material in this report is concerned with the effects of a vertically oriented rectangular dislocation loop on the tilts observed at the free surface of an elastic half-space. Part I examines the effect of a spatially variable static strike-slip distribution across the slip surface. The tilt components as a function of distance parallel, or perpendicular, to the strike of the slip surface are displayed for different slip-versus-distance profiles. Part II examines the effect of spatially and temporally variable slip distributions across the dislocation loop on the quasi-static tilts at the free surface of an elastic half space. The model discussed in part II may be used to generate theoretical tilt versus time curves produced by creep events.

INTRODUCTION:

Program RCTNGL computes the tilt-versus-distance profiles observed for a variable strike-slip distribution across a slip surface. The model assumes a vertically oriented rectangular dislocation loop embedded in an elastic half-space. The distance axis of the profile may be parallel or perpendicular to the strike of the slip surface. The position of the slip surface relative to the position of the profile is arbitrary, and various slip distributions may be modelled by changing a few lines in the program.

Program SLPPRP also assumes a vertically oriented rectangular dislocation loop embedded in an elastic half-space to generate a quasi-static tilt-time profile produced by spatially and temporally variable slip distributions. This program may be used to model creep-related tilt changes; and provides a self-consistent method of separating the tilt changes produced by propagation effects of the boundaries of the slip surface from the tilt changes produced by time variations in the slip distribution across the slip surface.

CONTENTS PAGE

DOCUMENTATION OF PROGRAM RCTNGL

PART I

DOCUMENTATION FOR PROGRAM RCTNGL

INTRODUCTION:

The tilts, produced by a vertically oriented rectangular dislocation loop, observed at the free surface of an elastic half-space depend upon the source-station geometry, the amount and type of slip, and the slip distribution. This program uses the expressions for tilt given in Press (1965), and allows the user to select the profile desired and the distribution of slip. The output consists of the tilt components and tilt amplitude and azimuth versus distance. With this program, the user may, for example, examine the dependence of the surface tilts on the spatial frequency of the slip distribution.

INPUT:

The geometry and notation assumed are shown in Figures 1.1 and 1.2, and definitions of the parameters used are provided in Appendix A. Although this program assumes strike slip displacement, dip-slip displacement may be used by changing the expressions in subroutine TILT to conform to those given in Press (1965). This program also assumes a cosinusoidal slip distribution with spatial frequencies FREQX1 in the X1 direction and FREQX3 in the X3 direction. If other slip distributions are desired, lines 57 - 59 in the main program may be altered. For example, if line 59 is deleted, the slip will be alternately right- and left-lateral across the slip surface (if line 59 is retained, the absolute magnitude of the cosinusoidal slip distribution is computed). If, for example, lines 57 through 59 are changed to:

```
ALPHA = + .5* SQRT (((I-(N2/2.))**2)+((J-(N1/2.))**2))
U = UMAX * (-1. + EXP (ALPHA))
```

the slip distribution will increase exponentially away from the center of the slip zone. Or, if lines 57-59 are changed to:

```
ALPHA = -.5* SQRT (((I-(N2/2.)**2)+((5-(N1/2.)**2))
U = -UMAX * ((N2 + N1)/2.)* EXP (ALPHA)
```

the slip distribution will decrease exponentially away from the center of the slip zone. (The value of .5 and the pre-exponential factor were arbitrarily chosen to scale the slip magnitude.) The subrectangles are labelled as shown in Figure 1.3. Notice that the slip is constant across <u>each</u> subrectangle, but may vary from subrectangle to subrectangle.

The program is intended for use on the LBL 6600 B or C machine and the Tektronix 4010-1 terminal. It requires 40K of core and may be accessed using

```
^LOAD, RCTNGL, MCHUGH   .
```

It will automatically link to the appropriate plotting subroutines, and the ^LOAD command may be followed by a ^RUN. The computer responds as follows:

1) 1 = plot vs. X1, 2 = plot vs. X2

Enter 1 or 2 to select the profile desired.

2) Specify initial and final points of plot.

Enter the beginning and end points of the profile in kilometers.

3) Specify value of {X1, X2} that profile is computed for.

Enter the X1 (X2) position of the X2 (X1) profile in kilometers.

4) Specify C1X1FN, C1X3FN, C2X3FN, C3X1FN

Enter the 4 endpoints of the slip zone.

5) Specify number of points in profile,

Number of columns and rows,

Spatial frequency of slip in X1 direction,

Spatial frequency of slip in X3 direction

And maximum slip

Enter the 5 quantities required.

6) Specify angle of fault to NS axis, in degrees.

Enter the angle between the strike of the slip surface and north.

OUTPUT:

1) MIN/MAX values of EW component (numerical values)

MIN/MAX values of NS component (numerical values)

MIN/MAX values of amplitude (numerical values)

MIN/MAX values of azimuth (numerical values)

(Note tilt amplitudes are in microradians azimuth in degrees)

The following are plots of the EW and NS components of tilt and the tilt amplitude and azimuth (measured clockwise from north).

0 = Re-Start, 1 = Continue

Entering a zero causes the program to start at step (1). In the Input section, a 1 causes the program to continue.

2) Write plot title, 80 characters

Enter up to 80 alphanumeric characters.

3) Set horizontal scale? Y or N (= Blank)

Entering a Y causes the computer to respond:

MIN/MAX X values

Entering an N or (blank) causes the horizontal scale to terminate at the endpoints selected in part 2 of Input.

4) Set vertical scale? Y or N (= Blank)

The procedure is the same as in part 3 above.

The output consists of a sequence of plots of the EW tilt component (equivalent to $\frac{\partial W}{\partial X_2}$ if theta equals zero), the NS component (equivalent to $\frac{\partial W}{\partial X_1}$ if theta equals zero), the tilt amplitude, and the tilt azimuth. The profiles are parallel to the X1 or X2 directions as specified in part 1 in Input.

RESULTS:

Examples of the variation in tilt for the profiles and slip distributions indicated are shown in the 'Examples' section. The examples are shown on pages 1-7 through 1-38. The major points to be noted are that 1) as the spatial frequency of slip is increased the surface tilts approach those expected for constant displacement across the slip zone and 2) as the slip distribution is confined to some small region of the slip surface, the tilts approach those expected for a point source at the region of maximum slip. In principle, at least, the surface tilt field may provide some information about spatial variations in the slip distribution.

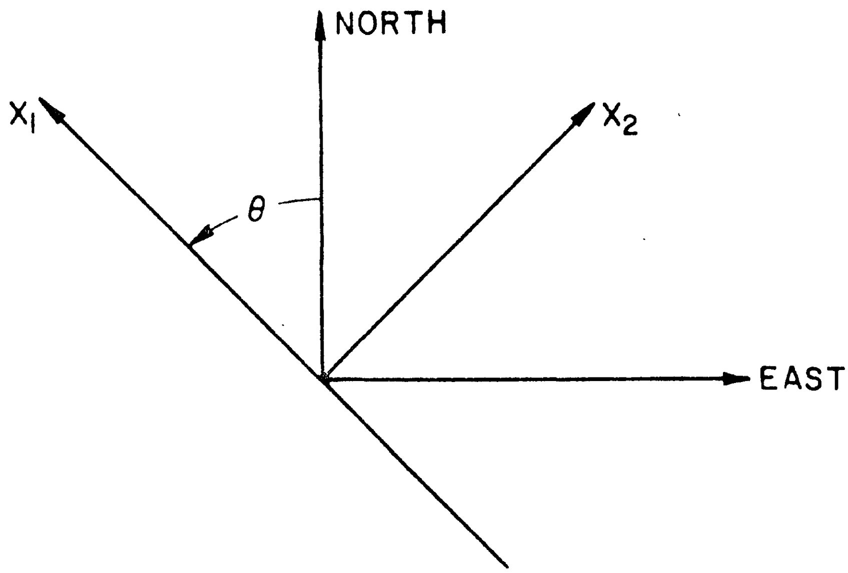


FIG. 1.1 SLIP SURFACE ORIENTATION RELATIVE TO OBSERVER

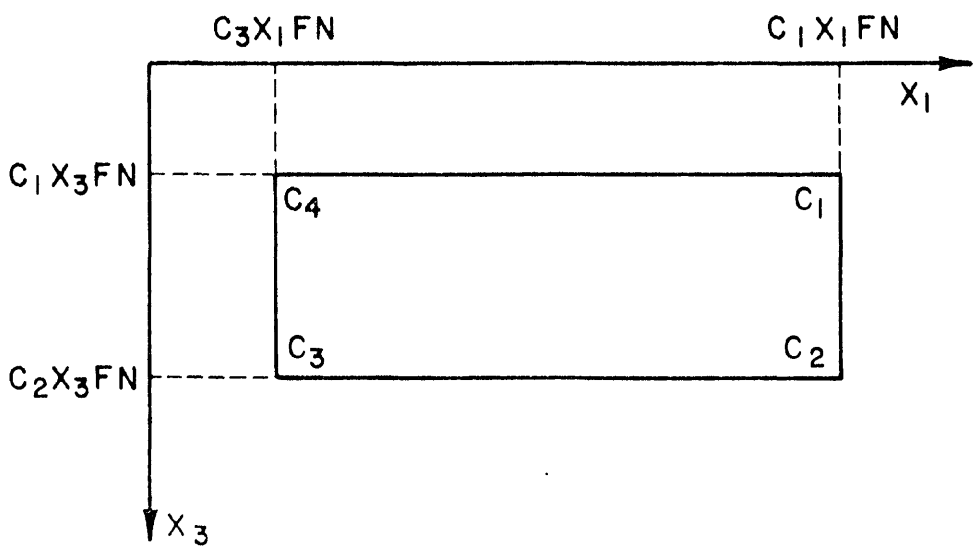


FIG. 1.2 POSITION OF SLIP SURFACE BOUNDARIES

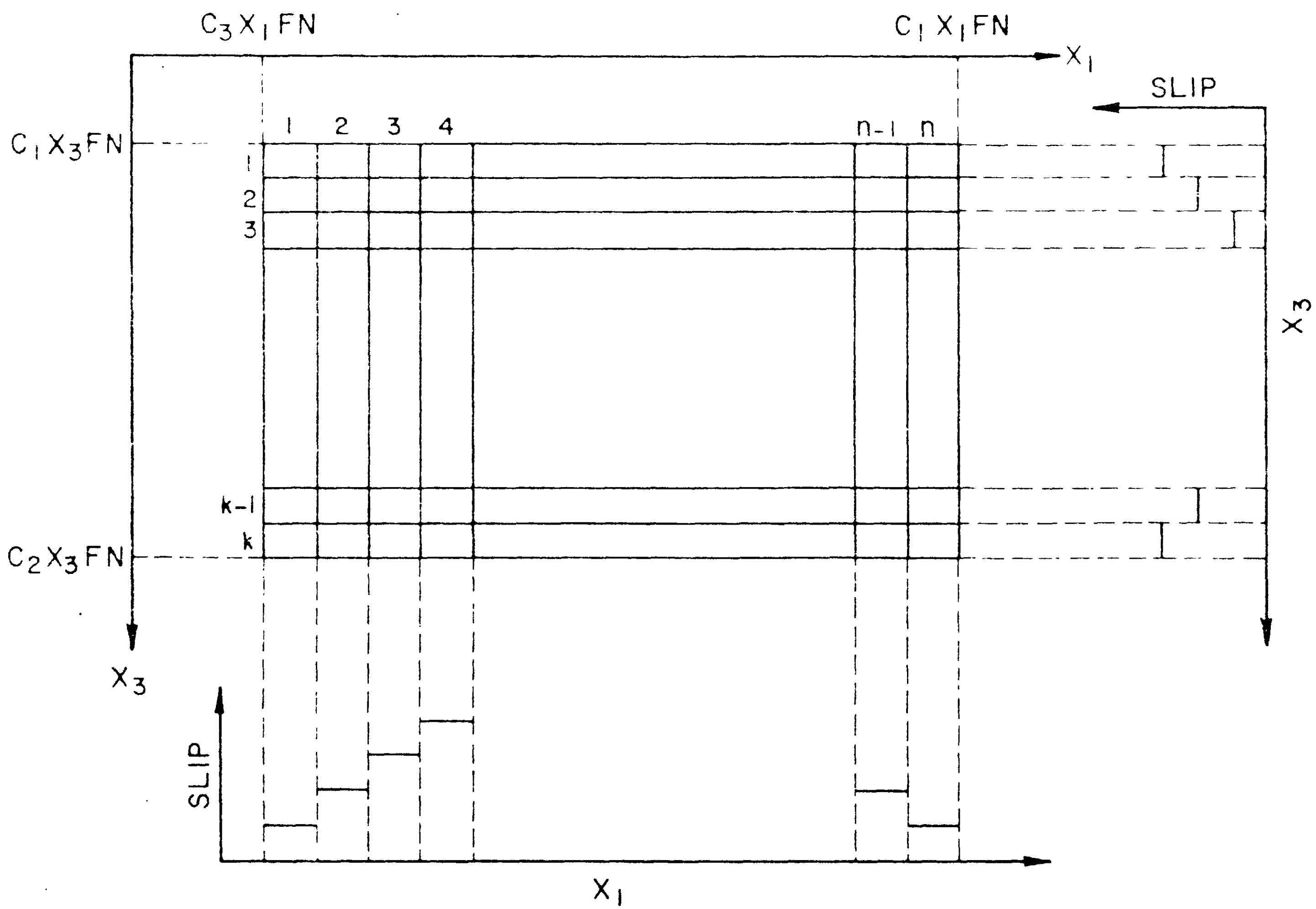


FIG. 13 LABELLING OF SUBRECTANGLES AND A HYPOTHETICAL SLIP DISTRIBUTION

APPENDIX A

Parameters used in RCTNGL:

IFLAG - determines whether tilt profile is parallel to the X1 or X2 the axis

BEGIN, END - endpoints of the tilt profile

CNSTNT - position of profile

C1X1FN, C1X3FN, C2X3FN, C3X1FN - coordinates of slip surface

M - number of points in tilt profile

N1, N2 - number of columns and rows respectively that comprise slip surface

FREQX1, FREQX3 - spatial frequencies in X1 and X3 directions respectively

UMAX - maximum displacement on slip surface

Theta - angle between strike of fault and north

EXAMPLES OF RCTNGL

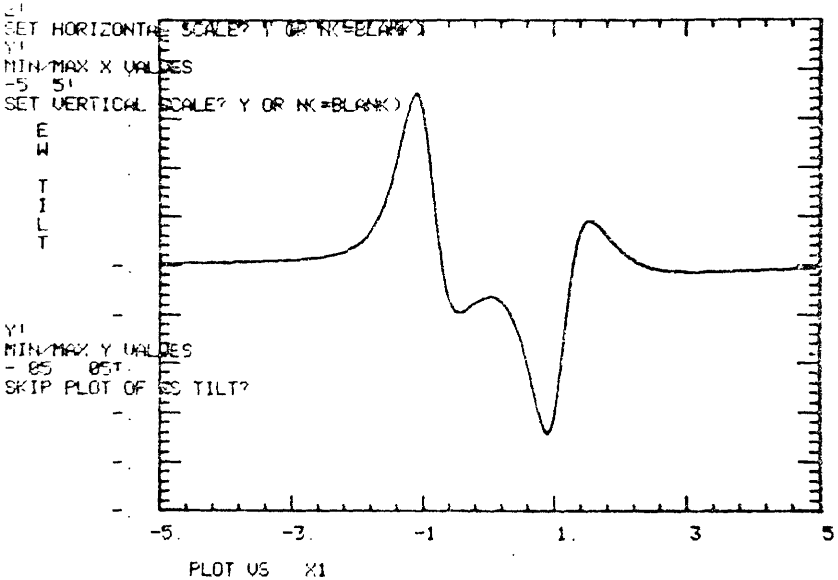
RCTNGL

Cosinusoidal Slip Distribution

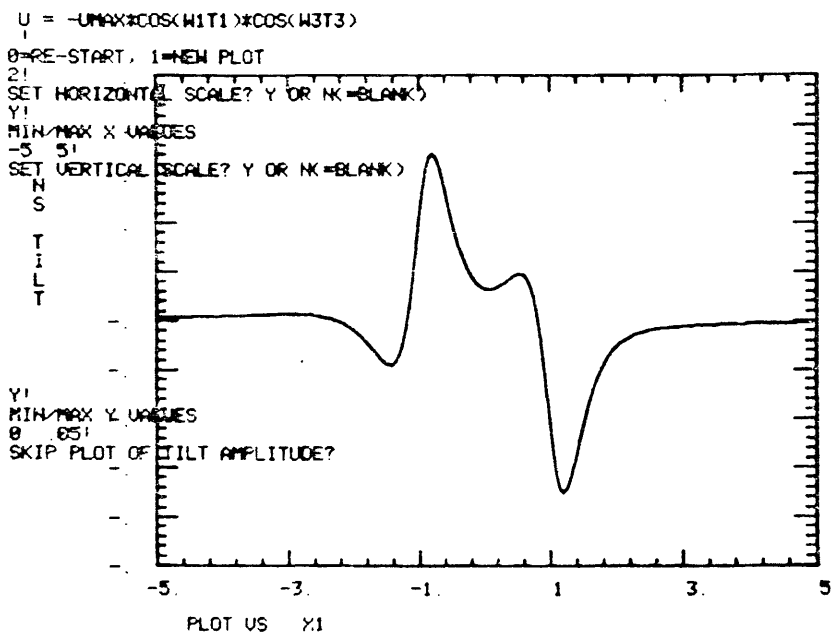
The slip distribution across the slip zone is specified in lines 57-59 of the documentation for RCTNGL (Page 1-39). Note that the absolute value of the slip is used in these examples.

```
^RUN!
1=PLOT VS. X1, 2=PLOT VS. X2
1!
SPECIFY INITIAL AND FINAL POINTS OF PLOT
-5  5!
SPECIFY VALUE OF X2 THAT PROFILE IS COMPUTED FOR
 5!
SPECIFY C1X1FN,C1X3FN,C2X3FN,C3X1FN
1 0  1 -1!
SPECIFY NUMBER OF POINTS IN PROFILE,
NUMBER OF COLUMNS AND ROWS,
SPATIAL FREQUENCY OF SLIP IN X1 DIRECTION,
SPATIAL FREQUENCY OF SLIP IN X3 DIRECTION,
AND MAXIMUM SLIP
2?? ??  1  5  0  1!
SPECIFY ANGLE OF FAULT TO NS AXIS, IN DEGREES
45!
MIN /MAX. VALUES OF EW COMPONENT  -3 423E-02   3.587E-02
MIN /MAX. VALUES OF NS COMPONENT  -3 587E-02   3.422E-02
MIN /MAX  VALUES OF AMPLITUDE      7 175E-04   3 781E-02
MIN /MAX  VALUES OF AZIMUTH        5 975     358 148
(NOTE  TILT AMPLITUDES ARE IN MICRORADIANS
AZIMUTH IN DEGREES)
THE FOLLOWING ARE PLOTS OF THE EW AND NS
COMPONENTS OF TILT, AND THE TILT AMPLITUDE
AND AZIMUTH (MEASURED CLOCKWISE FROM NORTH).
0=RE-START, 1=CONTINUE
1!
WRITE PLOT TITLE, 80 CHARACTERS
U = -UMAX*COS(W1T1)*COS(W3T3)!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
-5  5!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
- 05  .05
```

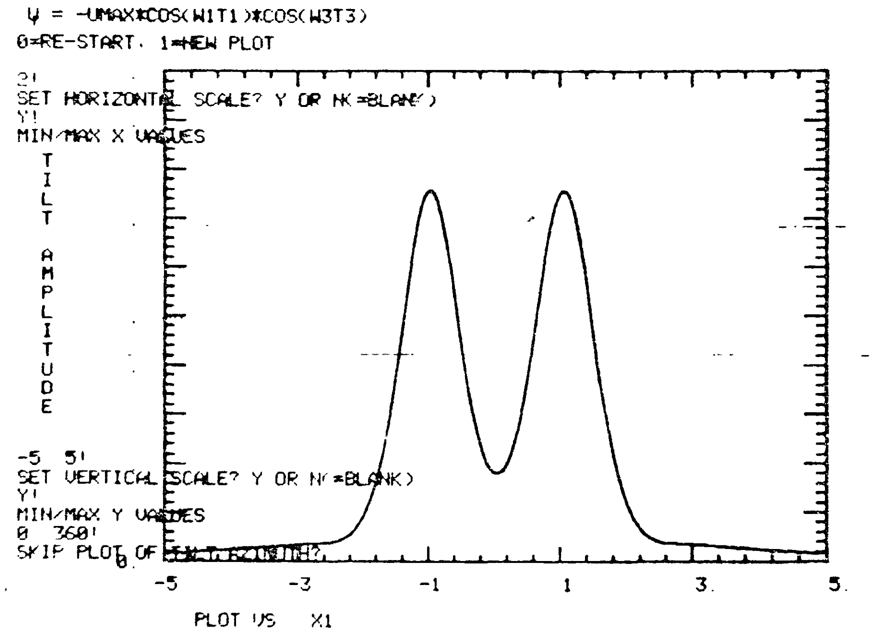
```
U = -UMAX*COS(W1T1)*COS(W3T3)
0=RE-START, 1=NEW PLOT
1!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
-5  5!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
- 05  05!
SKIP PLOT OF NS TILT?
```



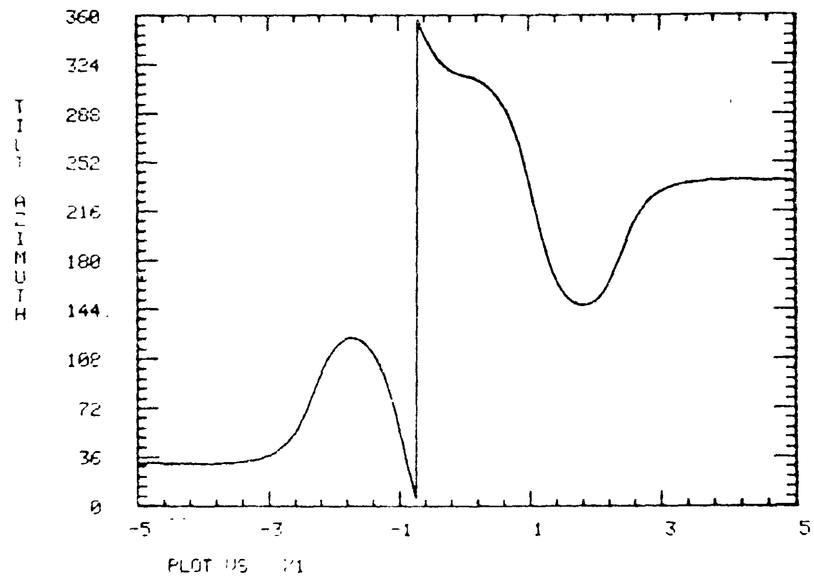
1-11


U = -UMAX*COS(W1T1)*COS(W3T3)
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR N(=BLANK)
NS TILT
Y!
MIN/MAX Y VALUES
0 .05!
SKIP PLOT OF TILT AMPLITUDE?
-5.
-3.
-1.
1
3.
5.
PLOT VS X1

1-1

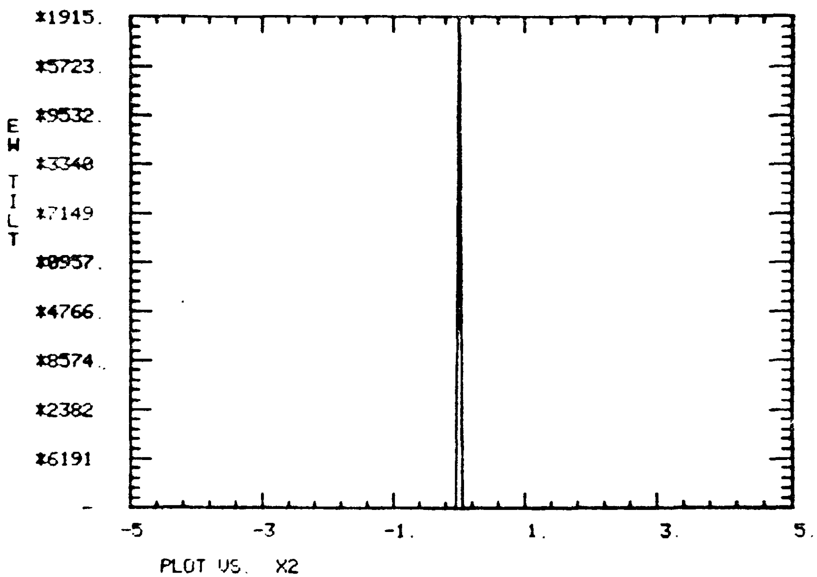

U = -UMAX*COS(W1T1)*COS(W3T3)
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
TILT AMPLITUDE
-5 5!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
0 360!
SKIP PLOT OF
0.
-5
-3
-1
1
3.
5.
PLOT VS X1

U = -UMAX*COS(W1T1)*COS(W3T3)



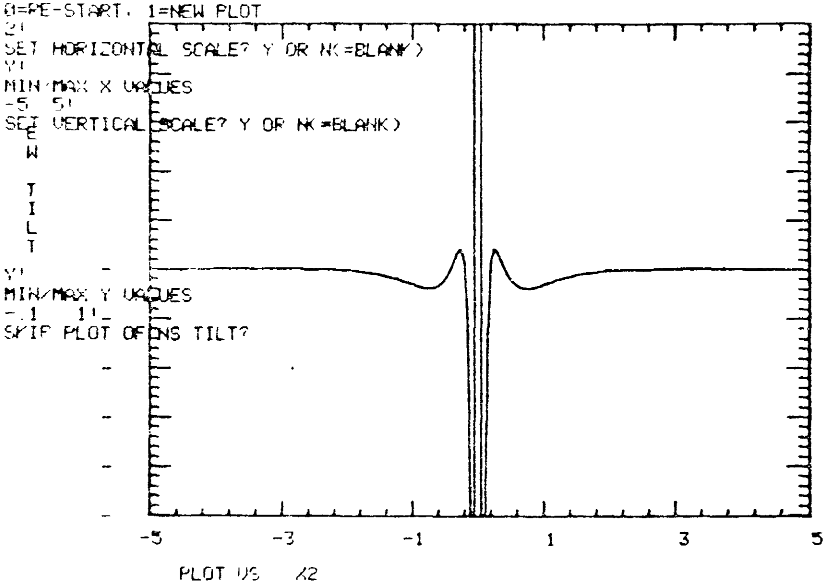

```
1=PLOT VS X1, 2=PLOT VS X2          SKIP PLOT OF EW TILT?
2'
SPECIFY INITIAL AND FINAL POINTS OF PLOT
-5 5'
SPECIFY VALUE OF X1 THAT PROFILE IS COMPUTED FOR
 5'
SPECIFY C1X1FN,C1X3FN,C2X3FN,C3X1FN
1 0 1 -1'
SPECIFY NUMBER OF POINTS IN PROFILE,
NUMBER OF COLUMNS AND ROWS,
SPATIAL FREQUENCY OF SLIP IN X1 DIRECTION,
SPATIAL FREQUENCY OF SLIP IN X3 DIRECTION,
AND MAXIMUM SLIP
200  20 20 4 4 1'
SPECIFY ANGLE OF FAULT TO NS AXIS, IN DEGREES
0'
MIN /MAX  VALUES OF EW COMPONENT  -1 893E-01   3 412E+10
MIN /MAX  VALUES OF NS COMPONENT  -2 607E-02   1 365E+11
MIN /MAX  VALUES OF AMPLITUDE      5 454E-04   1.407E+11
MIN /MAX  VALUES OF AZIMUTH          852      359 240
(NOTE  TILT AMPLITUDES ARE IN MICRORADIANS
AZIMUTH IN DEGREES)
THE FOLLOWING ARE PLOTS OF THE EW AND NS
COMPONENTS OF TILT, AND THE TILT AMPLITUDE
AND AZIMUTH (MEASURED CLOCKWISE FROM NORTH)
0=RE-START, 1=CONTINUE
1'
WRITE PLOT TITLE, 80 CHARACTERS
U = -ABS(UMAX*COS(W1X1)*COS(W2* *\23X3))'
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y'
MIN/MAX X VALUES
-5 5'
SET VERTICAL SCALE? Y OR N(=BLANK)
```

U = -ABS(UMAX*COS(W1X1)*COS(W3X3))

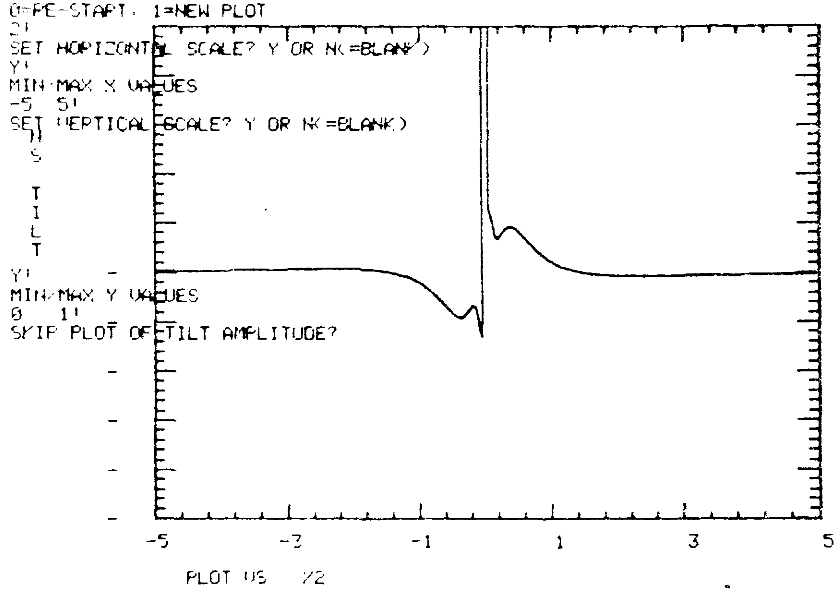



```
0=RE-START, 1=NEW PLOT
1!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
-5  5!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
-.1  .1!
SKIP PLOT OF EW TILT?
```

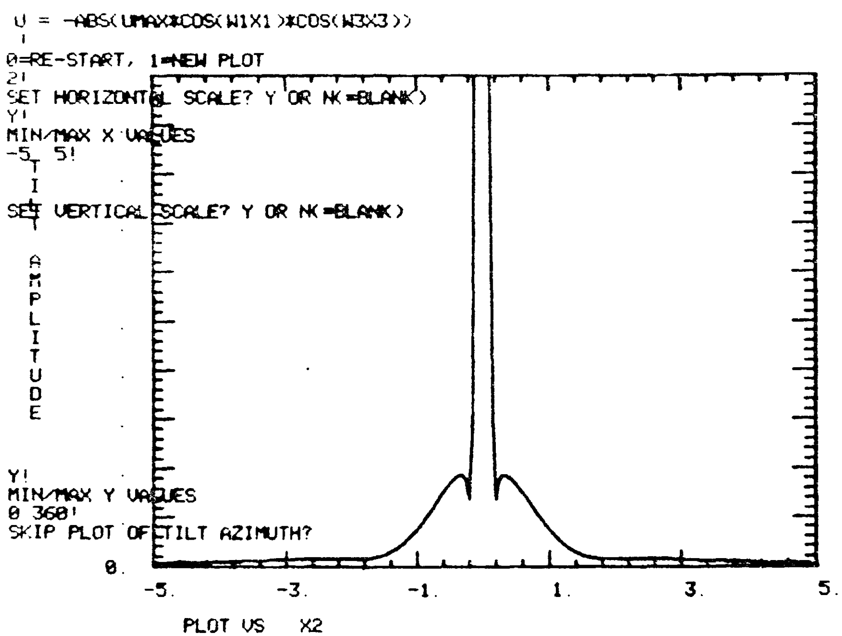
```
U = -ABS(UMAX*COS(W1X1)*COS(W3X3))
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
-.1 1!
SKIP PLOT OF NS TILT?
```



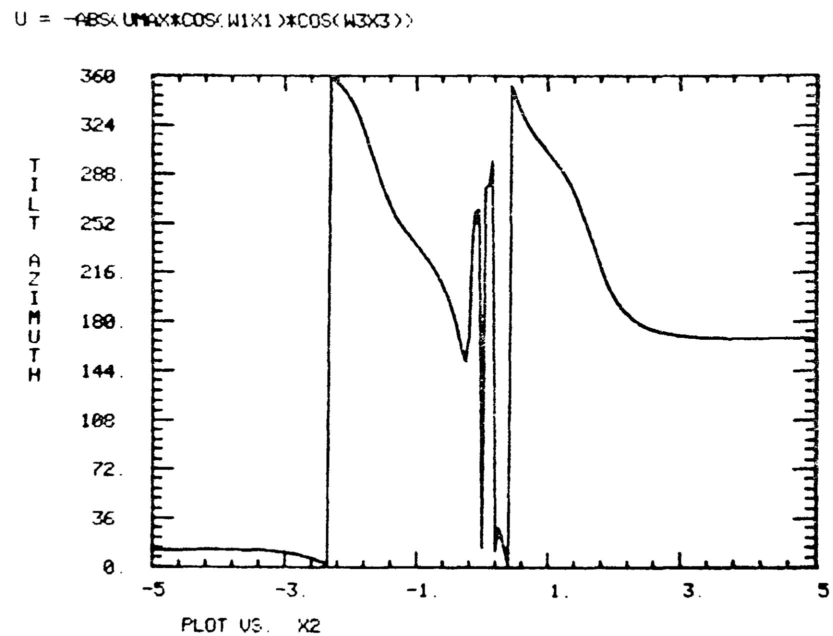

```
U = -ABS(UMAX*COS(W1X1)*COS(W3X3))
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
0 1!
SKIP PLOT OF TILT AMPLITUDE?
```



1-19

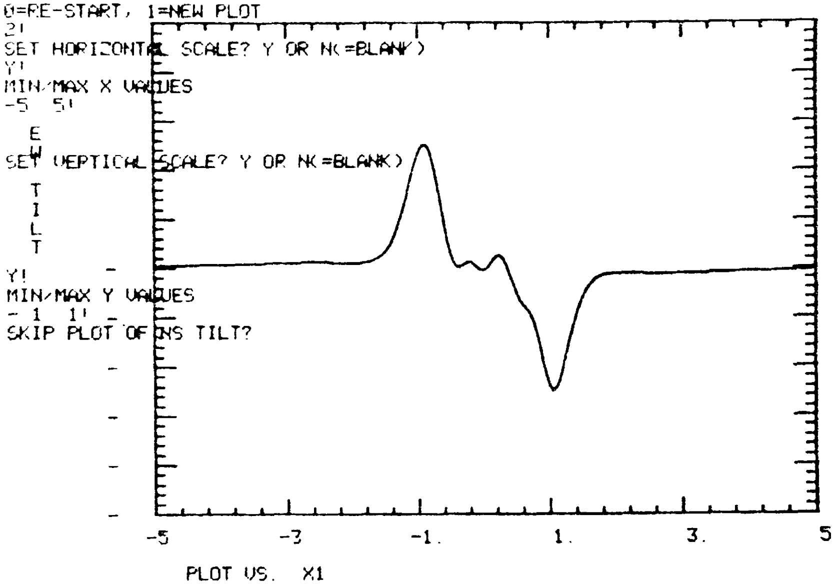

U = -ABS(UMAX*COS(W1X1)*COS(W3X3))
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
TILT
SET VERTICAL SCALE? Y OR NK=BLANK)
AMPLITUDE
Y!
MIN/MAX Y VALUES
0 360!
SKIP PLOT OF TILT AZIMUTH?
0.
-5.
-3.
-1.
1.
3.
5.
PLOT VS X2

1-20

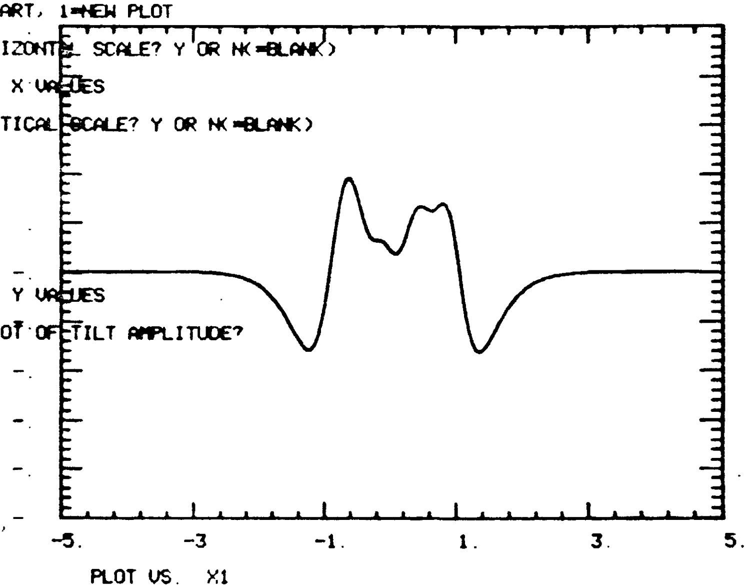

U = -ABS(UMAX*COS(W1X1)*COS(W3X3))
TILT AZIMUTH
360
324
288.
252
216.
180.
144.
108
72.
36
0.
-5
-3.
-1.
1.
3.
5.
PLOT VS. X2

```
0=RE-START, 1=NEW PLOT
0!
1=PLOT VS  X1, 2=PLOT VS. X2
1!
SPECIFY INITIAL AND FINAL POINTS OF PLOT
-5  5!
SPECIFY VALUE OF X2 THAT PROFILE IS COMPUTED FOR
 5!
SPECIFY C1X1FN,C1X3FN,C2X3FN,C3X1FN
1 0 1 -1!
SPECIFY NUMBER OF POINTS IN PROFILE,
NUMBER OF COLUMNS AND ROWS,
SPATIAL FREQUENCY OF SLIP IN X1 DIRECTION,
SPATIAL FREQUENCY OF SLIP IN X3 DIRECTION,
AND MAXIMUM SLIP
200 10 1 2 0 1!
SPECIFY ANGLE OF FAULT TO NS AXIS, IN DEGREES
0!
MIN./MAX  VALUES OF EW COMPONENT  -4.974E-02   5.024E-02
MIN./MAX. VALUES OF NS COMPONENT  -3.221E-02   3.809E-02
MIN./MAX. VALUES OF AMPLITUDE      9.266E-04   5.067E-02
MIN./MAX. VALUES OF AZIMUTH          945      357.370
(NOTE  TILT AMPLITUDES ARE IN MICRORADIANS
AZIMUTH IN DEGREES)
THE FOLLOWING ARE PLOTS OF THE EW AND NS
COMPONENTS OF TILT, AND THE TILT AMPLITUDE
AND AZIMUTH (MEASURED CLOCKWISE FROM NORTH).
0=RE-START, 1=CONTINUE
1!
WRITE PLOT TITLE, 80 CHARACTERS
U = -ABS(UMAX*COS(W1X1)*COS(W3X3))!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
- 1  .1!
SKIP PLOT OF EW TILT?
```

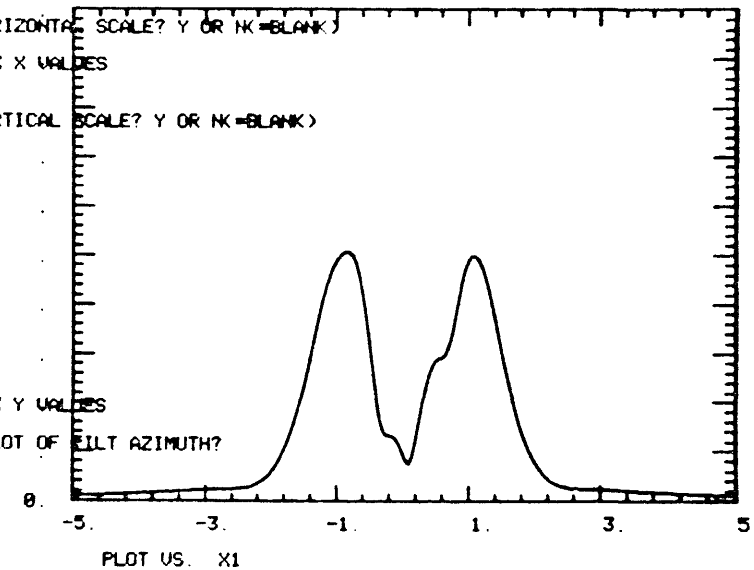
```
U = -ABS(UMAX*COS(W1X1)*COS(W3X3))
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
-5  5!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
- 1  1!
SKIP PLOT OF NS TILT?
```



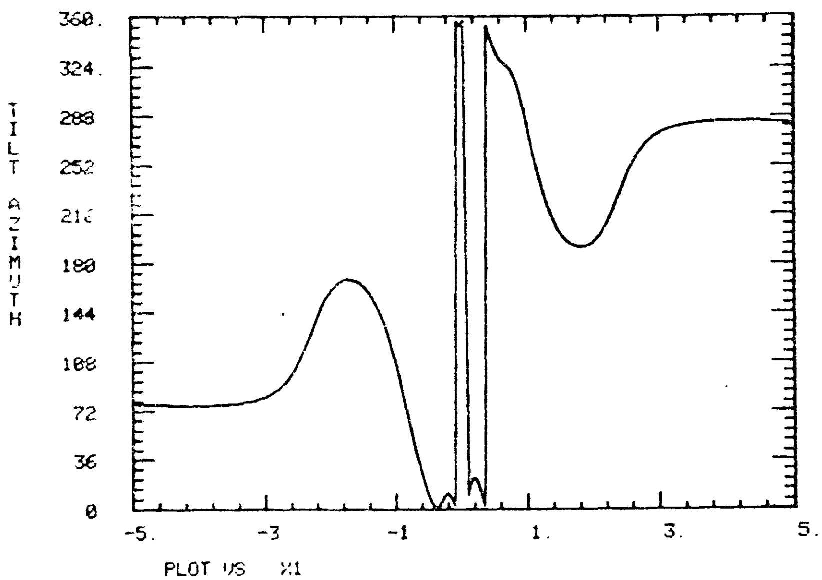
```
U = -ABS(UMAX*COS(W1X1)*COS(W3X3))
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX Y VALUES
0 .1!
SKIP PLOT OF TILT AMPLITUDE?
```



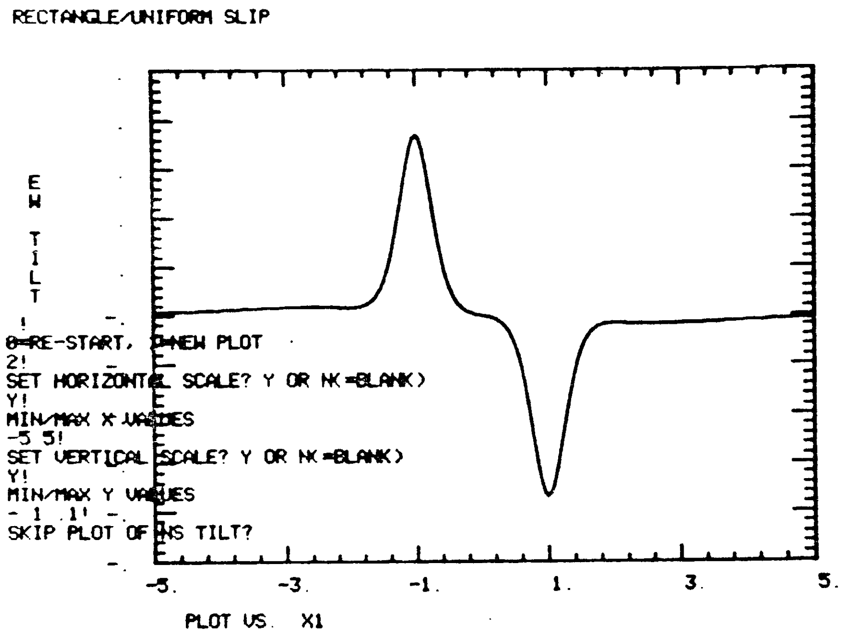

```
U = -ABS(UMAX*COS(W1X1)*COS(W3X3))
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX Y VALUES
0 360!
SKIP PLOT OF TILT AZIMUTH?
```

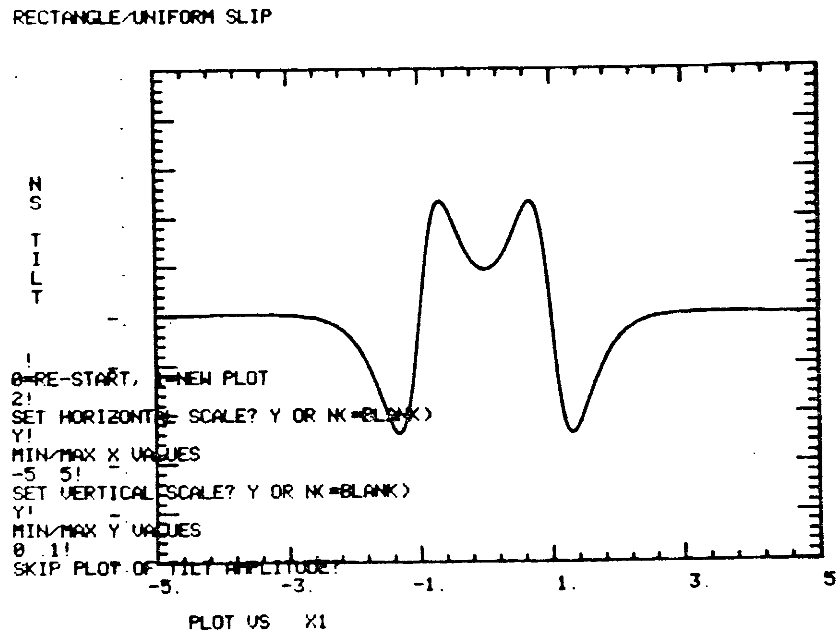


U = -ABS(UMAX*COS(W1X1)*COS(W3X3))

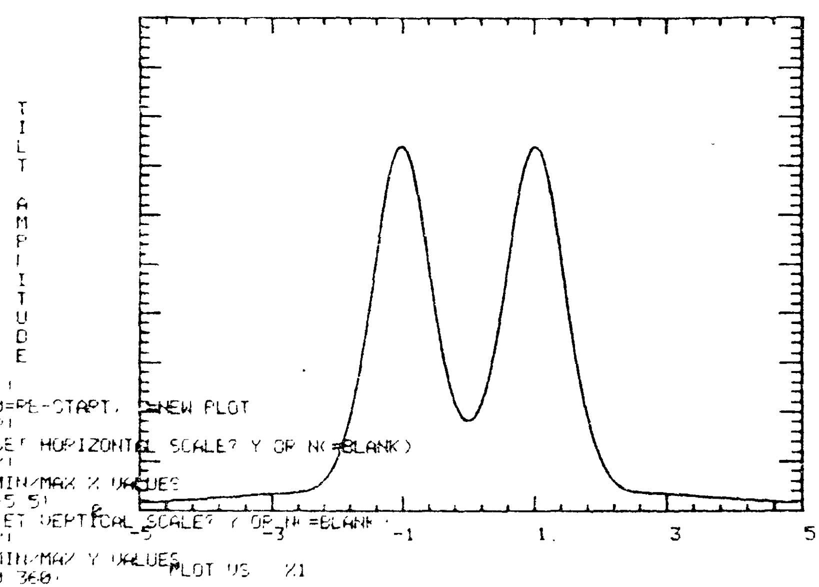



```
!
0=RE-START, 1=NEW PLOT
0!
1=PLOT VS  X1, 2=PLOT VS. X2
1!
SPECIFY INITIAL AND FINAL POINTS OF PLOT
-5  5!
SPECIFY VALUE OF X2 THAT PROFILE IS COMPUTED FOR
 5!
SPECIFY C1X1FN,C1X3FN,C2X3FN,C3X1FN
1 0 1 -1!
SPECIFY NUMBER OF POINTS IN PROFILE,
NUMBER OF COLUMNS AND ROWS,
SPATIAL FREQUENCY OF SLIP IN X1 DIRECTION,
SPATIAL FREQUENCY OF SLIP IN X3 DIRECTION,
AND MAXIMUM SLIP
200 1 1 0 0 1!
SPECIFY ANGLE OF FAULT TO NS AXIS, IN DEGREES
0!
MIN /MAX  VALUES OF EW COMPONENT  -7.383E-02   7.383E-02
MIN./MAX. VALUES OF NS COMPONENT  -4.813E-02   4.674E-02
MIN /MAX  VALUES OF AMPLITUDE      1.468E-03   7.384E-02
MIN /MAX. VALUES OF AZIMUTH          .719     360.000
(NOTE  TILT AMPLITUDES ARE IN MICRORADIANS
AZIMUTH IN DEGREES)
THE FOLLOWING ARE PLOTS OF THE EW AND NS
COMPONENTS OF TILT, AND THE TILT AMPLITUDE
AND AZIMUTH (MEASURED CLOCKWISE FROM NORTH)
0=RE-START, 1=CONTINUE
1!
WRITE PLOT TITLE, 80 CHARACTERS
RECTANGLE/UNIFORM SLIP!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
-.1  .1!
SKIP PLOT OF EW TILT?
```


RECTANGLE/UNIFORM SLIP
E
W
T
I
L
T
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX Y VALUES
-.1 .1!
SKIP PLOT OF NS TILT?
-5.
-3.
-1.
1.
3.
5.
PLOT VS. X1

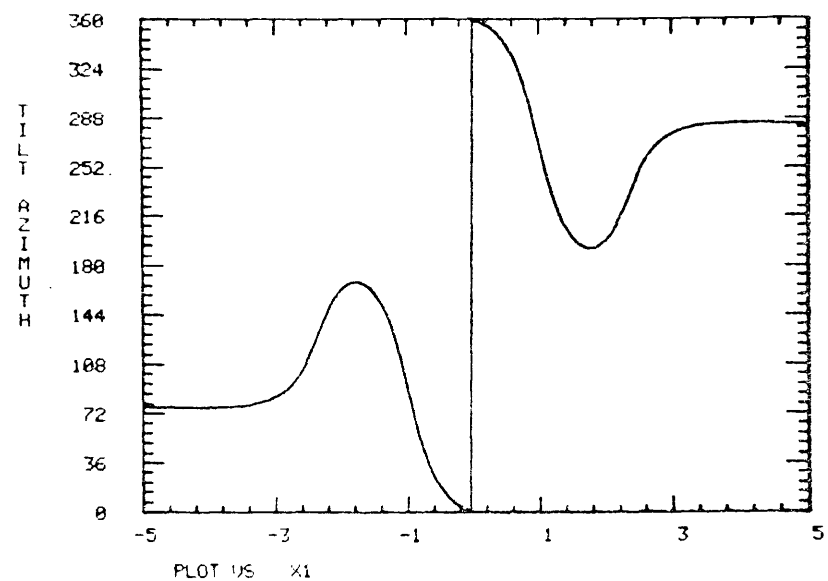

RECTANGLE/UNIFORM SLIP
N
S
T
I
L
T
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR NK=BL[illegible]K)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX Y VALUES
0 .1!
SKIP PLOT OF TILT AMPLITUDE?
-5.
-3.
-1.
1.
3.
5.
PLOT VS X1

RECTANGLE UNIFORM SLIP



0=RE-START, 1=NEW PLOT
?1
SET HORIZONTAL SCALE? Y OR N(=BLANK)
?Y
MIN/MAX X VALUES
-5 5)
SET VERTICAL SCALE? Y OR N(=BLANK)
?Y
MIN/MAX Y VALUES
0 360)
SKIP PLOT OF TILT AZIMUTH?

RECTANGLE/UNIFORM SLIP



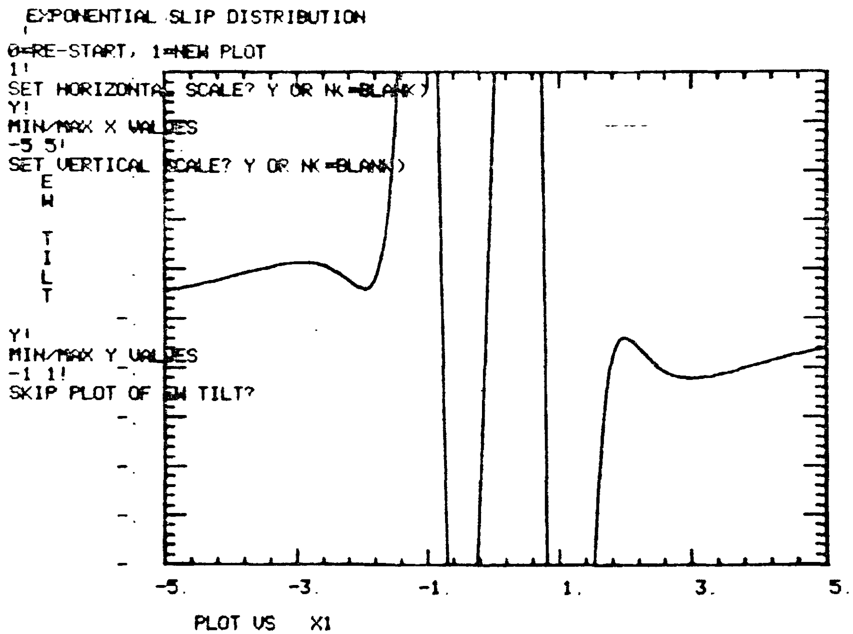
RCTNGL

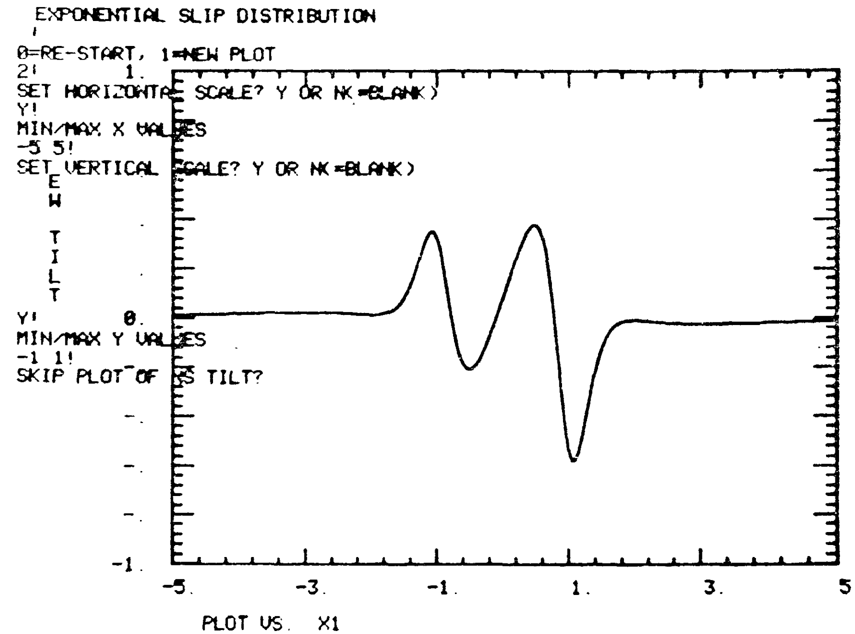
Exponential Slip Distribution

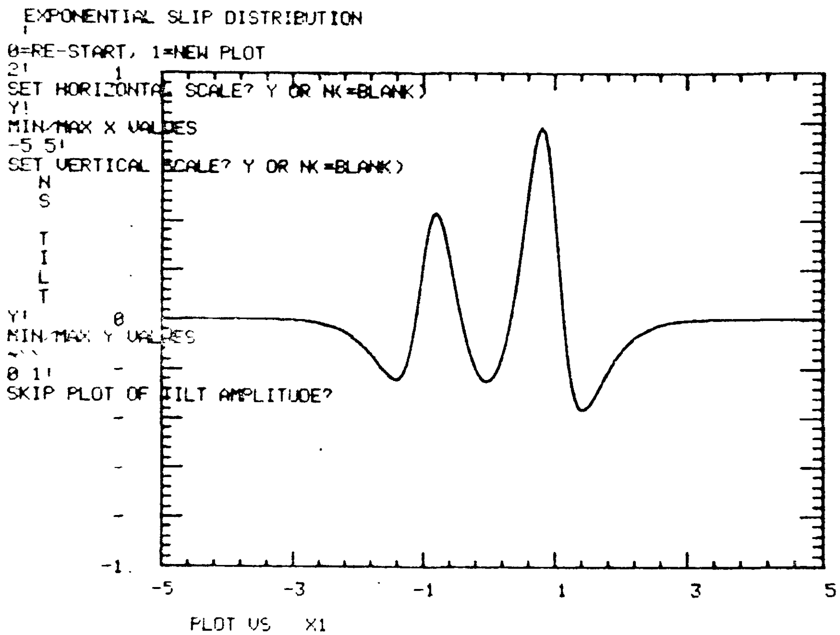
Lines 57-59 have been changed to the exponential slip distribution indicated on the following page. Note that the tilt amplitude is not symmetric (page 1-37). The lack of symmetry is caused by the slip distribution specifying a larger displacement at the right-hand margin than at the left-hand margin of the slip zone.

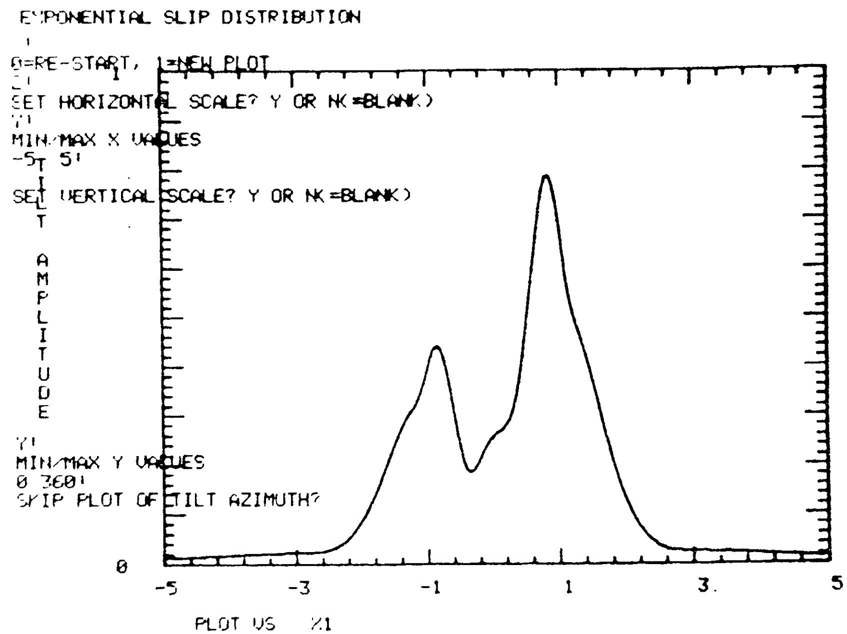
```
50        IF(IFLAG EQ 1)AX2(K)=CNSTNT
51.       IF(IFLAG.EQ 2)AX1(K)=CNSTNT
52        IF(IFLAG EQ 2)AX2(K)=BEGIN+(XINC*(K-1))
53        DO 2 I=1,N2
54        DO 3 J=1,N1
55        C1X1=C3X1FN+((AL/N1)*J) $C1X3=C1X3FN+((W/N2)*(I-1))
56        C2X3=C1X3+(W/N2) $C3X1=C1X1-(AL/N1)
57        ALPHA= 5*SQRT(((I-(N2/2.))**2)+((J-(N1/2 ))**2))
58        U=-UMAX*EXP(ALPHA)
60        X1=AX1(K) $X2=AX2(K)
0 - ^EDIT
```

```
^RUN!
1=PLOT VS  X1, 2=PLOT VS  X2
1!
SPECIFY INITIAL AND FINAL POINTS OF PLOT
-5  5!
SPECIFY VALUE OF X2 THAT PROFILE IS COMPUTED FOR
 5!
SPECIFY C1X1FN,C1X3FN,C2X3FN,C3X1FN
1 0 1 -1!
SPECIFY NUMBER OF POINTS IN PROFILE,
NUMBER OF COLUMNS AND ROWS,
SPATIAL FREQUENCY OF SLIP IN X1 DIRECTION,
SPATIAL FREQUENCY OF SLIP IN X3 DIRECTION,
AND MAXIMUM SLIP
200 10 10 0 0 1!
SPECIFY ANGLE OF FAULT TO NS AXIS, IN DEGREES
0!
MIN /MAX  VALUES OF EW COMPONENT  -5 781E-01    3 740E-01
MIN /MAX  VALUES OF NS COMPONENT  -3 653E-01    7 817E-01
MIN /MAX  VALUES OF AMPLITUDE      1 139E-02    7 877E-01
MIN /MAX  VALUES OF AZIMUTH        4 924      355 236
(NOTE  TILT AMPLITUDES ARE IN MICRORADIANS
AZIMUTH IN DEGREES)
THE FOLLOWING ARE PLOTS OF THE EW AND NS
COMPONENTS OF TILT  AND THE TILT AMPLITUDE
AND AZIMUTH (MEASURED CLOCKWISE FROM NORTH)
0=RE-START, 1=CONTINUE
1!
WRITE PLOT TITLE, 80 CHARACTERS
EXPONENTIAL SLIP DISTRIBUTION!
SET HORIZONTAL SCALE? (Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? (Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
- 1  1!
SKIP PLOT OF EW TILT?
```

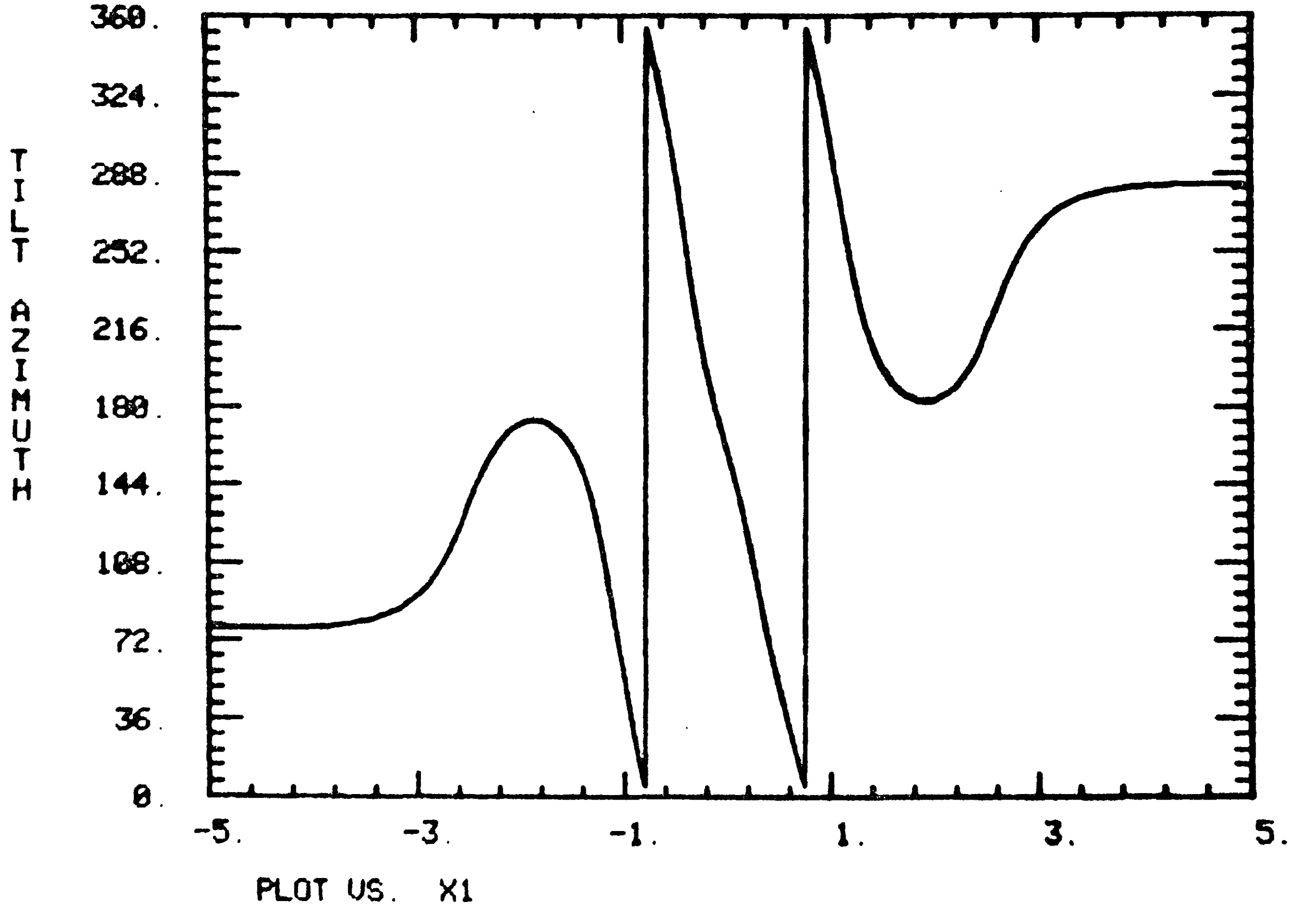

EXPONENTIAL SLIP DISTRIBUTION
0=RE-START, 1=NEW PLOT
1!
SET HORIZONTAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR NK=BLANK)
E
W
T
I
L
T
Y!
MIN/MAX Y VALUES
-1 1!
SKIP PLOT OF EW TILT?
-5.
-3.
-1.
1.
3.
5.
PLOT VS X1


EXPONENTIAL SLIP DISTRIBUTION
0=RE-START, 1=NEW PLOT
2! 1.
SET HORIZONTAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR NK=BLANK)
E
W
T
I
L
T
Y! 0.
MIN/MAX Y VALUES
-1 1!
SKIP PLOT OF NS TILT?
-1.
-5.
-3.
-1.
1.
3.
5.
PLOT VS. X1


EXPONENTIAL SLIP DISTRIBUTION
0=RE-START, 1=NEW PLOT
2!
1
SET HORIZONTAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR NK=BLANK)
NS TILT
Y!
0
MIN/MAX Y VALUES
0 1!
SKIP PLOT OF TILT AMPLITUDE?
-1.
-5
-3
-1
1
3
5
PLOT VS X1


EXPONENTIAL SLIP DISTRIBUTION
0=RE-START, 1=NEW PLOT
1
SET HORIZONTAL SCALE? Y OR NK=BLANK)
MIN/MAX X VALUES
-5 5!
SET VERTICAL SCALE? Y OR NK=BLANK)
TILT AMPLITUDE
Y!
MIN/MAX Y VALUES
0 360!
SKIP PLOT OF TILT AZIMUTH?
0
-5
-3
-1
1
3.
5.
PLOT VS X1

EXPONENTIAL SLIP DISTRIBUTION



DOCUMENTATION OF RCTNGL

```
DELETE(LGO,OUTPUT,RCTNGL)
RCTNGL.
EXIT.
LIBCOPY(GRAPHIC,TXLGO/RR,TXLGO)
LIBCOPY(JDRAT,NPLGO/RR,NPLGO)
DELETE(LGO,OUTPUT,RCTNGL)
RUN76(L)
LINK(F=LGO,F=TXLGO,F=NPLGO,B=RCTNGL)
RCTNGL.
FIN.
EOF
      PROGRAM RCTNGL(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
      COMMON/TVPOOL/TVPOL(8)
      COMMON/TVTUNE/ITUNE(30)
      COMMON/JPLOT/XLT,XRT,YLO,YUP,MAJX,MAJY,KX(2),KY(2),
     1LTITL(8),LG,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
      DIMENSION IFET(8)
      DIMENSION AX1(200),AX2(200),THETA1(20,20),THETA2(20,20)
      DIMENSION TNS(200),TEW(200),TAMP(200),TAZM(200),T(200),AB(20)
      CALL FET(5LTAPE7,IFET,8)
      IFET(2)=IFET(2).OR.0003 0010 0000 0000 0000B
      IFET(8)=IFET(8).OR.4000 0000 0000 0000 0000B
      CALL FET(5LTAPE7,IFET,-8)
  113 CONTINUE
      AAA=1.234E-20 $WRITE(7,8)
    8 FORMAT(*1=PLOT VS. X1, 2=PLOT VS. X3*)
      CALL GETNUM(AB) $IFLAG=AB(1) $WRITE(7,9)
    9 FORMAT(*SPECIFY INITIAL AND FINAL POINTS OF PLOT*)
      CALL GETNUM(AB) $BEGIN=AB(1) $END=AB(2)
      IF(IFLAG.EQ.1)KHOLD=2HX2 $IF(IFLAG.EQ.2)KHOLD=2HX1
      WRITE(7,10)KHOLD $CALL GETNUM(AB) $CNSTNT=AB(1)
   10 FORMAT(*SPECIFY VALUE OF *,A2,* THAT PROFILE IS COMPUTED FOR*)
      WRITE(7,11)
   11 FORMAT(*SPECIFY C1X1FN,C1X3FN,C2X3FN,C3X1FN*)
      CALL GETNUM(AB) $C1X1FN=AB(1) $C1X3FN=AB(2) $C2X3FN=AB(3)
      C3X1FN=AB(4) $WRITE(7,12)
   12 FORMAT(*SPECIFY NUMBER OF POINTS IN PROFILE,*,/,
     1       *NUMBER OF COLUMNS AND ROWS,*,/,
     1       *SPATIAL FREQUENCY OF SLIP IN X1 DIRECTION,*,/,
     1       *SPATIAL FREQUENCY OF SLIP IN X3 DIRECTION,*,/,
     1       *AND MAXIMUM SLIP*)
      CALL GETNUM(AB) $M=AB(1) $N1=AB(2) $N2=AB(3) $FREQX1=AB(4)
      FREQX3=AB(5) $UMAX=AB(6) $WRITE(7,13)
   13 FORMAT(*SPECIFY ANGLE OF FAULT TO NS AXIS, IN DEGREES*)
      CALL GETNUM(AB) $THETA=AB(1) $AL=C1X1FN-C3X1FN
      W=C2X3FN-C1X3FN $CT=COS(THETA*.01745) $ST=SIN(THETA*.01745)
      DO 1 K=1,M
      XINC=(END-BEGIN)/M
      IF(IFLAG.EQ.1)AX1(K)=BEGIN+(XINC*(K-1))
      IF(IFLAG.EQ.1)AX2(K)=CNSTNT
      IF(IFLAG.EQ.2)AX1(K)=CNSTNT
      IF(IFLAG.EQ.2)AX2(K)=BEGIN+(XINC*(K-1))
      DO 2 I=1,N2
      DO 3 J=1,N1
      C1X1=C3X1FN+((AL/N1)*J) $C1X3=C1X3FN+((W/N2)*(I-1))
      C2X3=C1X3+(W/N2) $C3X1=C1X1-(AL/N1)
      U=-UMAX*COS(((C1X1-C3X1FN)/AL)*2.*3.14159*FREQX1)*
     1   COS(((C2X3-C1X3FN)/W)*2.*3.14159*FREQX3)
```

```
      U=-ABS(U)
      X1=AX1(K) $X2=AX2(K)
      CALL CMPTLT(U,X1,X2,C1X1,C1X3,C2X3,C3X1,A,B)
      THETA1(I,J)=A
    5 THETA2(I,J)=B
    2 CONTINUE
      SUM1=SUM2=0.
      DO 4 I=1,N2
      DO 3 J=1,N1
      TNS(K)=THETA1(I,J)+SUM1
      TEW(K)=THETA2(I,J)+SUM2
      SUM1=TNS(K)
    3 SUM2=TEW(K)
    4 CONTINUE
      A=TNS(K) $B=TEW(K)
      TNS(K)=A*CT+B*ST
      TEW(K)=-A*ST+B*CT
      TAMP(K)=SQRT((TEW(K)**2)+(TNS(K)**2))
      IF(TNS(K).EQ.0.)TNS(K)=1.E-20
      TAZM(K)=(ATAN(TEW(K)/TNS(K)))*(180./3.1415926)
      IF(TNS(K).LT.0.)TAZM(K)=TAZM(K)+180.
      IF(TAZM(K).LT.0.)TAZM(K)=TAZM(K)+360.
    1 IF(TAZM(K).GT.360.)TAZM(K)=TAZM(K)-360.
      DO 7 I=1,M
      IF(IFLAG.EQ.1)T(I)=AX1(I)
    7 IF(IFLAG.EQ.2)T(I)=AX2(I)
      TEWMID=TEW(1) $TNSMID=TNS(1) $TAMPMD=TAMP(1) $TAZMMD=TAZM(1)
      TEWMIN=TEWMAX=TEWMID $TNSMIN=TNSMAX=TNSMID
      TAMPMN=TAMPMX=TAMPMD $TAZMMN=TAZMMX=TAZMMD
      DO 100 I=1,M
      IF (TEW(I) .LT. TEWMIN)TEWMIN=TEW(I)
      IF (TEW(I) .GT. TEWMAX)TEWMAX=TEW(I)
      IF (TNS(I) .LT. TNSMIN)TNSMIN=TNS(I)
      IF (TNS(I) .GT. TNSMAX)TNSMAX=TNS(I)
      IF (TAMP(I) .LT. TAMPMN)TAMPMN=TAMP(I)
      IF (TAMP(I) .GT. TAMPMX)TAMPMX=TAMP(I)
      IF(TAZM(I) .LT. TAZMMN)TAZMMN=TAZM(I)
      IF(TAZM(I) .GT. TAZMMX)TAZMMX=TAZM(I)
  100 CONTINUE
      WRITE(7,17)TEWMIN,TEWMAX,TNSMIN,TNSMAX,TAMPMN,TAMPMX,TAZMMN,TAZMMX
   17 FORMAT(*MIN./MAX. VALUES OF EW COMPONENT*,2X,E10.3,2X,E10.3,/,
     1 *MIN./MAX. VALUES OF NS COMPONENT*,2X,E10.3,2X,E10.3,/,
     1 *MIN./MAX. VALUES OF AMPLITUDE*,5X,E10.3,2X,E10.3,/,
     1 *MIN./MAX. VALUES OF AZIMUTH*,3X,F10.3,2X,F10.3,/,
     1 *(NOTE  TILT AMPLITUDES ARE IN MICRORADIANS*,/,
     1 *AZIMUTH IN DEGREES)*)
      WRITE(7,170)
  170 FORMAT(*THE FOLLOWING ARE PLOTS OF THE EW AND NS *,/,
     1  *COMPONENTS OF TILT, AND THE TILT AMPLITUDE*,/,
     1  *AND AZIMUTH (MEASURED CLOCKWISE FROM NORTH).*,/,
     1 *ZERO-START, 1=CONTINUE*)
      CALL GETNUM(AB) $IKFLAG=AB(1)
      IF(IKFLAG.EQ. 0) GO TO 113
      LU=7  $LNLGX=1  $LNLGY=1 $NCLX=2  $NCLY=2
  113 WRITE(7,199)
  199 FORMAT(*WRITE PLOT TITLE, 80 CHARACTERS*)
      READ(7,200)(LTITL(JM),JM=1,8)
  200 FORMAT(8A10)
```

```
      IF(IFLAG.EQ.1)KX(2)=10HX1
      IF(IFLAG.EQ.2)KX(2)=10HX2
      KX(1)=10HPLOT VS.
      KY(1)=10H EW TILT
      KY(2)=10H                 $MAJX=5 $MAJY=10 $LTITL2(1)=1
  900 XLT=T(1)$XRT=T(M)  $YLO=TEWMIN  $YUP=TEWMAX
      WRITE(7,70)
   70 FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
      READ(7,66)CHARAC  $IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GO TO 71
      WRITE(7,72)
   72 FORMAT(*MIN/MAX X VALUES*)
      CALL GETNUM(AB)  $XLT=AB(1)   $XRT=AB(2)
   71 WRITE(7,73)
   73 FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
      READ(7,66)CHARAC  $IF(CHARAC .EQ.1HN.OR.CHARAC.EQ.1H )GO TO 74
      WRITE(7,75)
   75 FORMAT(*MIN/MAX Y VALUES*)
      CALL GETNUM(AB)  $YLO=AB(1)   $YUP=AB(2)
   74 IF(TEWMIN.EQ.TEWMAX)YLO=YUP-1.
      IF(TEWMIN.EQ.TEWMAX)YUP=YUP+1.
      WRITE(7,109)
  109 FORMAT(*SKIP PLOT OF EW TILT?*)
      READ(7,66)IJVAR
   66 FORMAT(A1)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TEW,T,1,M)
      WRITE(7,19) $CALL GETNUM(AB)$IKFLAG=AB(1)
      IF(IKFLAG.EQ.0)GOTO113$IF(IKFLAG.EQ.1)GOTO903
  903 KY(1)=10H NS TILT      $XLT=T(1)$XRT=T(M)
      KY(2)=10H
      YLO=TNSMIN $YUP=TNSMAX $WRITE(7,70) $READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO710
      WRITE(7,72)$CALL GETNUM(AB)$XLT=AB(1)$XRT=AB(2)
  710 WRITE(7,73)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO740
      WRITE(7,75)$CALL GETNUM(AB)$YLO=AB(1)$YUP=AB(2)
  740 IF(YLO.EQ.YUP)AAA=YLO$IF(YLO.EQ.AAA)YUP=YUP+1.
      IF(YLO.EQ.AAA)YLO=YLO-1.
      WRITE(7,910) $READ(7,66)IJVAR
  910 FORMAT(*SKIP PLOT OF NS TILT?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TNS,T,1,M)
      WRITE(7,19)
   19 FORMAT(*0=RE-START, 1=NEW PLOT*)
      CALL GETNUM(AB) $IKFLAG=AB(1)
      IF(IKFLAG.EQ.0)GO TO 113 $IF(IKFLAG.EQ.1)GOTO900
  911 KY(1)=10HTILT AMPLI  $XLT=T(1) $XRT=T(M)
      KY(2)=10HTUDE
      YLO=TAMPMN $YUP=TAMPMX $WRITE(7,70)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO741
      WRITE(7,72)$CALL GETNUM(AB) $XLT=AB(1)$XRT=AB(2)
  741 WRITE(7,73)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO742
      WRITE(7,75)$CALL GETNUM(AB)$YLO=AB(1)$YUP=AB(2)
  742 IF(YLO.EQ.YUP)AAA=YLO$IF(YLO.EQ.AAA)YUP=YUP+1.
      IF(YLO.EQ.AAA)YLO=YLO-1.
      WRITE(7,912) $READ(7,66)IJVAR
  912 FORMAT(*SKIP PLOT OF TILT AMPLITUDE?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAMP,T,1,M)
      WRITE(7,19)$CALL GETNUM(AB)$IKFLAG=AB(1)
```

```
      IF(IKFLAG.EQ.0)GOTO113$IF(IKFLAG.EQ.1)GOTO911
  913 KY(1)=10HTILT AZIMU   $XLT=T(1)$XRT=T(M)
      KY(2)=10HTH           $YLO=TAZMMN  $YUP=TAZMMX
      WRITE(7,70) $READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO743
      WRITE(7,72)$CALL GETNUM(AB)$XLT=AB(1)$XRT=AB(2)
  743 WRITE(7,73)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO744
      WRITE(7,75)$CALL GETNUM(AB)$YLO=AB(1)$YUP=AB(2)
  744 IF(YLO.EQ.YUP)AAA=YLO$IF(YLO.EQ.AAA)YUP=YUP+1.
      IF(YLO.EQ.AAA)YLO=YLO-1.
      WRITE(7,914) $READ(7,66)IJVAR
  914 FORMAT(*SKIP PLOT OF TILT AZIMUTH?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAZM,T,1,M)
      WRITE(7,19) $CALL GETNUM(AB) $IF(AB(1).EQ.0.)GOTO113
      STOP $END
      SUBROUTINE CMPTLT(U,X1,X2,C1X1,C1X3,C2X3,C3X1,T1,T2)
      A1=A2=A3=A4=B1=B2=B3=B4=0.
      X3=0.$C2X1=C1X1$C3X3=C2X3$C4X1=C3X1$C4X3=C1X3
      CALL TILT(U,X1,X2,X3,C2X1,C2X3,A1,B1)
      CALL TILT(U,X1,X2,X3,C1X1,C1X3,A2,B2)
      CALL TILT(U,X1,X2,X3,C3X1,C3X3,A3,B3)
      CALL TILT(U,X1,X2,X3,C4X1,C4X3,A4,B4)
      T1=A1-A2-A3+A4
      T2=B1-B2-B3+B4
      RETURN
      END
      SUBROUTINE TILT(U1,X1,X2,X3,P1,P3,T1,T2)
      R=SQRT((X1-P1)**2+X2**2+(X3-P3)**2)
      RP=R+P3
      T1=(U1/12.5664)*(X2*(X1-P1)*(R*RP-(R+2.*P3)*(2.*R+P3)))/
     1 (R**3*RP**2)
      T2=(U1/12.5664)*(X2**2*(R*RP-(R+2.*P3)*(2.*R+P3))/(R**3*RP**2)
     1 +(R+2.*P3)/(R*RP))
      RETURN
      END
      SUBROUTINE GETNUM(R)
      DIMENSION R(1),L(80)
      READ(7,9)L $ I=J=0
    6 J=J+1 $ N=P=S=0 $ M=F=1
    5 I=I+1 $ IF(I.GT.80)RETURN $ J=L(I) $ K=4
      IF(L.EQ.38)K=2 $ IF(L.GT.27.A.D.LE.36)K=1
      IF(J.EQ.47)K=3 $ K=K+S $ GOTO(1,2,3,5,1,4,3,4)K
    1 N=N*10+D-27                 $ S=4          $ GOTO 5
    2 M=-1                        $ S=4          $ GOTO 5
    3 P=I                         $ S=4          $ GOTO 5
    4 IF(P.NE.0)F=10.**(I-P-1) $ R(J)=N/F*M $ GOTO 6
    9 FORMAT(80R1)
      END
```

PROGRAM SLPPRP

PART 2

DOCUMENTATION FOR PROGRAM SLPPRP

INTRODUCTION:

Creep-related tilt changes can be approximated by modeling the slip distribution on the fault plane as an expanding dislocation loop with spatially and temporally varying displacement as discussed in McHugh and Johnston (1976) and McHugh (1976). The program and model in this section (which will be referred to as Model II) is an extension of the programs XPND/XPND01/XPND02 (which will be referred to as Model I) discussed in McHugh (1976).

Model I reproduces the creep-related tilt changes by computing the quasi-static tilts, at the free surface of an elastic medium, associated with a vertically oriented rectangular dislocation loop (Press, 1965). The slip distribution is constant, at any given instant, across the slip surface; but varies in time. This particular approximation however does not reproduce the entire slip distribution as a function of time as shown in figure 2.1. Notice that when the slip surface encompasses rectangle HIJD the displacement versus time profile seen by a creepmeter is a single point.

Model II (figure 2.2) uses a somewhat more realistic approximation to reproduce the creep-related tilts. In this model, the entire slip versus time function is computed for each rectangle as the dislocation loop expands. With this procedure the effect of the propagating boundary is somewhat independent of the effect of the slip as a function of time at points within the slip zone. Because the medium is elastic the rule of superposition can be used to express the tilt at a point on the free surface. If Θ is the tilt component, *i* is the time after the start of the slip zone expansion, and *j* is the time after initiation of slip at a point *m*, the tilt versus time profiles will appear as in figure 2.3. The tilt, T_k, (either $\frac{\partial w}{\partial x}$ or $\frac{\partial w}{\partial y}$ at a time *k* after the start of the zone expansion) is given by

$$T_k = \sum_{j=1}^{k} \Theta_{j,\,k-j+1} - \sum_{j=1}^{j=k-1} \Theta_{j,\,k-j} \quad .$$

For example, the tilt at t=3 units after the zone starts to expand (figure 2.3) is:

$$T_3 = \Theta_{1,3} + (\Theta_{2,2} - \Theta_{1,2}) + (\Theta_{3,1} - \Theta_{2,1})$$

$$= (\Theta_{1,3} + \Theta_{2,2} + \Theta_{3,1}) - (\Theta_{2,1} + \Theta_{1,2}) \quad .$$

INPUT:

The working arrays, examples of various slip distributions, and notation used are given in McHugh (1976). The expressions used to compute the tilts are given in Press (1965). Unlike Model I, the slip zone expansion velocity need not be constant but can be exponential or some combination of exponential and constant velocities. The positions of the boundaries of the slip zones (strike-slip or dip-slip) can be incremented exponentially (eg. curve P Q R, figure 2.2) and/or linearly (eg. curve TUV, figure 2.2). The slip-time function may be linear or exponential (profiles PP' through VV' - figure 2.2), and will be the same for each slip rectangle (eg. rectangles ABCD, EFGD, and HIJD - figure 2.2). Notice too that curve PQR (figure 2.2) physically corresponds to creep onset-times that decay exponentially, while curve TUV produces a linear creep onset-time distribution. To produce a reference line on the tilt-time displays, the slip zone is fixed at its initial position for the first N units after the start of the computations and fixed in its final position for the last 'NRECPT-N3' units (figure 2.4). The tilts produced by a strike-slip zone and a dip-slip zone may be combined as discussed in McHugh (1976) for Model I. Provision is also made in Model II for one zone to trigger the other's growth.

Model II is intended for use at the Tektronix 4010-1 terminal and is stored at LBL. Once the user is logged onto the 6600 B or C machine with 100 K of core, Model II may be accessed using:

^LOAD, SLPPRP, MCHUGH

The program links automatically to the necessary plotting routines so that the ^LOAD command may be followed immediately by:

^RUN

Examples of the program in operation are given on pages 2-19 to 2-59. The following is a step-by-step list of the input the computer requires.

1) 1 = Zone Expands, 2 = Zone Contracts

Entering a 1 causes the zone to grow from its initial to its final values; a 2 causes the reverse to occur (ie. the zone appears to 'collapse').

2) 1/2 = Slip incremented exponentially/linearly
Strike-Slip/Dip-Slip

Enter 2 numbers (eg. 1 (space) 1). The first number causes the displacement on the strike-slip zone to be incremented linearly (2) or exponentially (1), while the second number controls the slip-time function on the dip-slip zone.

3) 0 = Increment corners separately

1/2 = Increment all corners exponentially/linearily

Strike - slip/Dip-slip

Enter 2 numbers (eg. 1 (space) 0). The corners of the strike-slip zone (controlled by the first number) and dip-slip zone (second number) will be incremented exponentially (1) or linearly (2); or the corners may be controlled individually by entering a zero. If both numbers are a 1 or a 2, the program skips to part 4 below. If a zero is entered the computer responds:

a) 1 = Variable incremented exponentially

2 = Variable incremented linearly

b) activated if the second number is zero

D1X1, D1X3, D2X3, D3X1

Enter 4 numbers.

c) activated if the first number is zero

C1X1, C1X3, C2X3, C3X1

Enter 4 numbers.

4) 'U1>0' = left-lateral strike-slip

'U3>0' = 'X2>0' side down

U1IN, U1FN, U3IN, U3FN

Enter 4 numbers for the initial (IN) and final (FN) slip values. The slip will be incremented between its initial and final values in a linear or exponential fashion depending upon which options were selected in part 2 above.

5) 'TRIGGER' option desired?

Enter a yes (Y) or no (N). This option is the same as discussed in programs XPND/XPND01/XPND02 (McHugh, 1976). If N is entered, the program moves to part 6 below.

a) 0 = D(I1) > C(I2), 1 = D(I1) < C(I2)

The '0' option is equivalent to the trigger option in programs XPND/XPND01, the '1' option to program XPND02 (McHugh, 1976). When the corner designated by D(I1) is greater than (0) or less than (1) the corner designated by C(I2), the appropriate zone is triggered (ie. starts 'growing').

b) 0 = Strike-slip/1 = Dip-slip zone triggered

Entering a zero causes the dip slip zone to trigger the strike-slip zone; a 1 causes the strike-slip zone to trigger the dip-slip zone.

c) Specify I1 and I2

Enter 2 numbers corresponding to numbers of the C and D arrays as shown in part 5a above. The C and D arrays are given in McHugh (1976).

6)

a) activated if U1IN and U1FN in part 4 are both non-zero

C1X1IN, C1X3IN, C2X3IN, C3X1IN, C1X1FN, C1X3FN, C2X3FN, C3X1FN

Enter the 8 coordinates required for the strike-slip zone.

b) activated if U3IN and U3FN in part 4 are both non-zero.

D1X1IN, D1X3IN, D2X3IN, D3X1IN, D1X1FN, D1X3FN, D2X3FN, D3X1FN

Enter the 8 coordinates required for the dip-slip zone.

7) Enter station coordinates - (X1, X2)

Enter the X1 and X2 coordinates of the station.

8) Specify 2 corners of dislocation surface for display

1 = D1X1 = D2X1	2 = D1X3 = D4X3
4 = D2X3 = D3X3	5 = D3X1 = D4X1
9 = C1X1 = C2X1	10 = C1X3 = C4X3
12 = C2X3 = C3X3	13 = C3X1 = C4X1

Enter 2 numbers to display the position of the desired corners as a function of time.

9) To display slip as a function of time

Enter zone index, Caution -

Index must be between 1 and (INDEX)

(INDEX) is the total number of slip rectangles used in the computation. Enter the number corresponding to the slip zone desired. This number will cause the slip as a function of time for that slip zone to be displayed in the output.

10) Theta = angle between strike of fault

And north = (NUMBER) Degrees

No response required. The value of theta is printed.

11) MIN/MAX values of EW component (numerical values)

MIN/MAX values of NS component (" ")

MIN/MAX values of amplitude (" ")

MIN/MAX values of azimuth (" ")

(Note: Tilt amplitudes are in microradians, azimuth in degrees)

The following are plots of the EW and NS components of tilt, and the tilt amplitude and azimuth (measured clockwise from north).

0 = Re-start, 1 = Continue

Enter a zero or one as desired. A zero will cause all the output to be skipped and the program to be re-started. The following occur if a 1 is entered.

12) Write plot title, 80 characters

Enter up to 80 alphanumeric characters.

13) The procedure for scaling the graphs in the output are the same as discussed in McHugh (1976) and will not be discussed here.

14) The graphs are displayed automatically once the scaling and 'skip' options are selected.

OUTPUT:

The plots occur in the following order:

1) EW tilt component versus time
2) NS tilt component versus time
3) Theta 2 versus time
4) Theta 1 versus time
5) Strike-slip versus time
6) Dip-slip versus time
7) Tilt amplitude versus time
8) Tilt azimuth versus time
9) Corner 1 versus time
10) Corner 2 versus time
11) Strike-slip area versus time
12) Dip-slip area versus time

'Time' in parts 1, 2 and 7 through 12 is the time after the start of the slip zone expansion, and corresponds to the individual rectangles for which the tilts, $\Theta^m_{i,j}$, are computed, 'Time' in parts 3 through 6 is the time after the start of slip on the individual slip zone. Theta 1 and Theta 2 are $\frac{\partial w}{\partial x_1}$ and $\frac{\partial w}{\partial x_2}$ in Press' (1965) notation and correspond to $\Theta^m_{i,j}$ in figure 2.3. Corners 1 and 2 correspond to the corners selected in part 8 of the Input section.

After the dip-slip area has been displayed, or if the '0 = Re-start' option is selected, the C array is printed out as follows:

```
C(1)      C(2) . . .    C(6)
 :                       :
C(30)      . . . .      C(36)
```

The C array contains the initial and final values of the coordinates and slip (McHugh, 1976). This array is followed by:

ICRNR(1) . . . . ICRNR(8)
⋮ ⋮
ICRNR(9) . . . . ICRNR(16)

IT(1) . . . IT(4) ITIME

KFLAGS KFLAGD LFLAG MFLAG IFLAG

NRECPT N N1 N2 N3

Definitions of these variables will be found in Appendix A (page 2- 11). A list of options for re-starting the program follows these variables:

1 = Re-start with all new values

2 = Re-start with new strike-slip value and zone coordinates

3 = Re-start with new dip-slip value and zone coordinates

4 = Re-start with new strike-slip value only

5 = Re-start with new dip-slip value only

6 = Re-start with new tiltmeter coordinates only

7 = Stop

Entering a 1 through 6 will re-start the program with the changes specified above. The computer will request the information it needs. A 7 causes the program to stop.

RESULTS AND DISCUSSION:

Examples of the program operation start on page 2-19, and the program listing starts on page 2- 60. Appendix A (page 2- 11) lists some of the flags and their definitions.

The tilts from Model II are nearly the same as those from Model I if 1) the slip is constant in time (ie. U1IN = UIFN and U3IN = U3FN) and 2) the corners of the slip zone are incremented linearly. If the slip is constant in time, but the corners are incremented exponentially, the tilt-time curve (from Model II) will be compressed in time accordingly. If the slip is not constant in time and the corners are incremented exponentially and/or linearly, the tilt-time curve (Model II) will be reduced in total amplitude and stretched or compressed compared to Model I. The reduction in amplitude occurs because the slip is no longer constant over the entire slip zone but rather varies from near-zero at the edges to its maximum in the center. When the rectangle (slip zone) has stopped expanding, the slip at its margins is still increasing to the final slip value; consequently the tilt approaches its final value more slowly than the corresponding case in Model I.

Discontinuities may appear if the 'trigger' parameters allow the triggered and triggering zones to be meshed improperly (eg. if the slip on the triggered zone increases from its initial to final value discontinuously). If the tilt-time curves are continuous, the discontinuity near the start of the tilt azimuth curve is produced by zeroes in the initial values of the north-south tilt component. This particular discontinuity is an artifact of the computational procedure and does not represent an actual discontinuity in the tilt field.

STRTCH

Program STRTCH is a variation of SLPPRP that computes the maximum tilt amplitude seen at a station (with coordinates X1, X2) as a function of depth to the lower edge of the slip zone. STRTCH requires 100k of core and is accessed by: ^LOAD, STRTCH, MCHUGH. The geometry and notation in, operation of, and input to STRTCH is the same as SLPPRP. The only new parameters are DEPTHL (the depth to the lower boundary of the slip zone) and DELTAX (the amount by which DEPTHL is incremented). The initial and final values of the lower boundary are fixed for each computation (ie. C(4) = C(6) = C(12) = C(14) = C(20) = C(22) = C(28) = C(30) = DEPTHL). Although the lower edge is fixed, the other boundaries are free; and the tilt components at the station are computed in the same fashion as in SLPPRP. The maximum amplitude of the waveform is computed for each value of DEPTHL. The output is a display of the maximum tilt amplitude in microradians versus depth to the lower boundary of the slip zone in kilometers (examples are provided on pages 2-70 to 2-92, the program listing starts on page 2-93).

Note: Although the program requres specification of C2X3 and D2X3, the position of the lower boundary is actually determined by DELTAX and the maximum value of KTEST (line 217). Further, the position of the lower boundaries of the strike-slip and dip-slip zones are set equal to one another (lines 79-80). Each cycle through the program requires approximately 160 c.u.'s.

APPENDIX A

Flags Used In SLPPRP

Iflag: determines whether zone expands or contracts

Kflags, Kflagd: determines whether strike-slip (kflags) and dip-slip (kflagd) are incremented linearly or exponentially as a function of time

Lflag: determines whether all the strike-slip zone boundaries are incremented linearly or exponentially or are controlled separately

Mflag: acts in the same fashion as Lflag, but is used to control the dip-slip zone expansion

IT(1): determines whether triggered zone starts expanding when D(IT(3)) > C(IT(4)) or when D(IT(3)) < C(IT(4))

IT(2): determines whether the strike-slip or the dip-slip zone is triggered

IT(3), IT(4): determines which coordinates cause triggering (see IT(1))

ITIME: time, after start of triggering zone's expansion, that triggered zone's expansion starts; if not used, ITIME will have a value of 99999.

ICRNR: controls growth of zone coordinates (exponential = 1, linear = 2), index of ICRNR corresponds to index of D array

NRECPT: number of subrectangles used in computations (corresponds to number of time points after computations begin)

NSLPPT: number of time points in slip versus time profile

N, N1, N2, N3: indices used internally to scale zone expansion (see figure 2.4)

C and D arrays: defined in Appendix A of McHugh (1976); C is the array of initial and final values, D is the working array.

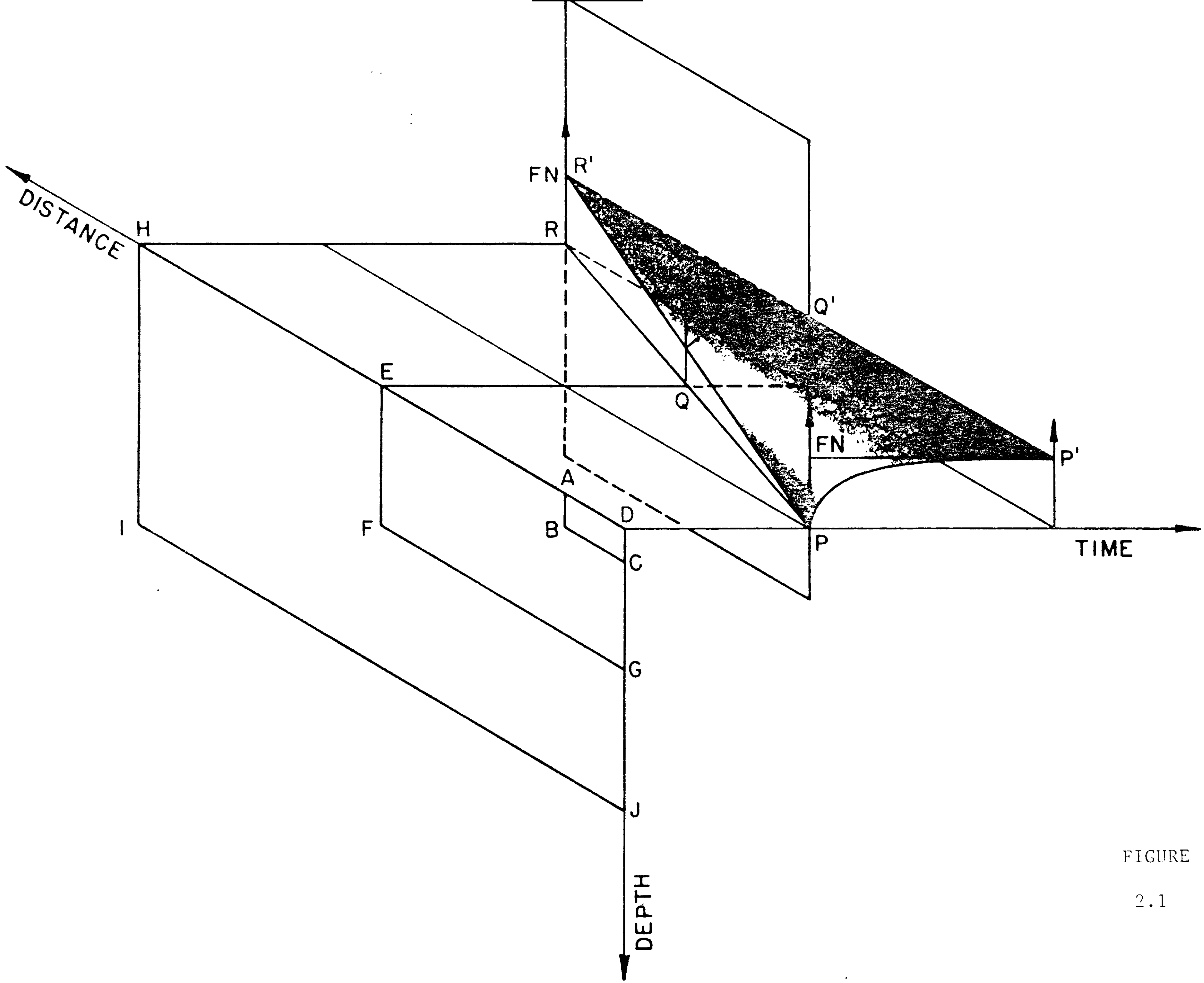
FIGURE CAPTIONS

2.1) Model I: Model of creep event used to predict tilt versus time profile. Dislocation loop expands from ABCD to HIJD. Corner A moves to position H linearly in time (indicated by line PQR). Only a point at position D 'sees' the entire slip versus time profile (PP'). Other points (eg. E and H) 'see' only a portion of the slip-time profile (QQ' and RR'). FN is the final slip value.

2.2) Model II: Model of creep event used to predict tilt versus time behavior. Dislocation loop expands in 'Distance-Depth' plane from ABCD to EFGD to HIJD. Curve PQR indicates that the position of the boundaries AB, EF, and HI vary exponentially in time (ie. the creep onset-times are distributed exponentially). Curve TUV indicates that the position of the lower boundary varies linearly with time. Curves PP' through VV' are the slip versus time profiles for the dislocation loop positions indicated. The slip as a function of position at a given instant can be determined by noting the slip, on profiles PP' through RR' or TT' through VV', along a constant-time line.

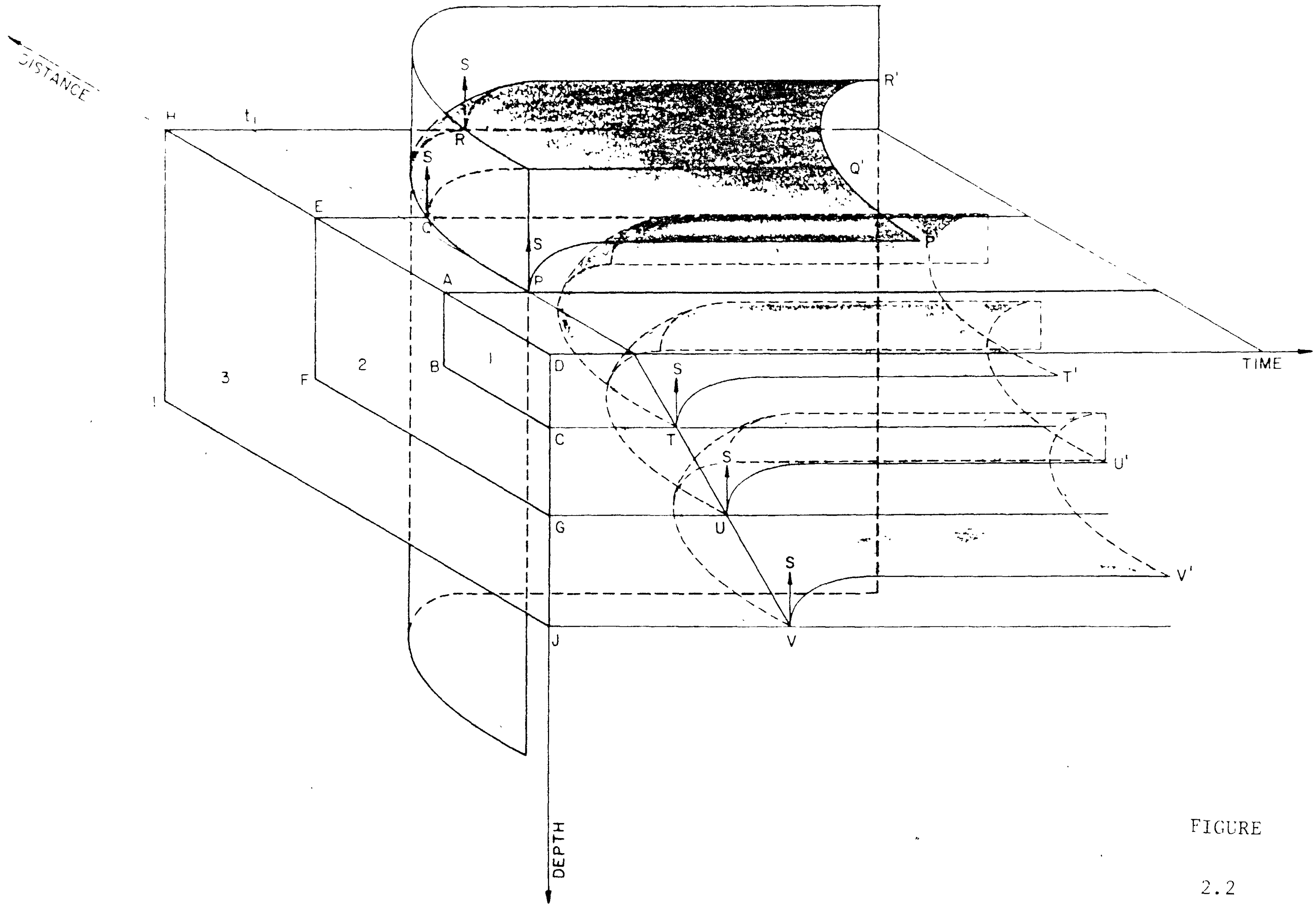
2.3) Model II: Tilts, $\Theta^{m}_{i,j}$, generated by expansion of dislocation. Subscript 'i' denotes time after start of rupture, 'j' denotes time after start of slip at position 'm'. Time lines designated by 'T_i' indicate tilts produced by spatially variable slip distribution; time lines designated by 'ΔT_i' indicate tilts produced by temporally variable slip distribution.

2.4) Model II: Slip versus time profiles used in program SLPPRP. The slip and zone coordinates are constant for the first N units after the computations begin. The slip zone coordinates are fixed at their final positions for computations steps N3 through NRECPT.



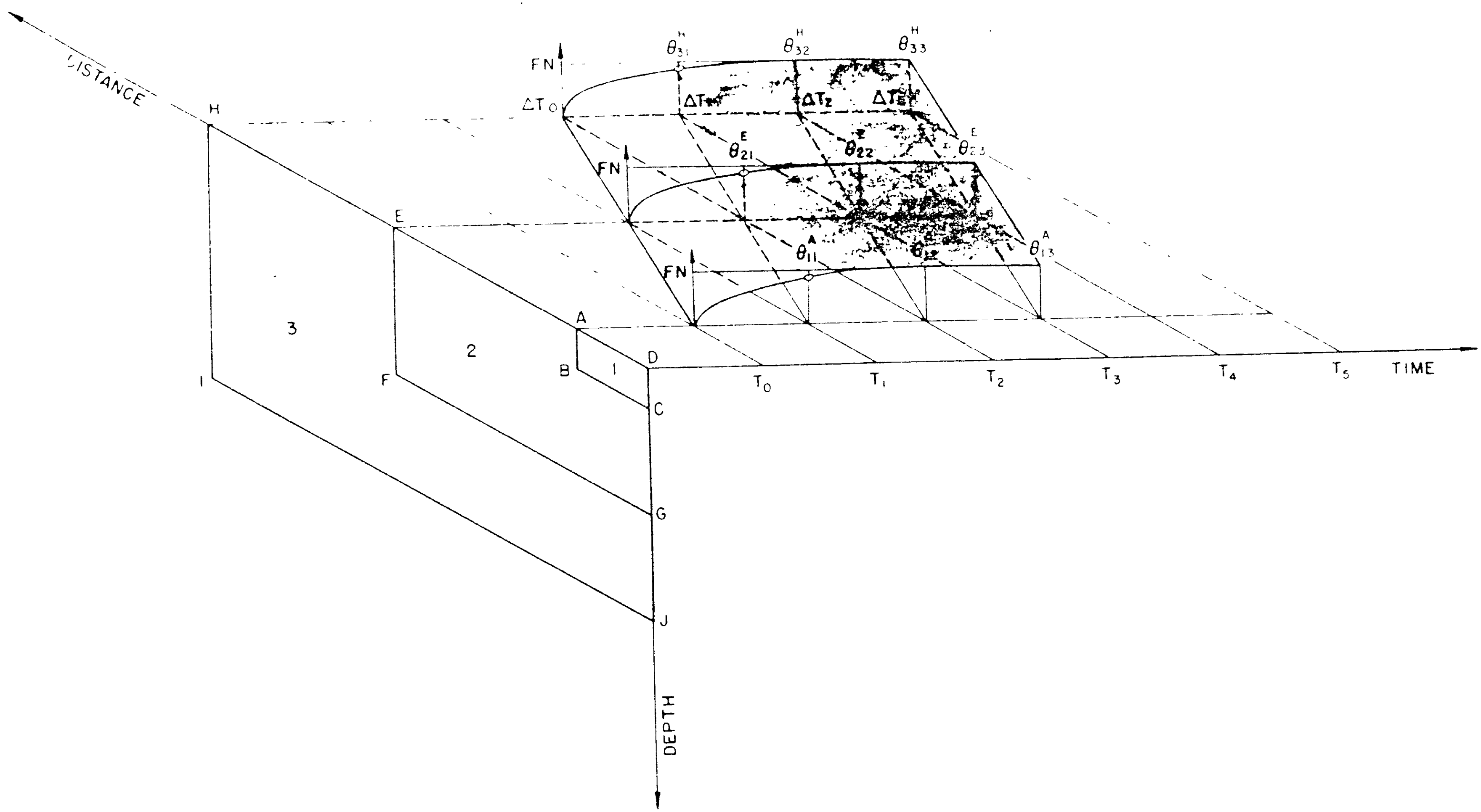
FIGURE

2.1



FIGURE

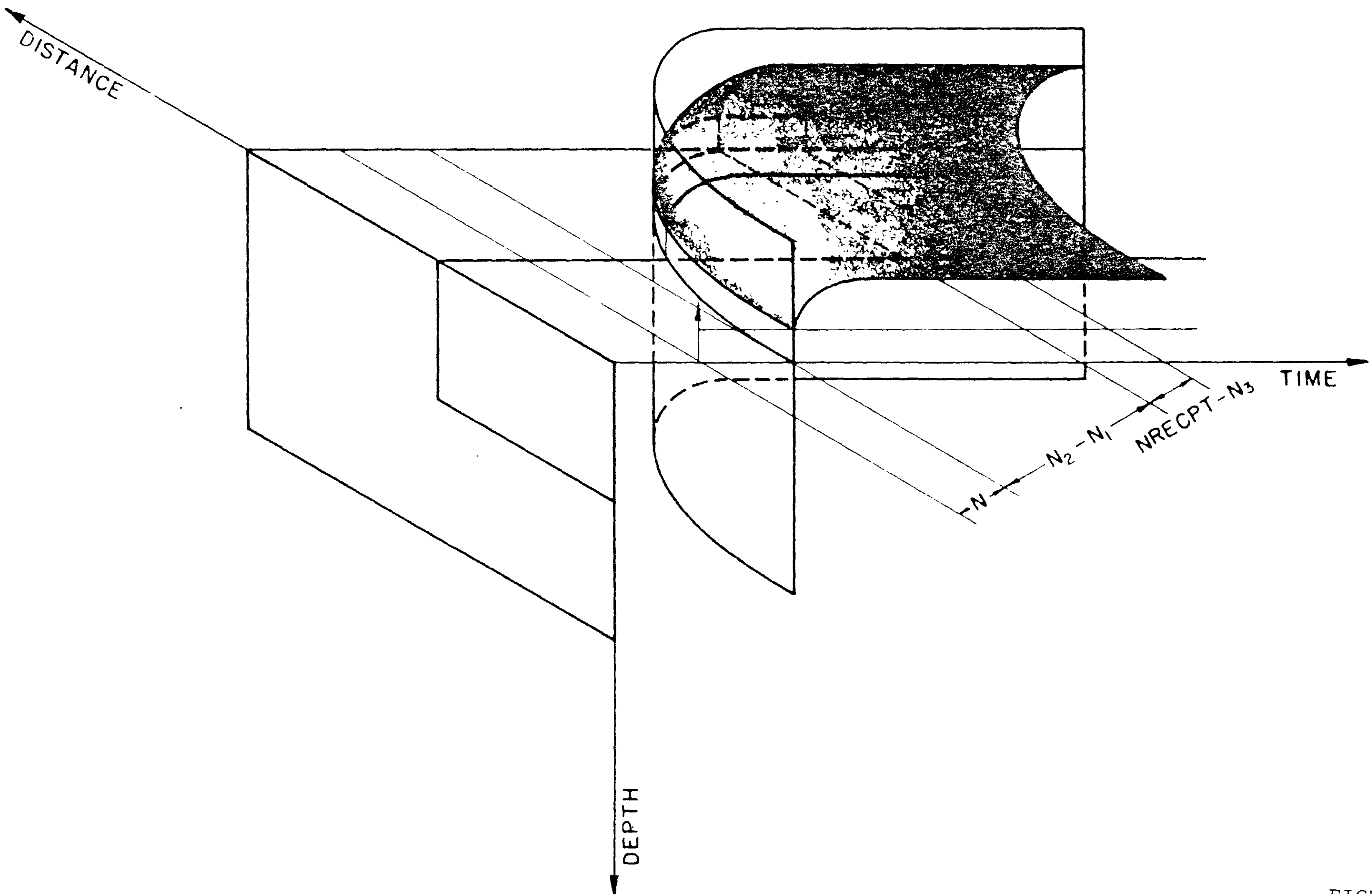
2.2



DISTANCE
TIME
DEPTH
FN
ΔT0
ΔT1
ΔT2
θ31 H
θ32 H
θ33 H
θ21 E
θ22 E
θ23 E
θ11 A
θ13 A
H
E
A
B
C
D
F
G
I
J
1
2
3
T0
T1
T2
T3
T4
T5

FIGURE

2.3



FIGURE

2.4

REFERENCES

McHugh, Stuart and Malcolm J. S. Johnston. Some short period nonseismic tilt perturbations and their relation to episodic slip on the San Andreas fault in central California, J.G.R., in press, 1976.

McHugh, Stuart. Documentation of programs that compute 1) quasi-static tilts produced by an expanding dislocation loop in an elastic and viscoelastic material, and 2) surface shear stresses, strains and shear displacements produced by screw dislocations in a vertical slab with modulus contrast, U.S. Geol. Survey Open-File Report, 76-484

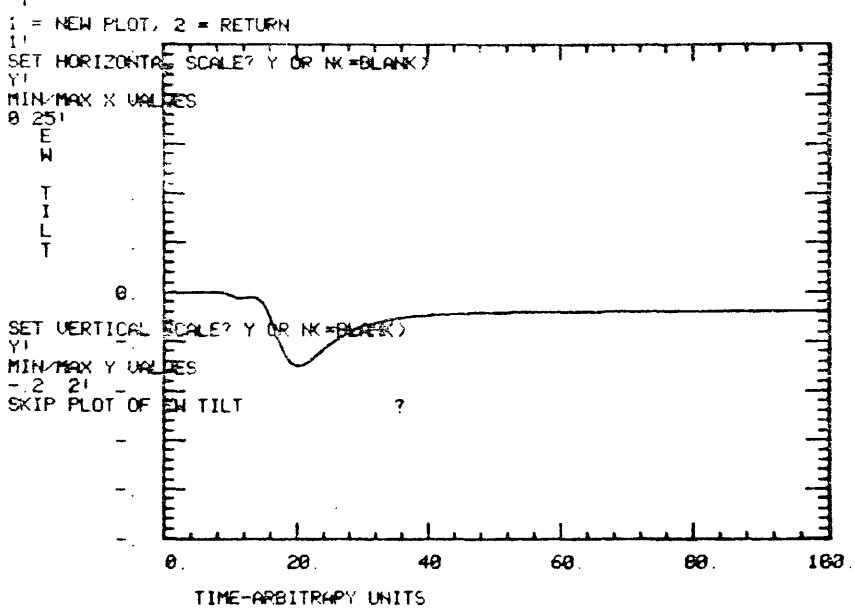
Press, Frank, Displacements, strains, and tilts at teleseismic distances, J.G.R., 70, 2395-2412, 1965.

EXAMPLES OF SLPPRP

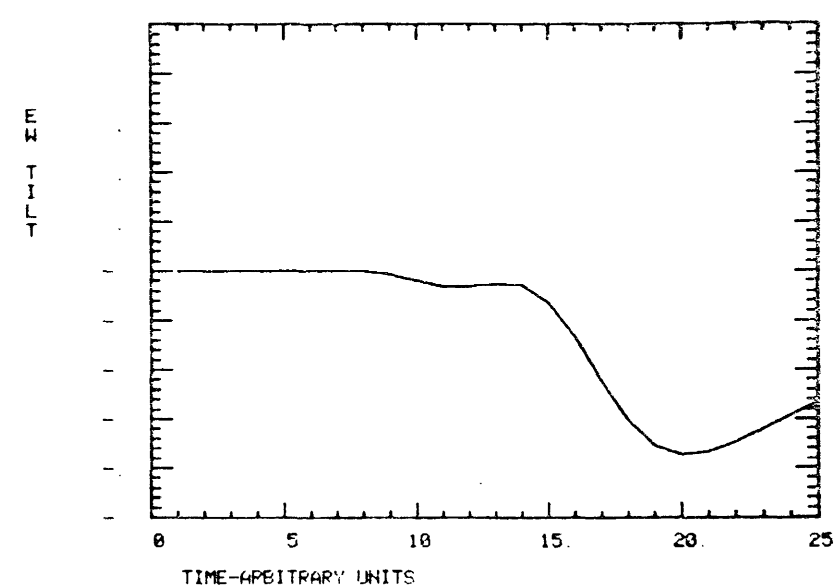
```
^LOAD,SLPPRP!
LOAD COMPLETE, ENTERING ^EDIT
OK - ^EDIT
^RUN!
1=ZONE EXPANDS, 2=ZONE CONTRACTS
1!
1/2=SLIP INCREMENTED EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
1 1!
0  =INCREMENT CORNERS SEPARATELY
1/2=INCREMENT ALL CORNERS EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
1 1!
'U1>0' = LEFT-LATERAL STRIKE-SLIP
[illegible] 'X2>0' SIDE DOWN
[illegible] U3IN, U3FN
[illegible] 0 0!
'TRIGGER' OPTION DESIRED?
N!
C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN
0 0 0 0 2 0 1 0!
ENTER STATION COORDINATES--(X1,X2)
1  .5!
SPECIFY 2 CORNERS OF DISLOCATION SURFACE FOR DISPLAY
 1=D1X1=D2X1            2=D1X3=D4X3
 4=D2X3=D3X3            5=D3X1=D4X1
 9=C1X1=C2X1           10=C3X3=C4X3
12=C2X3=C3X3           13=C3X1=C4X1
9 12!
TO DISPLAY SLIP AS A FUNCTION OF TIME
ENTER ZONE INDEX,  CAUTION---
INDEX MUST BE BETWEEN 1 AND  100
50
```

```
THETA=ANGLE BETWEEN STRIKE OF FAULT
AND NORTH =      45.000 DEGREES
MIN/MAX VALUES OF EW COMPONENT  -1.493E-01    0.
MIN/MAX VALUES OF NS COMPONENT  -1.292E-01    4.127E-02
MIN/MAX VALUES OF AMPLITUDE      0            1.703E-01
MIN/MAX VALUES OF AZIMUTH        0          315.000
(NOTE--TILT AMPLITUDES ARE IN MICRORADIANS,
AZIMUTH IN DEGREES)
THE FOLLOWING ARE PLOTS OF THE EW AND NS COMPONENTS
OF TILT, AND THE TILT AMPLITUDE
AND AZIMUTH (MEASURE CLOCKWISE FROM NORTH).
0=RE-START, 1=CONTINUE
1!
WRITE PLOT TITLE, 80 CHARACTERS
LATERAL AND VERTICAL MIGRATION/CONSTANT SLIP!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
 !
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
-.5 .5!
SKIP PLOT OF EW TILT              ?
```

LATERAL AND VERTICAL MIGRATION/CONSTANT SLIP

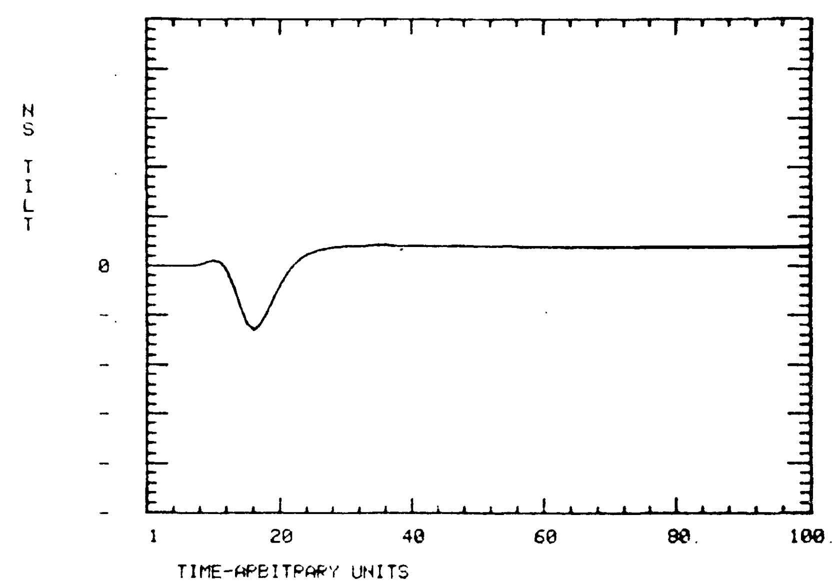


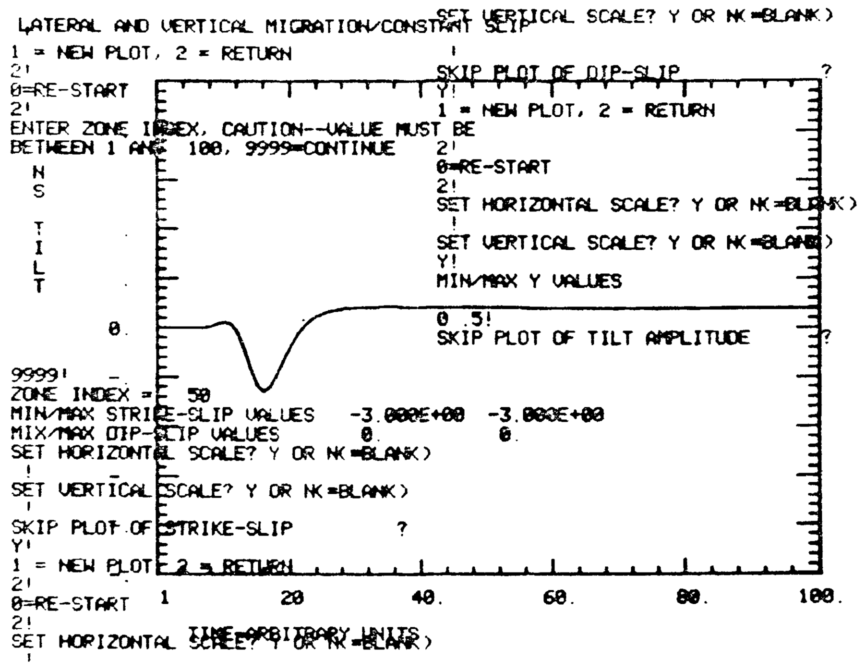
LATERAL AND VERTICAL MIGRATION/CONSTANT SLIP

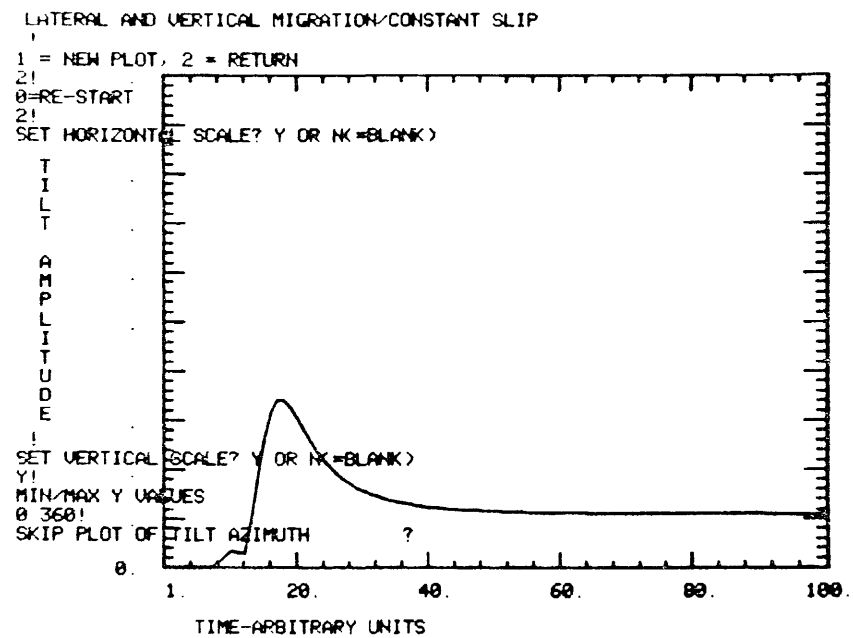


```
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y\\
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
- 5  5!
SKIP PLOT OF NS TILT              ?
```

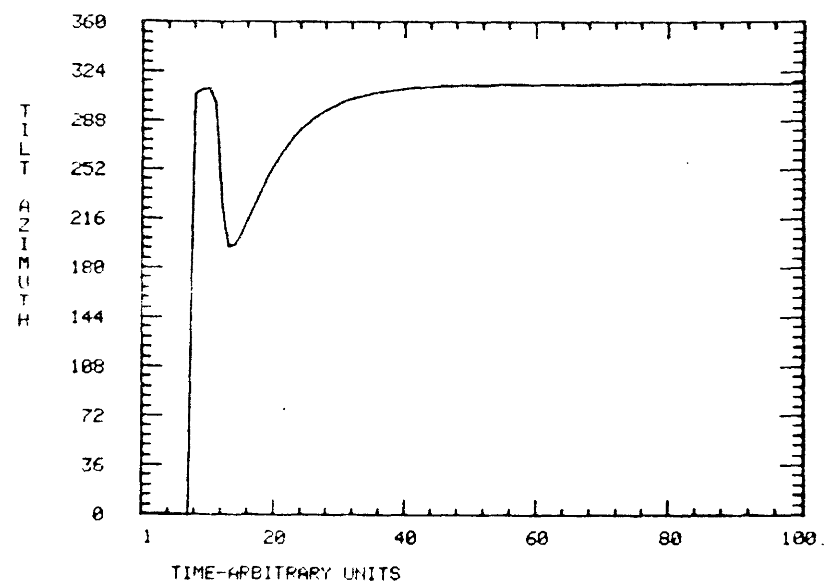
LATERAL AND VERTICAL MIGRATION/CONSTANT SLIP



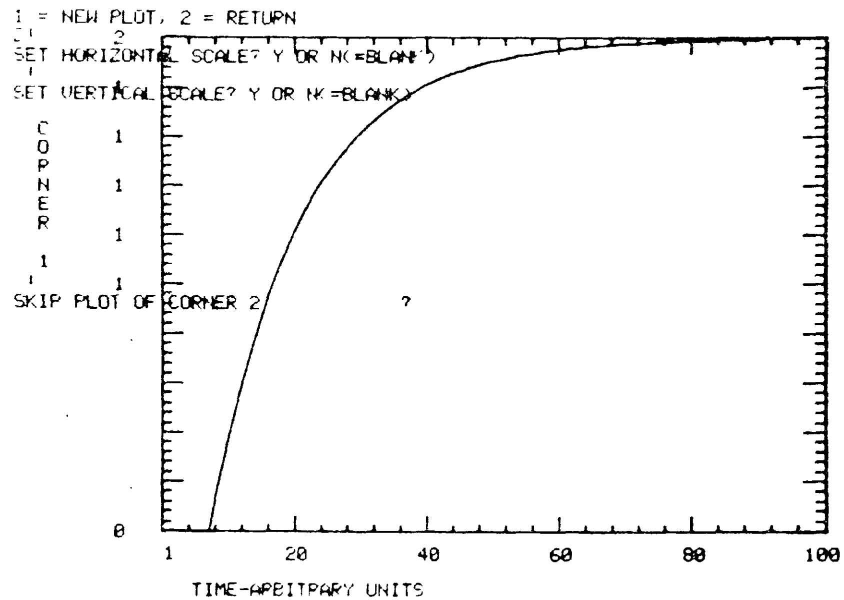

LATERAL AND VERTICAL MIGRATION/CONSTANT SLIP
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
ENTER ZONE INDEX, CAUTION--VALUE MUST BE
BETWEEN 1 AND 100, 9999=CONTINUE
NS TILT
0.
9999!
ZONE INDEX = 50
MIN/MAX STRIKE-SLIP VALUES -3.000E+00 -3.000E+00
MIX/MAX DIP-SLIP VALUES 0. 0.
SET HORIZONTAL SCALE? Y OR NK=BLANK)
SET VERTICAL SCALE? Y OR NK=BLANK)
SKIP PLOT OF STRIKE-SLIP ?
Y!
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
SET HORIZONTAL SCALE? Y OR NK=BLANK)
SET VERTICAL SCALE? Y OR NK=BLANK)
SKIP PLOT OF DIP-SLIP ?
Y!
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
SET HORIZONTAL SCALE? Y OR NK=BLANK)
SET VERTICAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX Y VALUES
0 .5!
SKIP PLOT OF TILT AMPLITUDE ?
1 20 40. 60. 80. 100.
TIME-ARBITRARY UNITS

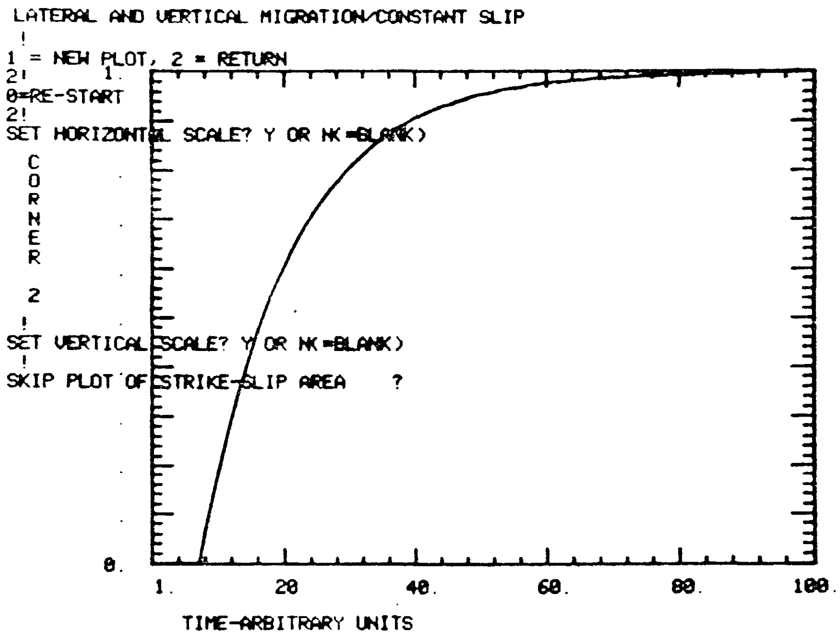

LATERAL AND VERTICAL MIGRATION/CONSTANT SLIP
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
SET HORIZONTAL SCALE? Y OR NK=BLANK)
TILT AMPLITUDE
SET VERTICAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX Y VALUES
0 360!
SKIP PLOT OF TILT AZIMUTH ?
0.
1. 20. 40. 60. 80. 100.
TIME-ARBITRARY UNITS

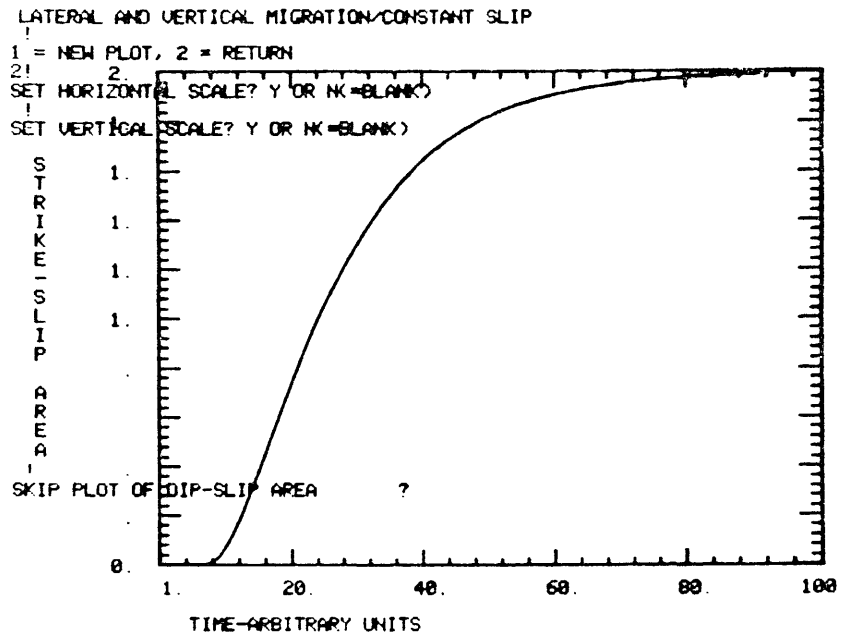
LATERAL AND VERTICAL MIGRATION/CONSTANT SLIP



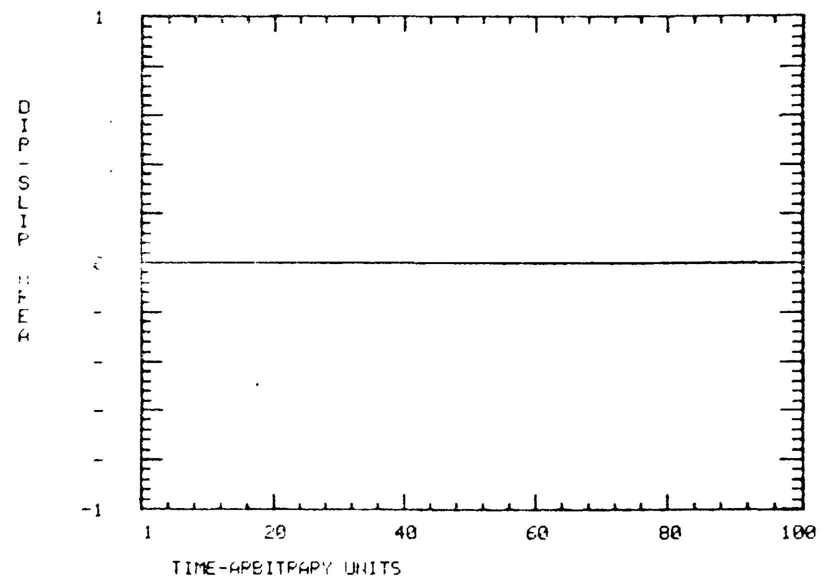
LATERAL AND VERTICAL MIGRATION/CONSTANT SLIP




LATERAL AND VERTICAL MIGRATION/CONSTANT SLIP
1 = NEW PLOT, 2 = RETURN
0=RE-START
SET HORIZONTAL SCALE? Y OR NK=BLANK)
CORNER 2
SET VERTICAL SCALE? Y OR NK=BLANK)
SKIP PLOT OF STRIKE-SLIP AREA ?
1.
0.
1. 20 40. 60. 80. 100.
TIME-ARBITRARY UNITS


LATERAL AND VERTICAL MIGRATION/CONSTANT SLIP
1 = NEW PLOT, 2 = RETURN
SET HORIZONTAL SCALE? Y OR NK=BLANK)
SET VERTICAL SCALE? Y OR NK=BLANK)
STRIKE-SLIP AREA
SKIP PLOT OF DIP-SLIP AREA ?
2.
1.
0.
1. 20. 40. 60. 80. 100.
TIME-ARBITRARY UNITS

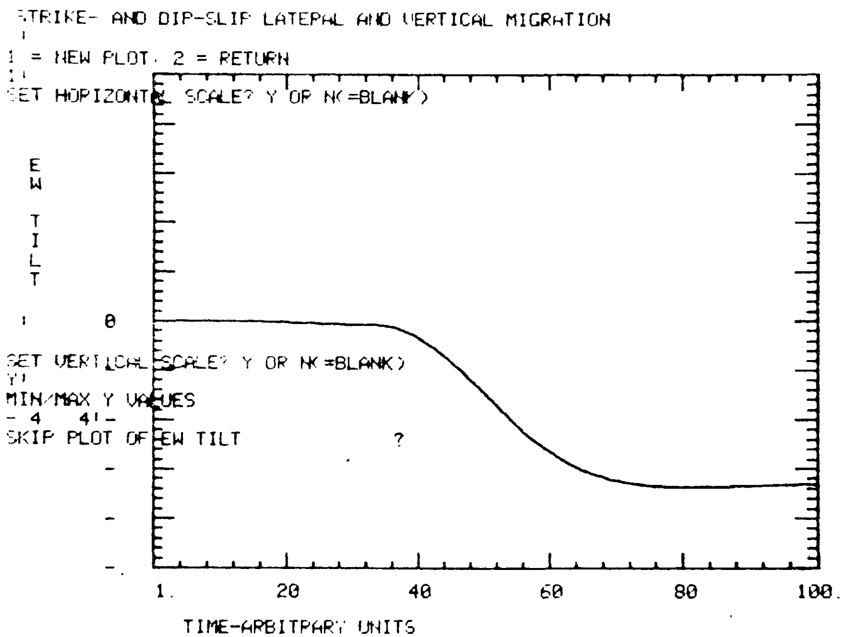
LATERAL AND VERTICAL MIGRATION/CONSTANT SLIP

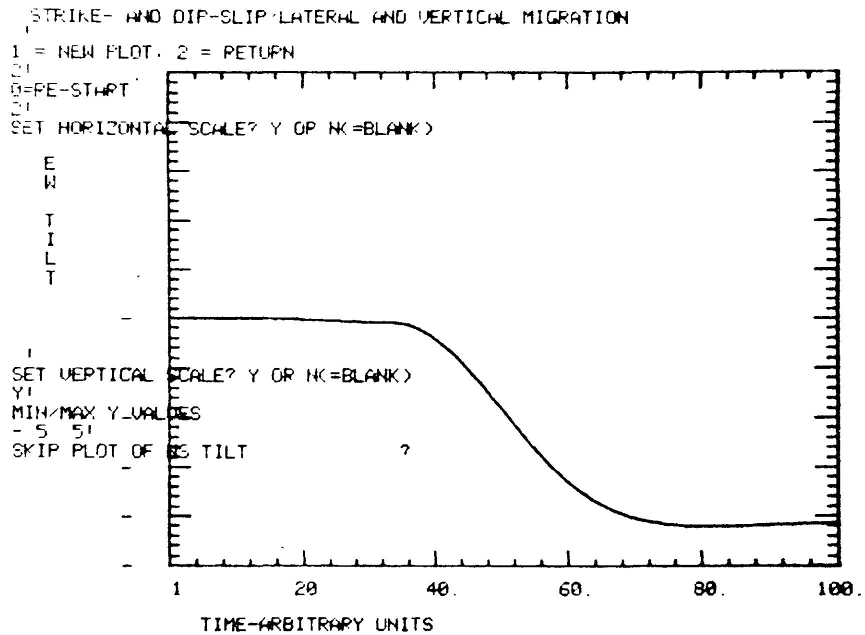


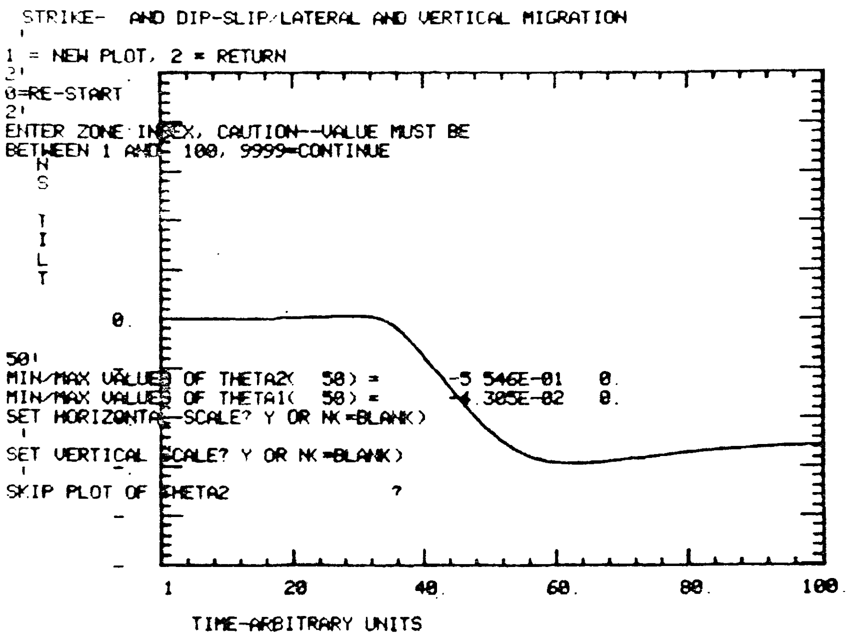

```
1 = NEW PLOT, 2 = RETURN
2
0=RE-START
2
     0          0          0          0          0          0
     0          0          0          0          0          0
     0          0          0          0          0          0
     0          0          0          0.         0          0
     2 000      0          2 000      1 000      0          1.000
     0          0         -3 000     -3 000      0          0

     1      1      1      1      1      1      1      1
     1      1      1      1      1      1      1      1

         5          5          5          5      99999
         1          1          1          1          1
       100          7          8         91         92
1=RE-START WITH ALL NEW VALUES
2=RE-START WITH NEW STRIKE-SLIP VALUE AND ZONE COORDINATES
3=RE-START WITH NEW DIP-SLIP VALUE AND ZONE COORDINATES
4=RE-START WITH NEW STRIKE-SLIP VALUE ONLY
5-RE-START WITH NEW DIP-SLIP VALUE ONLY
6=RE-START WITH NEW TILTMETER COORDINATES ONLY
7=STOP
1
```

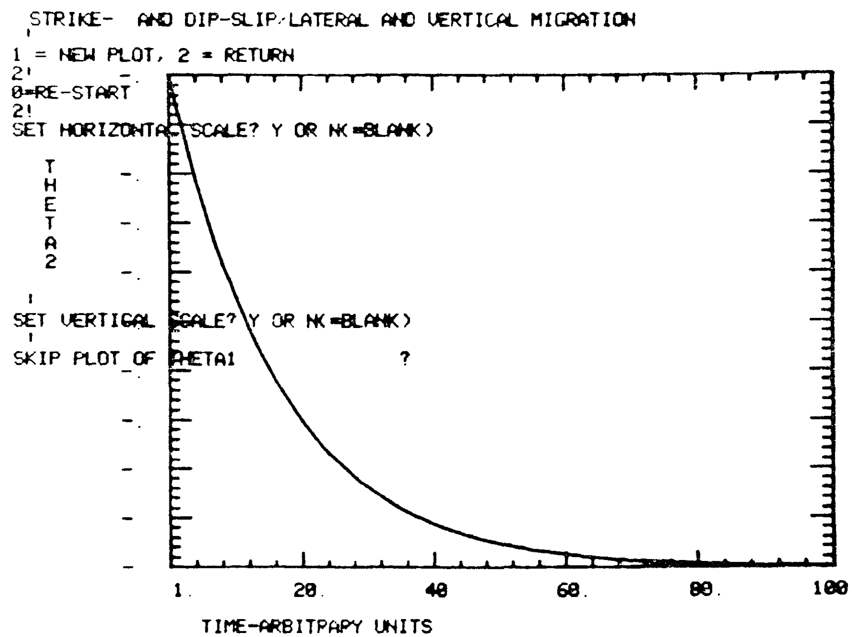
```
!
1=ZONE EXPANDS, 2=ZONE CONTRACTS
1!
1/2=SLIP INCREMENTED EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
1 1!
0  =INCREMENT CORNERS SEPARATELY
1/2=INCREMENT ALL CORNERS EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
0 0!
1=VARIABLE INCREMENTED EXPONENTIALLY
2=VARIABLE INCREMENTED LINEARLY
D1X1, D1X3, D2X3, D3X1
1 2 1 2!
C1X1, C1X3, C2X3, C3X1
2 1 21\1 1!
'U1>0' = LEFT-LATERAL STRIKE-SLIP
'U3>0' = 'X2>0' SIDE DOWN
U1IN, U1FN, U3IN, U3FN
0 -3 0 1!
'TRIGGER' OPTION DESIRED?
Y!
0 = D(I1) > C(I2),  1 = D(I1) < C(I2)
0!
0 = STRIKE-SLIP/1 = DIP-SLIP ZONE TRIGGERED
1!
SPECIFY I1 AND I2
9 5!
C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN
0 0 0 0 2 0 1 0!
D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,D3X1FN
 5 0 0  5  1 5 0 1  5!
ENTER STATION COORDINATES--(X1,X2)
1  5
```

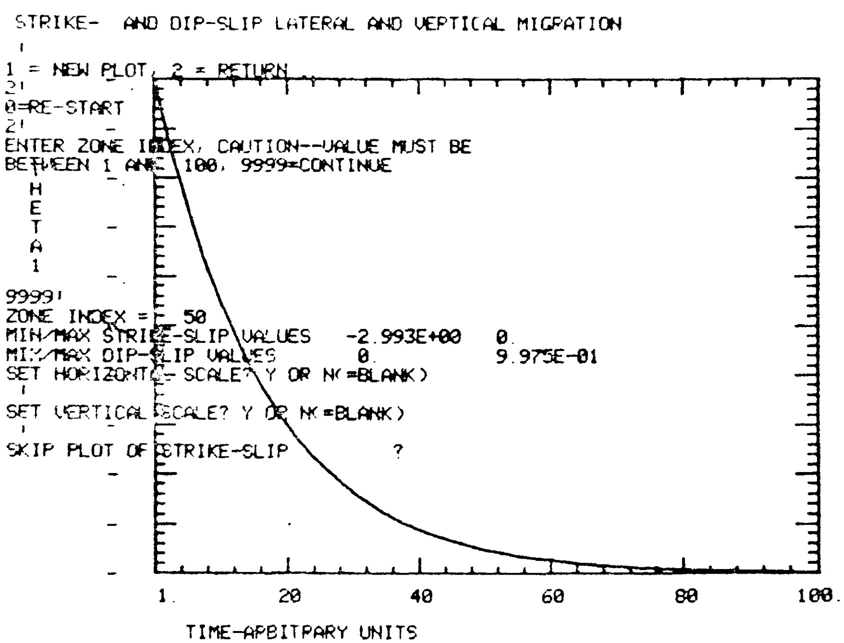
```
!
SPECIFY 2 CORNERS OF DISLOCATION SURFACE FOR DISPLAY
 1=D1X1=D2X1            2=D1X3=D4X3
 4=D2X3=D3X3            5=D3X1=D4X1
 9=C1X1=C2X1           10=C3X3=C4X3
12=C2X3=C3X3           13=C3X1=C4X1
1 9!
TO DISPLAY SLIP AS A FUNCTION OF TIME
ENTER ZONE INDEX,  CAUTION---
INDEX MUST BE BETWEEN 1 AND  100
50!
THETA=ANGLE BETWEEN STRIKE OF FAULT
AND NORTH =     45 000 DEGREES
MIN/MAX VALUES OF EW COMPONENT  -3.373E-01   0.
MIN/MAX VALUES OF NS COMPONENT  -2 935E-01   5.152E-03
MIN/MAX VALUES OF AMPLITUDE      0.          4 343E-01
MIN/MAX VALUES OF AZIMUTH        0.        310 803
(NOTE--TILT AMPLITUDES ARE IN MICRORADIANS,
AZIMUTH IN DEGREES)
THE FOLLOWING ARE PLOTS OF THE EW AND NS COMPONENTS
OF TILT, AND THE TILT AMPLITUDE
AND AZIMUTH (MEASURE CLOCKWISE FROM NORTH).
0=RE-START, 1=CONTINUE
1!
WRITE PLOT TITLE, 80 CHARACTERS
STRIKE-  AND DIP-SLIP/LATERAL AND VERTICAL MIGRATION!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
 !
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
- 5   5!
SKIP PLOT OF EW TILT              ?
```

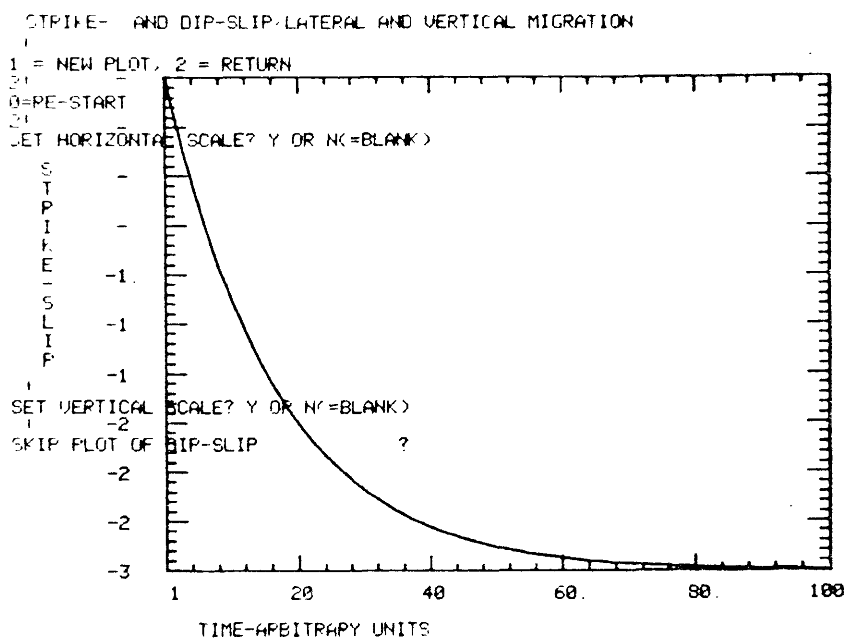

STRIKE- AND DIP-SLIP LATERAL AND VERTICAL MIGRATION
1 = NEW PLOT. 2 = RETURN
SET HORIZONTAL SCALE? Y OR N(=BLANK)
EW TILT
0
SET VERTICAL SCALE? Y OR N(=BLANK)
Y
MIN/MAX Y VALUES
- 4 4
SKIP PLOT OF EW TILT ?
1.
20
40
60
80
100.
TIME-ARBITRARY UNITS

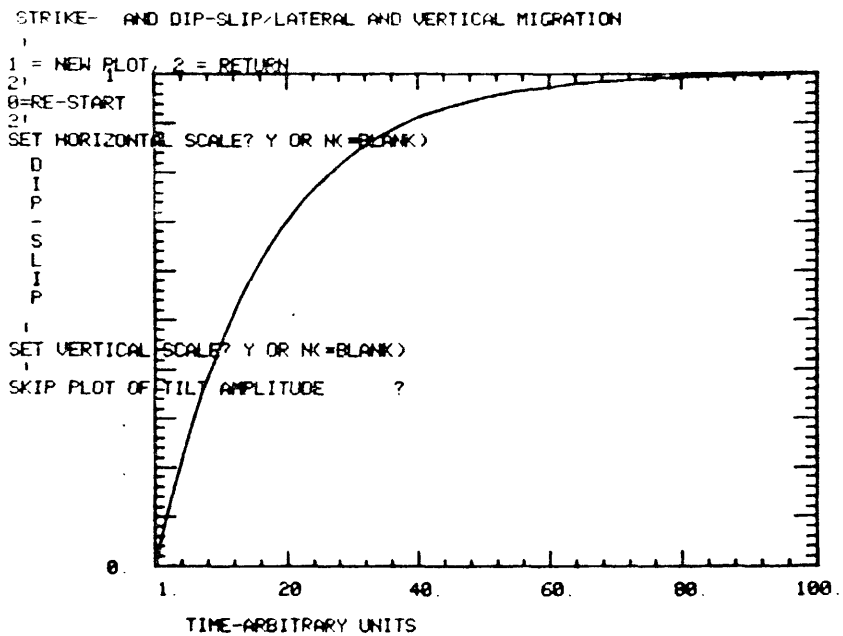

STRIKE- AND DIP-SLIP LATERAL AND VERTICAL MIGRATION
1 = NEW PLOT. 2 = RETURN
0=RE-START
SET HORIZONTAL SCALE? Y OR N(=BLANK)
EW TILT
SET VERTICAL SCALE? Y OR N(=BLANK)
Y
MIN/MAX Y VALUES
- 5 5
SKIP PLOT OF
TILT ?
1
20
40.
60.
80.
100.
TIME-ARBITRARY UNITS

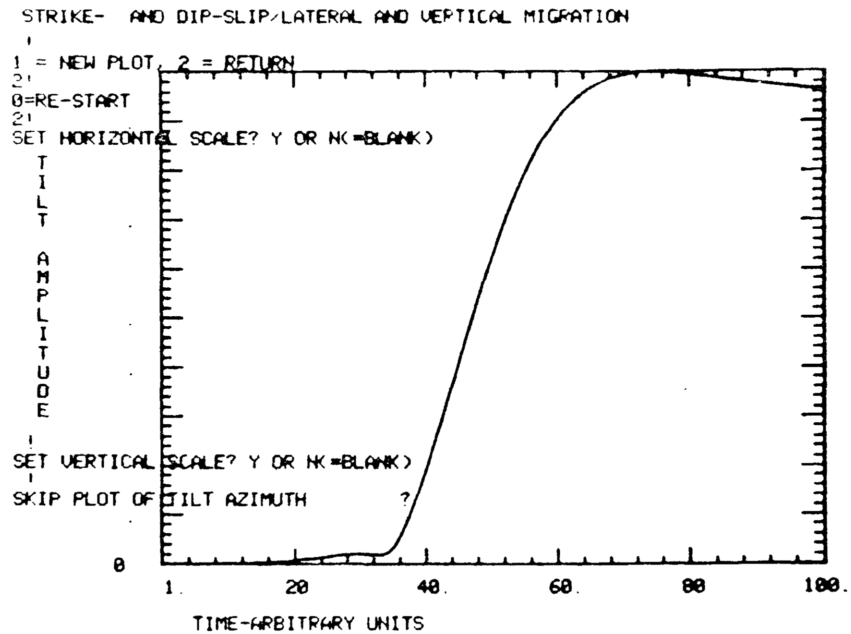

STRIKE- AND DIP-SLIP/LATERAL AND VERTICAL MIGRATION
1 = NEW PLOT, 2 = RETURN
0=RE-START
ENTER ZONE INDEX, CAUTION--VALUE MUST BE
BETWEEN 1 AND 100, 9999=CONTINUE
NS TILT
0.
50
MIN/MAX VALUES OF THETA2(50) = -5.546E-01 0.
MIN/MAX VALUES OF THETA1(50) = -4.305E-02 0.
SET HORIZONTAL SCALE? Y OR NK=BLANK)
SET VERTICAL SCALE? Y OR NK=BLANK)
SKIP PLOT OF THETA2 ?
1
20
40.
60.
80.
100.
TIME-ARBITRARY UNITS

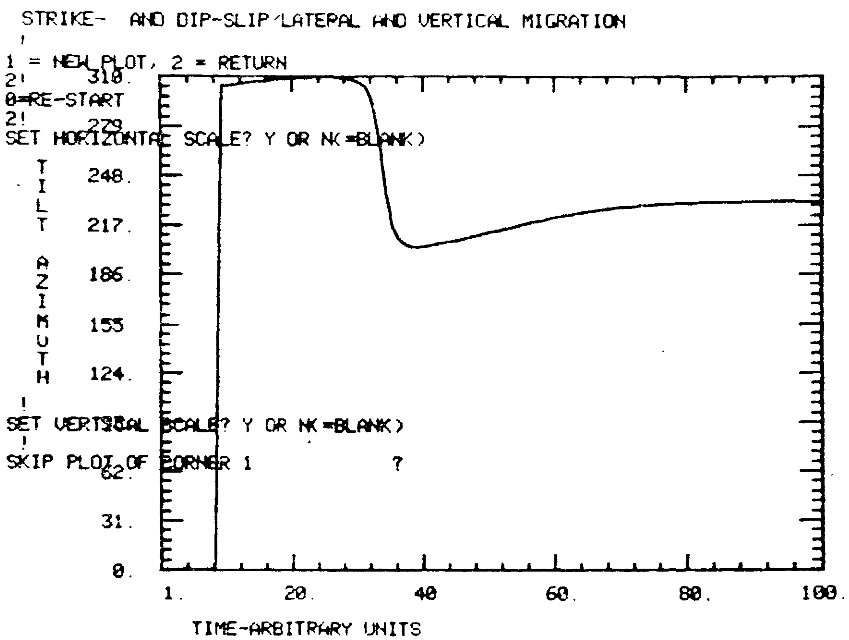

STRIKE- AND DIP-SLIP/LATERAL AND VERTICAL MIGRATION
1 = NEW PLOT, 2 = RETURN
0=RE-START
SET HORIZONTAL SCALE? Y OR NK=BLANK)
THETA2
SET VERTICAL SCALE? Y OR NK=BLANK)
SKIP PLOT OF THETA1 ?
1.
20.
40
60.
80.
100.
TIME-ARBITPAPY UNITS

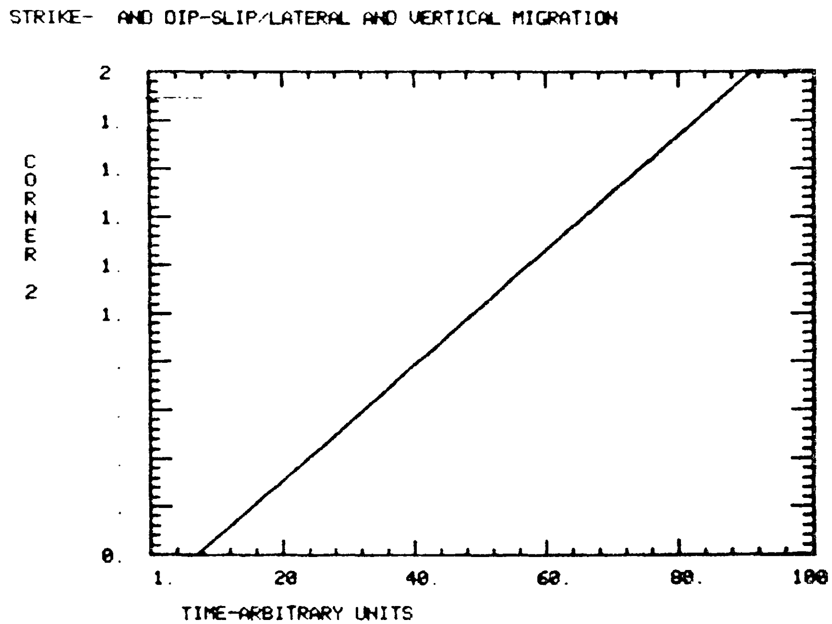

STRIKE- AND DIP-SLIP LATERAL AND VEPTICAL MIGPATION
1 = NEW PLOT, 2 = RETURN
2
0=RE-START
2
ENTER ZONE INDEX, CAUTION--VALUE MUST BE
BETWEEN 1 AND 100, 9999=CONTINUE
THETA 1
9999
ZONE INDEX = 50
MIN/MAX STRIKE-SLIP VALUES -2.993E+00 0.
MIN/MAX DIP-SLIP VALUES 0. 9.975E-01
SET HORIZONTAL SCALE? Y OR N(=BLANK)
SET VERTICAL SCALE? Y OR N(=BLANK)
SKIP PLOT OF STRIKE-SLIP ?
1.
20
40
60
80
100.
TIME-APBITPARY UNITS

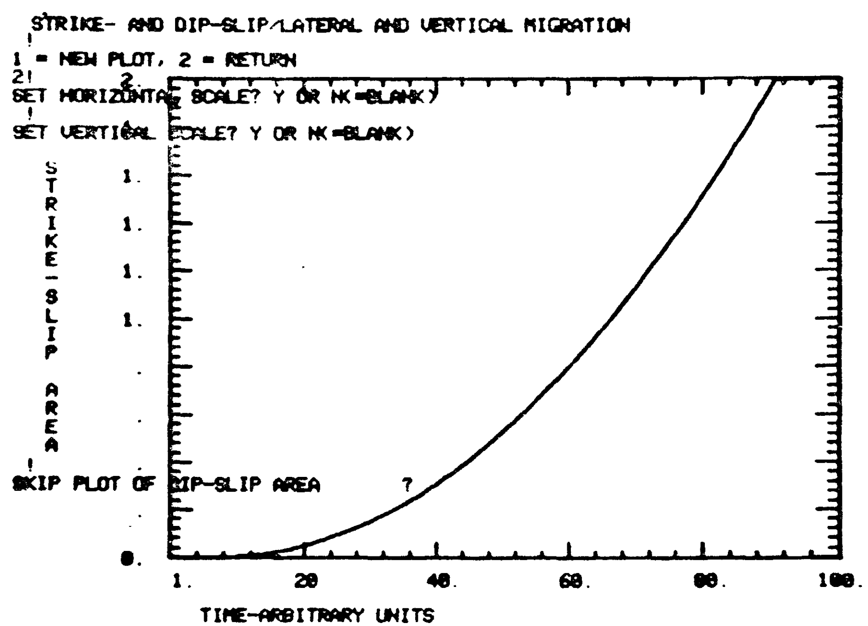

STRIKE- AND DIP-SLIP LATERAL AND VERTICAL MIGRATION
1 = NEW PLOT, 2 = RETURN
0=PE-START
SET HORIZONTAL SCALE? Y OR N(=BLANK)
STPIKE-SLIP
-1
-1
-1
SET VERTICAL SCALE? Y OR N(=BLANK)
-2
SKIP PLOT OF DIP-SLIP ?
-2
-2
-3
1
20
40
60.
80.
100.
TIME-APBITRAPY UNITS

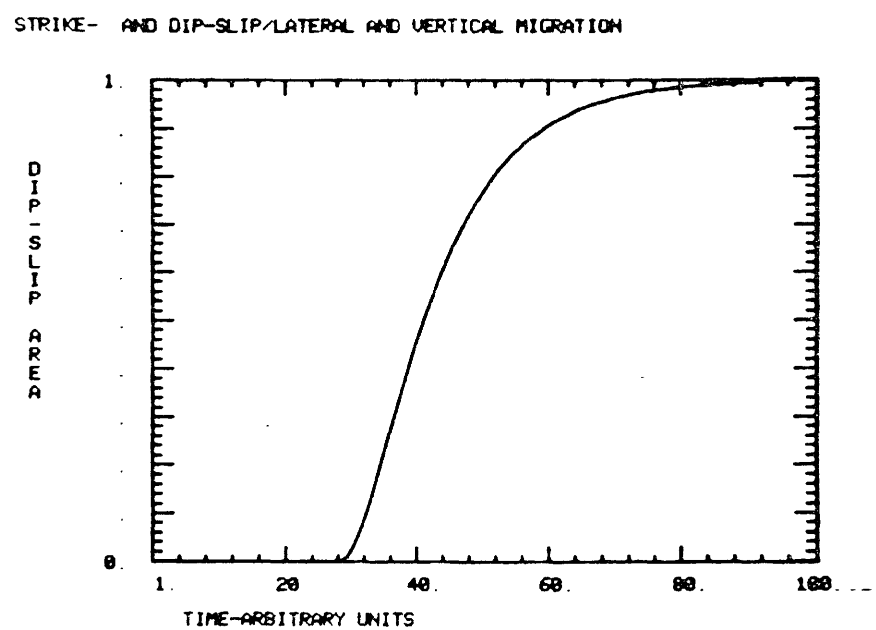

STRIKE- AND DIP-SLIP/LATERAL AND VERTICAL MIGRATION
1 = NEW PLOT, 2 = RETURN
0=RE-START
SET HORIZONTAL SCALE? Y OR N(=BLANK)
DIP-SLIP
SET VERTICAL SCALE? Y OR N(=BLANK)
SKIP PLOT OF TILT AMPLITUDE ?
1
0.
1.
20
40.
60.
80.
100.
TIME-ARBITRARY UNITS


STRIKE- AND DIP-SLIP/LATERAL AND VERTICAL MIGRATION
1 = NEW PLOT, 2 = RETURN
0=RE-START
SET HORIZONTAL SCALE? Y OR N(=BLANK)
TILT AMPLITUDE
SET VERTICAL SCALE? Y OR N(=BLANK)
SKIP PLOT OF TILT AZIMUTH ?
0
1.
20
40.
60.
80
100.
TIME-ARBITRARY UNITS


STRIKE- AND DIP-SLIP/LATEPAL AND VERTICAL MIGRATION
1 = NEW PLOT, 2 = RETURN
0=RE-START
SET HORIZONTAL SCALE? Y OR N(=BLANK)
SET VERTICAL SCALE? Y OR N(=BLANK)
SKIP PLOT OF CORNER 1 ?
TILT AZIMUTH
310.
279.
248.
217.
186.
155.
124.
62.
31.
0.
1.
20.
40
60.
80.
100.
TIME-ARBITRARY UNITS


STRIKE- AND DIP-SLIP/LATERAL AND VERTICAL MIGRATION
CORNER 2
2
1.
1.
1.
1.
1.
0.
1.
20
40.
60.
80.
100.
TIME-ARBITRARY UNITS


STRIKE- AND DIP-SLIP/LATERAL AND VERTICAL MIGRATION
1 = NEW PLOT, 2 = RETURN
2
SET HORIZONTAL SCALE? Y OR NK=BLANK)
SET VERTICAL SCALE? Y OR NK=BLANK)
STRIKE-SLIP AREA
1.
1.
1.
1.
0.
SKIP PLOT OF DIP-SLIP AREA ?
1.
20
40.
60.
80.
100.
TIME-ARBITRARY UNITS

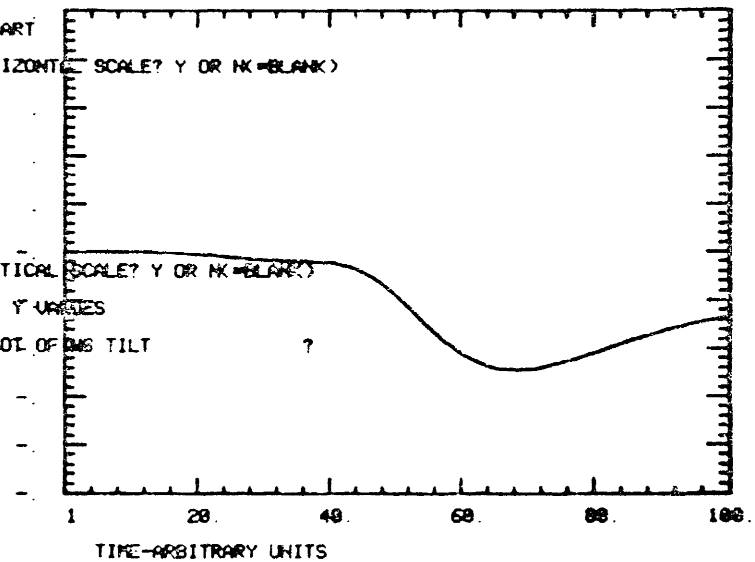

STRIKE- AND DIP-SLIP/LATERAL AND VERTICAL MIGRATION
DIP-SLIP AREA
1.
0.
1.
20
40.
60.
80.
100.
TIME-ARBITRARY UNITS

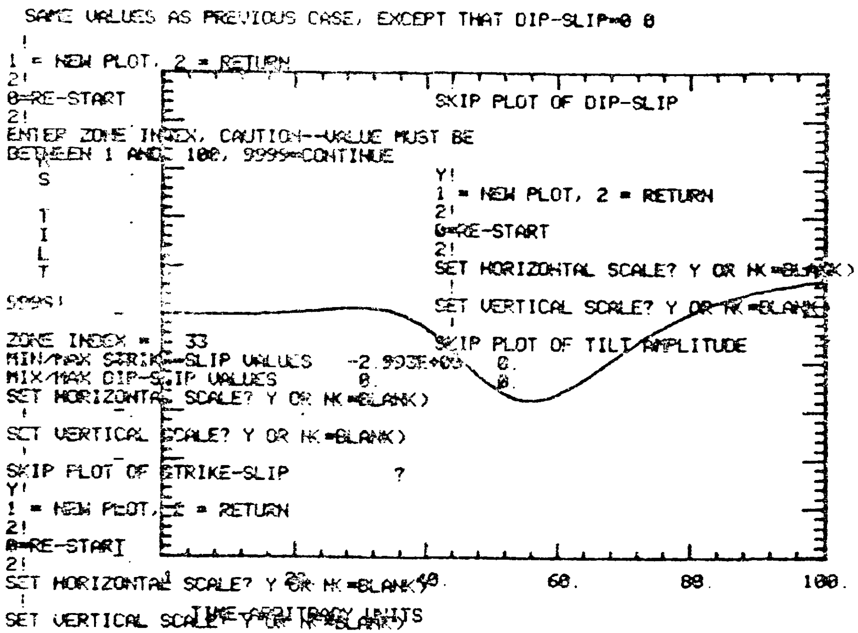
```
 !
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
       .500     0.        .500     0.        .500     0.
       .500     0.       1.500     0.       1.500     1.000
       .500     1.000     .500     0.        0.       0.
      0         0.       0.        0.        0.       0.
      2.000     0.       2.000     1.000     0.       1.000
      0         0.       0.       -3.000     0.       1.000

      1    2    1    2    1    1    2    2
      2    1    2    1    2    2    1    1

         0         1         9         5        25
         1         1         0         0         1
       100         7         6        91        92
1=RE-START WITH ALL NEW VALUES
2=RE-START WITH NEW STRIKE-SLIP VALUE AND ZONE COORDINATES
3=RE-START WITH NEW DIP-SLIP VALUE AND ZONE COORDINATES
4=RE-START WITH NEW STRIKE-SLIP VALUE ONLY
5=RE-START WITH NEW DIP-SLIP VALUE ONLY
6=RE-START WITH NEW TILTMETER COORDINATES ONLY
7=STOP
1
```

```
!
1=ZONE EXPANDS, 2=ZONE CONTRACTS
1!
1/2=SLIP INCREMENTED EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
1 1!
0  =INCREMENT CORNERS SEPARATELY
1/2=INCREMENT ALL CORNERS EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
0 0!
1=VARIABLE INCREMENTED EXPONENTIALLY
2=VARIABLE INCREMENTED LINEARLY
D1X1, D1X3, D2X3, D3X1
1 2 1 2!
C1X1, C1X3, C2X3, C3X1
2 1 2 1!
'U1>0' = LEFT-LATERAL STRIKE-SLIP
'U3>0' = 'X2>0' SIDE DOWN
U1IN, U1FN, U3IN, U3FN
0 -3 0 0!
'TRIGGER' OPTION DESIRED?
Y!
0 = D(I1) > C(I2),  1 = D(I1) < C(I2)
0!
0 = STRIKE-SLIP/1 = DIP-SLIP ZONE TRIGGERED
1 !
SPECIFY I1 AND I2
9 5!
C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN
0 0 0 0 2 0 1 0!
ENTER STATION COORDINATES--(X1,X2)
1 .5
```

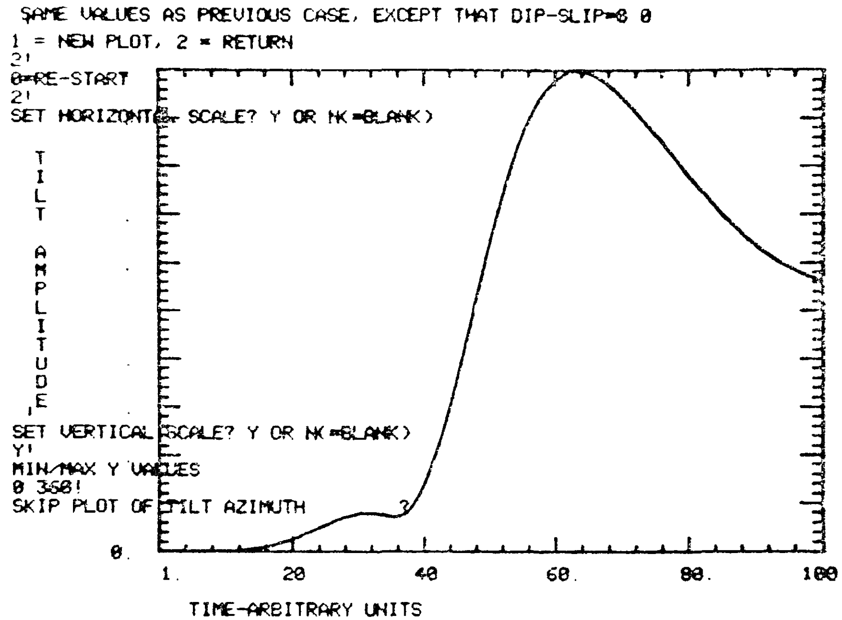
```
SPECIFY 2 CORNERS OF DISLOCATION SURFACE FOR DISPLAY
 1=D1X1=D2X1              2=D1X3=D4X3
 4=D2X3=D3X3              5=D3X1=D4X1
 9=C1X1=C2X1             10=C3X3=C4X3
12=C2X3=C3X3             13=C3X1=C4X1
1 12!
TO DISPLAY SLIP AS A FUNCTION OF TIME
ENTER ZONE INDEX.  CAUTION---
INDEX MUST BE BETWEEN 1 AND  100
33!
THETA=ANGLE BETWEEN STRIKE OF FAULT
AND NORTH =      45.000 DEGREES
MIN/MAX VALUES OF EW COMPONENT  -9.795E-02   0.
MIN/MAX VALUES OF NS COMPONENT  -7.836E-02   2.708E-02
MIN/MAX VALUES OF AMPLITUDE      0.          1.675E-01
MIN/MAX VALUES OF AZIMUTH        0.        310.803
(NOTE--TILT AMPLITUDES ARE IN MICRORADIANS,
AZIMUTH IN DEGREES)
THE FOLLOWING ARE PLOTS OF THE EW AND NS COMPONENTS
OF TILT, AND THE TILT AMPLITUDE
AND AZIMUTH (MEASURE CLOCKWISE FROM NORTH).
0=RE-START, 1=CONTINUE
1!
WRITE PLOT TITLE, 80 CHARACTERS
SAME VALUES AS PREVIOUS CASE, EXCEPT THAT DIP-SLIP=0.0!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
 !
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
-.2 .2!
SKIP PLOT OF EW TILT              ?
```

```
SAME VALUES AS PREVIOUS CASE, EXCEPT THAT DIP-SLIP=0 0
!
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
\\\
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
-.2 .2!
SKIP PLOT OF NS TILT              ?
```

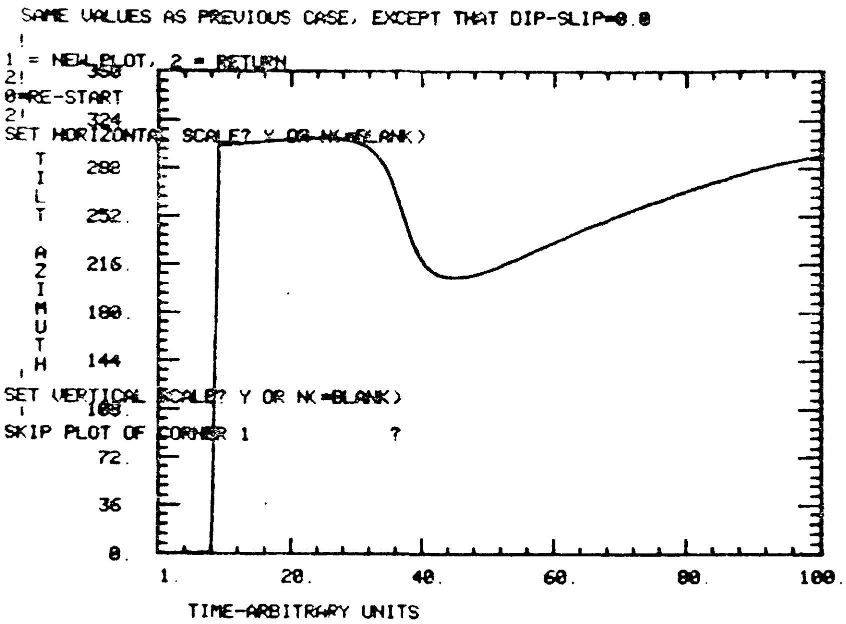


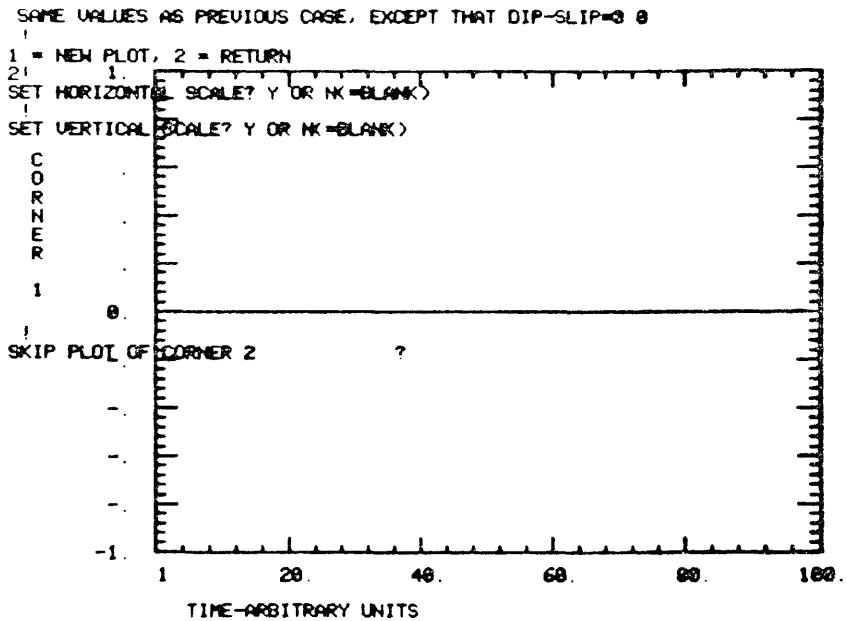


SAME VALUES AS PREVIOUS CASE, EXCEPT THAT DIP-SLIP=0 0
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
ENTER ZONE INDEX, CAUTION--VALUE MUST BE
BETWEEN 1 AND 100, 9999=CONTINUE
S
T
I
L
T
9999!
ZONE INDEX = 33
MIN/MAX STRIKE-SLIP VALUES -2.993E+03 0.
MIX/MAX DIP-SLIP VALUES 0. 0.
SET HORIZONTAL SCALE? Y OR NK=BLANK)
SET VERTICAL SCALE? Y OR NK=BLANK)
SKIP PLOT OF STRIKE-SLIP ?
Y!
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
SET HORIZONTAL SCALE? Y OR NK=BLANK)
SET VERTICAL SCALE? Y OR NK=BLANK)
SKIP PLOT OF DIP-SLIP
Y!
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
SET HORIZONTAL SCALE? Y OR NK=BLANK)
SET VERTICAL SCALE? Y OR NK=BLANK)
SKIP PLOT OF TILT AMPLITUDE
1
20
40
60.
80.
100.
TIME-ARBITRARY UNITS



SAME VALUES AS PREVIOUS CASE, EXCEPT THAT DIP-SLIP=0 0
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
SET HORIZONTAL SCALE? Y OR NK=BLANK)
TILT AMPLITUDE
SET VERTICAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX Y VALUES
0 360!
SKIP PLOT OF TILT AZIMUTH ?
0.
1.
20
40
60.
80.
100.
TIME-ARBITRARY UNITS


SAME VALUES AS PREVIOUS CASE, EXCEPT THAT DIP-SLIP=0.0
1 = NEW PLOT, 2 = RETURN
0=RE-START
SET HORIZONTAL SCALE? Y OR NK=BLANK)
TILT AZIMUTH
360
324
288
252.
216.
180.
144
108.
72.
36
0.
SET VERTICAL SCALE? Y OR NK=BLANK)
SKIP PLOT OF CORNER 1 ?
1.
20.
40.
60.
80.
100.
TIME-ARBITRARY UNITS


SAME VALUES AS PREVIOUS CASE, EXCEPT THAT DIP-SLIP=0 0
1 = NEW PLOT, 2 = RETURN
SET HORIZONTAL SCALE? Y OR NK=BLANK)
SET VERTICAL SCALE? Y OR NK=BLANK)
CORNER 1
1.
0.
-1.
SKIP PLOT OF CORNER 2 ?
1
20.
40.
60.
80.
100.
TIME-ARBITRARY UNITS

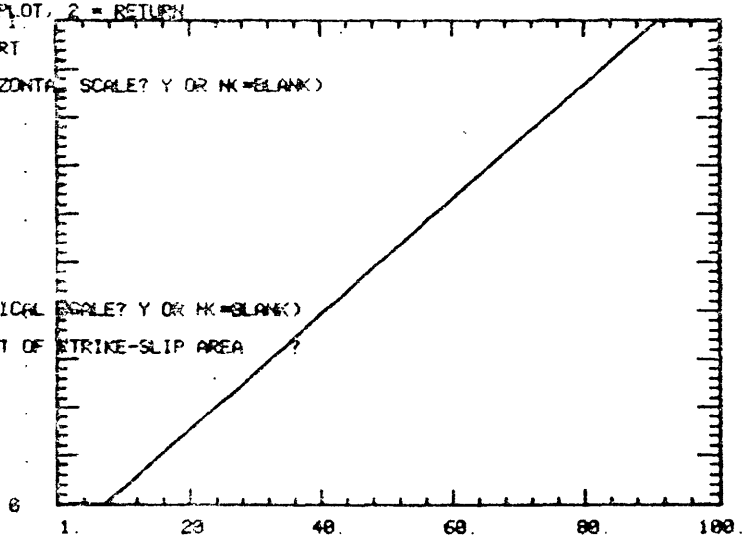
SAME VALUES AS PREVIOUS CASE, EXCEPT THAT DIP-SLIP=3 0

1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
SET HORIZONTAL SCALE? Y OR N(=BLANK)

CORNER 2

SET VERTICAL SCALE? Y OR N(=BLANK)

SKIP PLOT OF STRIKE-SLIP AREA ?



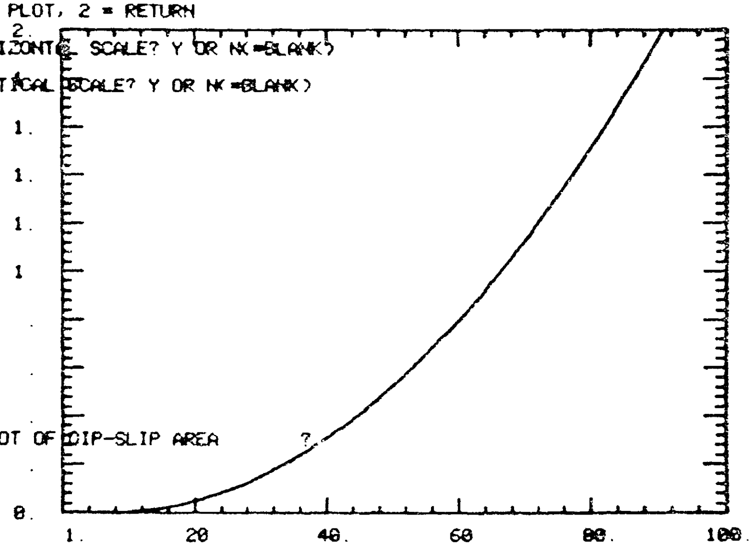
SAME VALUES AS PREVIOUS CASE, EXCEPT THAT DIP-SLIP=6 0

1 = NEW PLOT, 2 = RETURN
2!
SET HORIZONTAL SCALE? Y OR N(=BLANK)

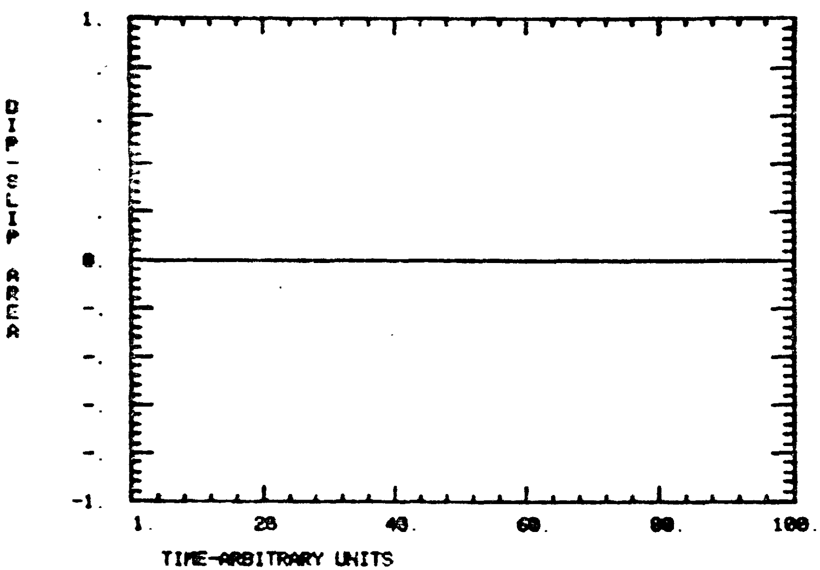
SET VERTICAL SCALE? Y OR N(=BLANK)

STRIKE-SLIP AREA

SKIP PLOT OF DIP-SLIP AREA ?



SAME VALUES AS PREVIOUS CASE, EXCEPT THAT DIP-SLIP=0.0




```
 !
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START
2!
     0.          0.          0.          0.          0.          0.
     0.          0.          0.          0.          0.          0.
     0.          0.          0.          0.          0.          0.
     0.          0.          0.          0.          0.          0.
     2.000       0.          2.000       1.000       0.          1.000
     0.          0.          0.         -3.000       0.          0.

     1     2     1     2     2     1     2     2
     2     1     2     2     1     2     1     1

         0          1          9          5          0
         1          1          0          0          1
        100         7          0          91         92
1=RE-START WITH ALL NEW VALUES
2=RE-START WITH NEW STRIKE-SLIP VALUE AND ZONE COORDINATES
3=RE-START WITH NEW DIP-SLIP VALUE AND ZONE COORDINATES
4=RE-START WITH NEW STRIKE-SLIP VALUE ONLY
5=RE-START WITH NEW DIP-SLIP VALUE ONLY
6=RE-START WITH NEW TILTMETER COORDINATES ONLY
7=STOP
7!
OK - ^EDIT
```

PROGRAM LISTING FOR SLPPRP

```
DELETE(LGO,OUTPUT,SLPPRP)
SLPPRP.
EXIT.
LIBCOPY(GRAPHIC,TXLGO/RR,TXLGO)
LIBCOPY(JDRAT,NPLGO/RR,NPLGO)
DELETE(LGO,OUTPUT,SLPPRP)
RUN76(S)
LINK(F=LGO,F=TXLGO,F=NPLGO,B=SLPPRP)
SLPPRP.
FIN.
EOF
         PROGRAM SLPPRP(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
         COMMON/TVPUCL/TVPUL(8)
         COMMON/TVTUNE/ITUNE(30)
         COMMON/JFLCT/XLT,XRT,YLC,YLP,MAJX,MAJY,KX(2),KY(2),
        1LTITL(8),LC,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
         DIMENSION IFET(8)
         DIMENSION TX(100),TY(100),T(100),CORNR1(100),CORNR2(100)
         DIMENSION B(20),C(40),A(30),IT(4),ICRNR(16),STSLIP(100)
         DIMENSION DIPSLP(100),SSAREA(100),DSAREA(100)
         DIMENSION THETA1(100,100),THETA2(100,100)
         DIMENSION TAMP(100),TAZM(100)
         CALL FET(5LTAPE7,IFET,8)
         IFET(2)=IFET(2).OR.0000 0010 0000 0000 0000B
         IFET(8)=IFET(8).OR.4000 0000 0000 0000 0000B
         CALL FET(5LTAPE7,IFET,-8)
      67 DO 101 J=1,18
         B(J)=0.  $C(J+18)=0. $C(J)=0.
  101    CONTINUE
         WRITE(7,1)
    1    FORMAT(*1=ZONE EXPANDS, 2=ZONE CONTRACTS*)
         CALL GETNUM(A) $IFLAG=A(1) $WRITE(7,2)
       2 FORMAT(*1/2=SLIP INCREMENTED EXPONENTIALLY/LINEARLY*,/,
        1       *STRIKE-SLIP/DIP-SLIP*)
         CALL GETNUM(A) $KFLAGS=A(1) $KFLAGD=A(2) $WRITE(7,3)
       3 FORMAT(*0   =INCREMENT CORNERS SEPARATELY*,/,
        1       *1/2=INCREMENT ALL CORNERS EXPONENTIALLY/LINEARLY*,/,
        1       *STRIKE-SLIP/DIP-SLIP*)
         CALL GETNUM(A) $LFLAG=A(1) $MFLAG=A(2)
         DO 110 I=1,4
         IT(I)=5
  110    CONTINUE
         DO 4 I=1,16
       4 ICRNR(I)=1
         IF(LFLAG.EQ.0.OR.MFLAG.EQ.0)CALL SLPCRN(LFLAG,MFLAG,ICRNR)
         IF(LFLAG.EQ.2)GOTO 5 $GOTO6
       5 DO 7 I=9,16
    7    ICRNR(I)=2
    6    IF(MFLAG.EQ.2)GOTO8 $GOTO9
    8    DO 10 I=1,8
  10     ICRNR(I)=2
       9 WRITE(7,11) $CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
  11     FORMAT(*'U1>0' = LEFT-LATERAL STRIKE-SLIP*,/,
        1       *'U3>0' = 'X2>0' SIDE DOWN*,/,
        1       *U1IN, U1FN, U3IN, U3FN*)
         C(35)=A(3) $C(36)=A(4) $WRITE(7,12)
   12    FORMAT(*"TRIGGER" OPTION DESIRED?*)
         READ(7,13)TRIG $IF(TRIG.EQ.1HY)CALL TRIGGR(IT)
```

```
   13 FORMAT(A1)
      IF(C(33).EQ.0..AND.C(34).EQ.0.)GOTO14
      WRITE(7,15) $CALL GETNUM(A) $C(17)=C(19)=A(1)
   15 FORMAT(*C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN*)
      C(18)=C(24)=A(2) $C(20)=C(22)=A(3) $C(21)=C(23)=A(4)
      C(25)=C(27)=A(5) $C(26)=C(32)=A(6) $C(28)=C(30)=A(7)
      C(29)=C(31)=A(8)
   14 IF(C(35).EQ.0..AND.C(36).EQ.0.)GOTO16
      WRITE(7,17) $CALL GETNUM(A) $C(1)=C(3)=A(1) $C(2)=C(8)=A(2)
   17 FORMAT(*D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,D3X1FN*)
      C(4)=C(6)=A(3) $C(5)=C(7)=A(4) $C(9)=C(11)=A(5)
      C(10)=C(16)=A(6) $C(12)=C(14)=A(7) $C(13)=C(15)=A(8)
   16 WRITE(7,18) $CALL GETNUM(A) $X1=A(1) $X2=A(2)
   18 FORMAT(*ENTER STATION COORDINATES--(X1,X2)*)
      X3=0. $NRECPT=100 $N=(.075*NRECPT) $NSLPPT=NRECPT
      N1=N+1 $N2=NRECPT-(.075*NRECPT)-1
      N3=N2+1 $THETA=45. $PI=3.1415926 $WRITE(7,30)
      RNSLPP=NSLPPT $RN1=N1 $RN2=N2 $ONE=1. $ITIME=99999
   30 FORMAT(*SPECIFY 2 CORNERS OF DISLOCATION SURFACE FOR DISPLAY*,/,
     1 * 1=D1X1=D2X1           2=D1X3=D4X3*,/,
     1 * 4=D2X3=D3X3           5=D3X1=D4X1*,/,
     1 * 9=C1X1=C2X1          10=C3X3=C4X3*,/,
     1 *12=C2X3=C3X3          13=C3X1=C4X1*)
      CALL GETNUM(A) $M1=A(1) $M2=A(2)
      WRITE(7,113)NRECPT $CALL GETNUM(A)  $INDEX=A(1)
  113 FORMAT(*TO DISPLAY SLIP AS A FUNCTION OF TIME*,/,
     1        *ENTER ZONE INDEX,  CAUTION---*,/,
     1        *INDEX MUST BE BETWEEN 1 AND *,I4)
   75 CONTINUE
      DO  21 J=1,8
      D(J)=C(J)  $D(J+8)=C(J+16)
   21 CONTINUE
      D(17)=C(33)  $D(18)=C(35)
      DO 19 I=1,N
      T(I)=I
   22 DO 63 K=1,NSLPPT
      CALL CMPTLT(D,X1,X2,X3,T1,T2)
      THETA1(I,K)=T1 $THETA2(I,K)=T2
      IF(I.EQ.INDEX)STSLIP(K)=D(17) $IF(I.EQ.INDEX)DIPSLP(K)=D(18)
   63 CONTINUE
      CORNR1(I)=D(M1) $CORNR2(I)=D(M2)
      SSAREA(I)=ABS((D(15)-D(9))*(D(12)-D(10)))
      DSAREA(I)=ABS((D(7)-D(1))*(D(4)-D(2)))
   19 CONTINUE
      DO 24 J=1,8
      D(J)=C(J+8)  $D(J+8)=C(J+24)
   24 CONTINUE
      D(17)=C(34)  $D(18)=C(36)
      DO 23 I=N3,NRECPT
      T(I)=I
      DO 64 K=1,NSLPPT
      RK=K $CALL XPNSHL(C(33),C(34),ONE,RNSLPP,RK,Y1)
      CALL XPNSHL(C(35),C(36),ONE,RNSLPP,RK,Y2)
      IF(KFLAGS.NE.1)CALL ALINAR(C(33),C(34),ONE,RNSLPP,RK,Y1)
      IF(KFLAGD.NE.1)CALL ALINAR(C(35),C(36),ONE,RNSLPP,RK,Y2)
      D(17)=Y1  $D(18)=Y2
      CALL CMPTLT(D,X1,X2,X3,T1,T2)
      THETA1(I,K)=T1 $THETA2(I,K)=T2
```

```
      IF(I.EQ.INDEX)STSLIP(K)=D(17) $IF(I.EQ.INDEX)DIPSLP(K)=D(18)
   64 CONTINUE
      CORNR1(I)=D(M1) $CORNR2(I)=D(M2)
      SSAREA(I)=ABS((D(15)-D(9))*(D(12)-D(10)))
      DSAREA(I)=ABS((D(7)-D(1))*(D(4)-D(2)))
   23 CONTINUE
      IFGREC=0
      DO 25 I=N1,N2
      T(I)=I  $RI=I
      DO 65 K=1,NSLPPT
      RK=K
      IF(TRIG.EQ.1HY)GOTO26
      DO 27 J=1,8
      J8=J+8 $A1=C(J) $B=C(J8) $E=C(J8+8) $F=C(J+24)
      CALL XPNSHL(A1,B,RN1,RN2,RI,Y1)
      CALL XPNSHL(E,F,RN1,RN2,RI,Y2)
      IF(ICRNR(J).EQ.2)CALL ALINAR(A1,B,RN1,RN2,RI,Y1)
      IF(ICRNR(J8).EQ.2)CALL ALINAR(E,F,RN1,RN2,RI,Y2)
      D(J)=Y1 $D(J8)=Y2
   27 CONTINUE
      CALL XPNSHL(C(33),C(34),ONE,RNSLPP,RK,Y1)
      CALL XPNSHL(C(35),C(36),ONE,RNSLPP,RK,Y2)
      IF(KFLAGS.NE.1)CALL ALINAR(C(33),C(34),ONE,RNSLPP,RK,Y1)
      IF(KFLAGD.NE.1)CALL ALINAR(C(35),C(36),ONE,RNSLPP,RK,Y2)
      D(17)=Y1 $D(18)=Y2 $GOTO28
   26 IF(IT(2).EQ.1.AND.KFLAGS.EQ.1)CALL XPNSHL(C(33),C(34),ONE,
     1 RNSLPP,RK,Y)
      IF(IT(2).EQ.1.AND.KFLAGS.NE.1)CALL ALINAR(C(33),C(34), ONE,
     1 RNSLPP,RK,Y)
      IF(IT(2).EQ.0.AND.KFLAGD.EQ.1)CALL XPNSHL(C(35),C(36),ONE,
     1 RNSLPP,RK,Y)
      IF(IT(2).EQ.0.AND.KFLAGD.NE.1)CALL ALINAR(C(35),C(36),ONE,
     1 RNSLPP,RK,Y)
      IF(IT(2).EQ.0)D(18)=Y $IF(IT(2).EQ.1)D(17)=Y
      DO 32 J=1,8
      J8=J+8 $A1=C(J) $B=C(J8) $E=C(J8+8) $F=C(J+24)
      CALL XPNSHL(A1,B,RN1,RN2,RI,Y1)
      CALL XPNSHL(E,F,RN1,RN2,RI,Y2)
      IF(ICRNR(J).EQ.2)CALL ALINAR(A1,B,RN1,RN2,RI,Y1)
      IF(ICRNR(J8).EQ.2)CALL ALINAR(E,F,RN1,RN2,RI,Y2)
      D(J)=Y1 $D(J8)=Y2
   32 CONTINUE
      MIT3=IT(3)  $MIT4=IT(4) $IF(IT(1).EQ.1)GOTO29
      IF(D(MIT3).LT.C(MIT4).AND.IT(2).EQ.0)D(17)=C(33)
      IF(D(MIT3).LT.C(MIT4).AND.IT(2).EQ.1)D(18)=C(35)
      IF(D(MIT3).LT.C(MIT4).AND.IT(2).EQ.1)GOTO33
      IF(D(MIT3).GE.C(MIT4))GOTO35
      DO 40 J=9,16
   40 D(J)=C(J+8)
      GOTO28
   29 IF(D(MIT3).GT.C(MIT4).AND.IT(2).EQ.0)D(17)=C(33)
      IF(D(MIT3).GT.C(MIT4).AND.IT(2).EQ.1)D(18)=C(35)
      IF(D(MIT3).GT.C(MIT4).AND.IT(2).EQ.1)GOTO33
      IF(D(MIT3).LE.C(MIT4))GOTO35
      DO 34 J=9,16
   34 D(J)=C(J+8)
      GOTO28
   35 DO 36 J=1,8
```

```
   36  D(J)=C(J)
       GO 1028
    35 IFGREC=IFGREC+1
       IF(IFGREC.EQ.1)ITIME=I
       IF(IFGREC.EQ.1)RITIME=ITIME
       IF(IT(2).EQ.0.AND.KFLAGS.EQ.1)CALL XPNSHL(C(33),C(34),ONE,
      1 RNSLPP,RK,Y)
       IF(IT(2).EQ.0.AND.KFLAGS.NE.1)CALL ALINAR(C(33),C(34),ONE,
      1 RNSLPP,RK,Y)
       IF(IT(2).EQ.1.AND.KFLAGD.EQ.1)CALL XPNSHL(C(35),C(36),ONE,
      1 RNSLPP,RK,Y)
       IF(IT(2).EQ.1.AND.KFLAGD.NE.1)CALL ALINAR(C(35),C(36),ONE,
      1 RNSLPP,RK,Y)
       IF(IT(2).EQ.1)D(18)=Y $IF(IT(2).EQ.0)D(17)=Y
       IF(IT(2).EQ.1)GOTO37
       DO 38 J=9,16
       A1=C(J+8)  $B=C(J+16) $CALL XPNSHL(A1,B,RITIME,RN2,RI,Y)
       IF(ICRNR(J).EQ.2)CALL ALINAR(A1,B,RITIME,RN2,RI,Y)
   38  D(J)=Y
       GOTO28
   37  DO 39 J=1,8
       E=C(J) $F=C(J+8) $CALL XPNSHL(E,F,RITIME,RN2,RI,Y)
       IF(ICRNR(J).EQ.2)CALL ALINAR(E,F,RITIME,RN2,RI,Y)
   39  D(J)=Y
   28  CALL CMPTLT(D,X1,X2,X3,T1,T2)
       THETA1(I,K)=T1 $THETA2(I,K)=T2
       IF(I.EQ.INDEX)STSLIP(K)=D(17)   $IF(I.EQ.INDEX)DIPSLP(K)=D(18)
   65  CONTINUE
       CORNR1(I)=D(M1) $CORNR2(I)=D(M2)
       SSAREA(I)=ABS((D(15)-D(9))*(D(12)-D(10)))
       DSAREA(I)=ABS((D(7)-D(1))*(D(4)-D(2)))
    25 CONTINUE
       DO 41 I=1,NRECPT
       SUM11=SUM12=SUM21=SUM22=0.
       DO 42 J=1,I
       AT=THETA1(J,I+1-J) $B=THETA2(J,I+1-J) $SUM11=AT+SUM11
       SUM21=B+SUM21
   42  CONTINUE
       LM=I-1 $IF(LM.LT.1)GOTO43
       DO 44 J=1,LM
       AT=THETA1(J,I-J) $B=THETA2(J,I-J) $SUM12=AT+SUM12 $SUM22=B+SUM22
    44 CONTINUE
   43  TX(I)=SUM21-SUM22 $TY(I)=SUM11-SUM12
   41  CONTINUE
       IF(IFLAG.EQ.2)GOTO46
       AR=TX(1) $B=TY(1)
       DO 47 I=1,NRECPT
       TX(I)=TX(I)-AR
   47  TY(I)=TY(I)-B
       GOTO48
    46 DO 49 I=1,NRECPT
       TX(I)=TX(NRECPT)-TX(I)
    49 TY(I)=TY(NRECPT)-TY(I)
   48  CT=COS(THETA*(PI/180.))
       ST=SIN(THETA*(PI/180.))
       DO 50 I=1,NRECPT
       B=TX(I) $AR=TY(I)
       TY(I)=AR*CT + B*ST $TX(I)=-AR*ST + B*CT
```

```
      TAMP(I)=SQRT((TX(I)**2)+(TY(I)**2))
      IF(TY(I).EQ.0.)TY(I)=1.E-20
      TAZM(I)=(ATAN(TX(I)/TY(I)))*(180./PI)
      IF(TY(I).LT.0.)TAZM(I)=TAZM(I)+180.
      IF(TAZM(I).LT.0.)TAZM(I)=TAZM(I)+360.
      IF(TAZM(I).GT.360.)TAZM(I)=TAZM(I)-360.
   50 CONTINUE
      WRITE(7,61)THETA
   61 FORMAT(*THETA=ANGLE BETWEEN STRIKE OF FAULT*,/,
     1        *AND NORTH = *,F10.3,* DEGREES*)
      CALL AMINMX(TX,NRECPT,TXMIN,TXMAX)
      CALL AMINMX(TY,NRECPT,TYMIN,TYMAX)
      CALL AMINMX(TAMP,NRECPT,TAMPMN,TAMPMX)
      CALL AMINMX(TAZM,NRECPT,TAZMMN,TAZMMX)
      WRITE(7,51)TXMIN,TXMAX,TYMIN,TYMAX,TAMPMN,TAMPMX,TAZMMN,TAZMMX
   51 FORMAT(*MIN/MAX VALUES OF EW COMPONENT*,2X,E10.3,2X,E10.3,/,
     1 *MIN/MAX VALUES OF NS COMPONENT*,2X,E10.3,2X,E10.3,/,
     1 *MIN/MAX VALUES OF AMPLITUDE*,5X,E10.3,2X,E10.3,/,
     1 *MIN/MAX VALUES OF AZIMUTH*,3X,F10.3,2X,F10.3,/,
     1 *(NOTE--TILT AMPLITUDES ARE IN MICRORADIANS,*,/,
     1 *AZIMUTH IN DEGREES)*)
      WRITE(7,52) $CALL GETNUM(A) $IF(A(1).EQ.0.)GOTO100
   52 FORMAT(*THE FOLLOWING ARE PLOTS OF THE EW AND NS COMPONENTS*,/,
     1 *OF TILT, AND THE TILT AMPLITUDE*,/,
     1 *AND AZIMUTH (MEASURE CLOCKWISE FROM NORTH).*,/,
     1 *0=RE-START, 1=CONTINUE*)
      LU=7 $LNLGX=1 $LNLGY=1 $NCLX=2 $NCLY=2
      DO 200 KM=1,8
  200 LTITL2(KM)=10H
      WRITE(7,53) $READ(7,54)(LTITL(I),I=1,8)
   53 FORMAT(*WRITE PLOT TITLE, 80 CHARACTERS*)
   54 FORMAT(8A10)
      MAJX=5 $MAJY=10 $KTER1=10HEW TILT      $KTER2=10H
      CALL AGRAPH(TX,T,TXMIN,TXMAX,NRECPT,KTER1,KTER2)
      WRITE(7,55) $CALL GETNUM(A) $IF(A(1).EQ.0.)GOTO100
   55 FORMAT(*0=RE-START*)
      KTER1=10HNS TILT      $KTER2=10H
      CALL AGRAPH(TY,T,TYMIN,TYMAX,NRECPT,KTER1,KTER2)
      WRITE(7,55) $CALL GETNUM(A) $IF(A(1).EQ.0.)GOTO100
      KTER1=10HTHETA2
   56 WRITE(7,57)NSLPPT
   57 FORMAT(*ENTER ZONE INDEX, CAUTION--VALUE MUST BE*,/,
     1 *BETWEEN 1 AND *,I4,*, 9999=CONTINUE*)
      CALL GETNUM(A) $I=A(1) $IF(I.EQ.9999)GOTO58
      IF(I.GT.NSLPPT)GOTO56 $IKJ=I
      DO 59 K=1,NSLPPT
      TY(K)=THETA1(IKJ,K)
   59 TX(K)=THETA2(IKJ,K)
      CALL AMINMX(TX,NSLPPT,TXMIN,TXMAX)
      CALL AMINMX(TY,NSLPPT,TYMIN,TYMAX)
      WRITE(7,60)IKJ,TXMIN,TXMAX,IKJ,TYMIN,TYMAX
   60 FORMAT(*MIN/MAX VALUES OF THETA2(*,I4,*) = *,5X,E10.3,2X,E10.3,/
     1 *MIN/MAX VALUES OF THETA1(*,I4,*) = *,5X,E10.3,2X,E10.3)
      CALL AGRAPH(TX,T,TXMIN,TXMAX,NSLPPT,KTER1,KTER2)
      WRITE(7,55) $CALL GETNUM(A) $IF(A(1).EQ.0.)GOTO100
      KTER1=10HTHETA1
      CALL AGRAPH(TY,T,TYMIN,TYMAX,NSLPPT,KTER1,KTER2)
      WRITE(7,55) $CALL GETNUM(A) $IF(A(1).EQ.0.)GOTO100
```

```
      GOTO56
   58 CALL AMINMX(STSLIP,NSLPPT,SSMIN,SSMAX)
      CALL AMINMX(DIPSLP,NSLPPT,DSMIN,DSMAX)
      WRITE(7,114)INDEX,SSMIN,SSMAX,DSMIN,DSMAX
  114 FORMAT(*ZONE INDEX = *,I4,/,
     1       *MIN/MAX STRIKE-SLIP VALUES *,2X,E10.3,2X,E10.3,/,
     1       *MIX/MAX DIP-SLIP VALUES *,5X,E10.3,2X,F10.3)
      KTER1=10HSTRIKE-SLI  $KTER2=10HP
      CALL AGRAPH(STSLIP,T,SSMIN,SSMAX,NSLPPT,KTER1,KTER2)
      WRITE(7,55) $CALL GETNUM(A) $IF(A(1).EQ.0.)GOTO100
      KTER1=10HDIP-SLIP   $KTER2=10H
      CALL AGRAPH(DIPSLP,T,DSMIN,DSMAX,NSLPPT,KTER1,KTER2)
      WRITE(7,55) $CALL GETNUM(A) $IF(A(1).EQ.0.)GOTO100
   68 KTER1=10HTILT AMPLI  $KTER2=10HTUDE
      CALL AGRAPH(TAMP,T,TAMPMN,TAMPMX,NRECPT,KTER1,KTER2)
      WRITE(7,55) $CALL GETNUM(A) $IF(A(1).EQ.0.)GOTO100
      KTER1=10HTILT AZIMU  $KTER2=10HTH
      CALL AGRAPH(TAZM,T,TAZMMN,TAZMMX,NRECPT,KTER1,KTER2)
      WRITE(7,55) $CALL GETNUM(A) $IF(A(1).EQ.0.)GOTO100
      KTER1=10HCORNER 1    $KTER2=10H
      CALL AMINMX(CORNR1,NRECPT,CMIN,CMAX)
      CALL AGRAPH(CORNR1,T,CMIN,CMAX,NRECPT,KTER1,KTER2)
      KTER1=10HCORNER 2
      CALL AMINMX(CORNR2,NRECPT,CMIN,CMAX)
      CALL AGRAPH(CORNR2,T,CMIN,CMAX,NRECPT,KTER1,KTER2)
      WRITE(7,55) $CALL GETNUM(A) $IF(A(1).EQ.0.)GOTO100
      KTER1=10HSTRIKE-SLI  $KTER2=10HP AREA
      CALL AMINMX(SSAREA,NRECPT,CMIN,CMAX)
      CALL AGRAPH(SSAREA,T,CMIN,CMAX,NRECPT,KTER1,KTER2)
      KTER1=10HDIP-SLIP A  $KTER2=10HREA
      CALL AMINMX(DSAREA,NRECPT,CMIN,CMAX)
      CALL AGRAPH(DSAREA,T,CMIN,CMAX,NRECPT,KTER1,KTER2)
      WRITE(7,55) $CALL GETNUM(A) $IF(A(1).EQ.0.)GOTO100
  100 WRITE(7,93)(C(I),I=1,36)
   93 FORMAT(6(6F10.3,/))
      WRITE(7,115)(ICRNR(I),I=1,16)
      WRITE(7,117)(IT(I),I=1,4),ITIME
      WRITE(7,117)KFLAGS,KFLAGD,LFLAG,MFLAG,IFLAG
      WRITE(7,117)NRECPT,N,N1,N2,N3
  115 FORMAT(2(8I6,/))
  117 FORMAT(5I10)
      WRITE(7,66)  $CALL GETNUM(A)
   66 FORMAT(*1=RE-START WITH ALL NEW VALUES*,/,
     1  *2=RE-START WITH NEW STRIKE-SLIP VALUE AND ZONE COORDINATES*,/,
     1  *3=RE-START WITH NEW DIP-SLIP VALUE AND ZONE COORDINATES*,/,
     1  *4=RE-START WITH NEW STRIKE-SLIP VALUE ONLY*,/,
     1  *5=RE-START WITH NEW DIP-SLIP VALUE ONLY*,/,
     1  *6=RE-START WITH NEW TILTMETER COORDINATES ONLY*,/,
     1  *7=STOP*)
      IF(A(1).EQ.1.)GOTO67  $IF(A(1).EQ.2.)GOTO62
      IF(A(1).EQ.3.)GOTO76  $IF(A(1).EQ.4.)GOTO70
      IF(A(1).EQ.5.)GOTO71  $IF(A(1).EQ.6.)GOTO72
      IF(A(1).EQ.7.)GOTO73  $GOTO67
   62 WRITE(7,74)
   74 FORMAT(*U1IN, U1FN*)
      CALL GETNUM(A)  $C(33)=A(1)  $C(34)=A(2)
      WRITE(7,15)$CALL GETNUM(A)
      C(17)=C(19)=A(1)  $C(18)=C(24)=A(2)  $C(20)=C(22)=A(3)
```

```
          C(21)=C(23)=A(4)  $C(25)=C(27)=A(5)  $C(26)=C(32)=A(6)
          C(28)=C(30)=A(7)  $C(29)=C(31)=A(8)  $WRITE(7,3)
          CALL GETNUM(A)  $LFLAG=A(1)  $MFLAG=A(2)
          DO 85 I=1,8
     85   ICRNR(I)=1
          IF(LFLAG.EQ.0.OR.MFLAG.EQ.0)CALL SLPCRN(LFLAG,MFLAG,ICRNR)
          IF(LFLAG.EQ.2)GOTO 86  $GOTO87
     86   DO 88 I=9,16
     88   ICRNR(I)=2
     87   IF(MFLAG.EQ.2)GOTO89  $GOTO90
     89   DO 91 I=1,8
     91   ICRNR(I)=2
     90   GOTO75
     76   WRITE(7,77)
     77   FORMAT(*U3IN, U3FN*)
          CALL GETNUM(A)  $C(35)=A(1)  $C(36)=A(2)
          WRITE(7,17)  $CALL GETNUM(A)
          C(1)=C(3)=A(1)  $C(2)=C(8)=A(2)
          C(4)=C(6)=A(3)  $C(5)=C(7)=A(4)  $C(9)=C(11)=A(5)
          C(10)=C(16)=A(6)  $C(12)=C(14)=A(7)  $C(13)=C(15)=A(8)
          WRITE(7,3)  $CALL GETNUM(A)  $LFLAG=A(1)  $MFLAG=A(2)
          DO 78 I=1,8
     78   ICRNR(I)=1
          IF(LFLAG.EQ.0.OR.MFLAG.EQ.0)CALL SLPCRN(LFLAG,MFLAG,ICRNR)
          IF(LFLAG.EQ.2)GOTO79  $GOTO80
     79   DO 81 I=9,16
     81   ICRNR(I)=2
     80   IF(MFLAG.EQ.2)GOTO82  $GOTO83
     82   DO 84 I=1,8
     84   ICRNR(I)=2
     83   GOTO75
     70   WRITE(7,74) $CALL GETNUM(A) $C(33)=A(1)  $C(34)=A(2)   $GOTO75
     71   WRITE(7,77)  $CALL GETNUM(A)  $C(35)=A(1)  $C(36)=A(2)
          GOTO75
     72   WRITE(7,18)  $CALL GETNUM(A)  $X1=A(1)  $X2=A(2)
          GOTO75
     73   STOP  $END
          SUBROUTINE XPNSHL(P,Q,R1,S1,T,Y)
          S=(T-R1)*6.  $R=S1-R1  $IF(R.EQ.0.)R=1.E-20 $ALPHA=-S/R
          Y=((Q-P)*(1.-EXP(ALPHA)))+P
          RETURN $END
          SUBROUTINE ALINAR(P,Q,R,S,T,Y)
          Y=(((Q-P)*(T-R))/(S-R)) + P
          RETURN  $END
          SUBROUTINE SLPCRN(LFLAG,MFLAG,ICRNR)
          DIMENSION A(20),ICRNR(16)
          WRITE(7,1)
      1   FORMAT(*1=VARIABLE INCREMENTED EXPONENTIALLY*,/,
         1 *2=VARIABLE INCREMENTED LINEARLY*)
          IF(MFLAG.NE.0)GOTO2
          WRITE(7,3) $CALL GETNUM(A) $ICRNR(1)=ICRNR(3)=A(1)
      3   FORMAT(*D1X1, D1X3, D2X3, D3X1*)
          ICRNR(2)=ICRNR(8)=A(2)  $ICRNR(4)=ICRNR(6)=A(3)
          ICRNR(5)=ICRNR(7)=A(4)
      2   IF(LFLAG.NE.0)RETURN
          WRITE(7,4)  $CALL GETNUM(A)  $ICRNR(9)=ICRNR(11)=A(1)
      4   FORMAT(*C1X1, C1X3, C2X3, C3X1*)
          ICRNR(10)=ICRNR(16)=A(2)  $ICRNR(12)=ICRNR(14)=A(3)
```

```
      ICRNR(13)=ICRNR(15)=A(4)  $RETURN  $END
      SUBROUTINE TRIGGR(IT)
      DIMENSION A(20),IT(4)
      WRITE(7,1)  $CALL GETNUM(A)  $IT(1)=A(1)  $WRITE(7,2)
    1 FORMAT(*0 = D(I1) > C(I2),  1 = D(I1) < C(I2)*)
    2 FORMAT(*0 = STRIKE-SLIP/1 = DIP-SLIP ZONE TRIGGERED*)
      CALL GETNUM(A) $IT(2)=A(1)  $WRITE(7,3)
    3 FORMAT(*SPECIFY I1 AND I2*)
      CALL GETNUM(A)  $IT(3)=A(1)  $IT(4)=A(2)
      RETURN  $END
      SUBROUTINE AGRAPH(R,T,A1,B,NRECPT,K1,K2)
      COMMON/JPLOT/XLT,XRT,YLO,YUP,MAJX,MAJY,KX(2),KY(2),
     1 LTITL(8),LU,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
      DIMENSION R(50),T(50),A(20),KX(2),KY(2)
      KX(1)=10HTIME-ARBIT  $KX(2)=10HRARY UNITS
      XLT=T(1)  $XRT=T(NRECPT)  $KY(1)=K1  $KY(2)=K2
      YLO=A1  $YUP=B
   13 WRITE(7,3)  $READ(7,4)CH  $IF(CH.EQ.1HN.OR.CH.EQ.1H )GOTO5
    3 FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
    4 FORMAT(A1)
      WRITE(7,6)  $CALL GETNUM(A)  $XLT=A(1)  $XRT=A(2)
    6 FORMAT(*MIN/MAX X VALUES*)
    5 WRITE(7,7)  $READ(7,4)CH  $IF(CH.EQ.1HN.OR.CH.EQ.1H )GOTO8
      WRITE(7,9)  $CALL GETNUM(A)  $YLO=A(1)  $YUP=A(2)
    7 FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
    9 FORMAT(*MIN/MAX Y VALUES*)
    8 AA=YUP $IF(YLO.EQ.AA)YUP=YUP+1.  $IF(YLO.EQ.AA)YLO=YLO-1.
      WRITE(7,10)KY(1),KY(2)
   10 FORMAT(*SKIP PLOT OF *,2A10,*?*)
      READ(7,4)IJVAR
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(R,T,1,NRECPT)
      WRITE(7,12) $CALL GETNUM(A)  $IRS=A(1)  $IF(IRS.EQ.1)GOTO13
   12 FORMAT(*1 = NEW PLOT, 2 = RETURN*)
      RETURN $END
      SUBROUTINE AMINMX(R,NRECPT,B,A)
      DIMENSION R(50)
      AMIDPT=((R(NRECPT)-R(1))/2.) + R(1)
      A = B =AMIDPT
      DO 1 I=1,NRECPT
      IF(R(I).GT.A)A=R(I)
      IF(R(I).LT.B)B=R(I)
    1 CONTINUE
      RETURN $END
      SUBROUTINE CMPTLT(D,X1,X2,X3,T1,T2)
      DIMENSION D(20)
      A1=A2=A3=A4=B1=B2=B3=B4=0.
      DA1=DA2=DA3=DA4=DB1=DB2=DB3=DB4=0.
      U1=D(17) $U3=D(18)
      IF(U1.EQ.0.)GO TO 1
      CALL TILT(D(17),X1,X2,X3,D(11),D(12),A1,B1)
      CALL TILT(D(17),X1,X2,X3,D(9),D(10),A2,B2)
      CALL TILT(D(17),X1,X2,X3,D(13),D(14),A3,B3)
      CALL TILT(D(17),X1,X2,X3,D(15),D(16),A4,B4)
    1 IF(U3.EQ.0.)GO TO 2
      CALL DPSPTL(D(18),X1,X2,X3,D(3),D(4),DA1,DB1)
      CALL DPSPTL(D(18),X1,X2,X3,D(1),D(2),DA2,DB2)
      CALL DPSPTL(D(18),X1,X2,X3,D(5),D(6),DA3,DB3)
      CALL DPSPTL(D(18),X1,X2,X3,D(7),D(8),DA4,DB4)
```

```
      2 T1=A1-A2-A3+A4+DA1-DA2-DA3+DA4
        T2=B1-B2-B3+B4+DB1-DB2-DB3+DB4
        RETURN $END
        SUBROUTINE TILT(U1,X1,X2,X3,P1,P3,T1,T2)
        R=SQRT((X1-P1)**2+X2**2+(X3-P3)**2)
        RP=R+P3
        T1=(U1/12.5664)*(X2*(X1-P1)*(R*RP-(R+2.*P3)*(2.*R+P3)))/
       1 (R**3*RP**2)
        T2=(U1/12.5664)*(X2**2*(R*RP-(R+2.*P3)*(2.*R+P3))/(R**3*RP**2)
       1 +(R+2.*P3)/(R*RP))
        RETURN $END
        SUBROUTINE DPSPTL(U3,X1,X2,X3,P1,P3,DT1,DT2)
        R=SQRT(((X1-P1)**2)+(X2**2)+((X3-P3)**2))
        DT1=(U3/6.28318)*((X2*P3)/R)*((1./(R**2))-(1./(((X1-P1)
       1 **2)+(X2**2))))
        DT2=(U3/6.28318)*(((X1-P1)*P3)/((X2**2)+(P3**2)))*(((P3**2)
       1 -(X2**2))/(R*((X2**2)+(P3**2)))+((((X1-P1)**2)+(P3**2))
       1 /(R**3))+((X2**2)+(P3**2))/(R*(((X1-P1)**2)+(X2**2))))
        RETURN $END
        SUBROUTINE GETNUM(R)
        DIMENSION R(1),L(80)
        READ(7,9)L $ I=J=0
      6 J=J+1 $ N=P=S=0 $ M=F=1
      5 I=I+1 $ IF(I.GT.80)RETURN $ D=L(I) $ K=4
        IF(D.EQ.38)K=2 $ IF(D.GE.27.A.D.LE.36)K=1
        IF(D.EQ.47)K=3 $ K=K+S $ GOTO(1,2,3,5,1,4,3,4)K
      1 N=N*10+D-27                   $ S=4        $ GOTO 5
      2 M=-1                          $ S=4        $ GOTO 5
      3 P=I                           $ S=4        $ GOTO 5
      4 IF(P.NE.0)F=10.**(I-P-1) $ R(J)=N/F*M $ GOTO 6
      9 FORMAT(80R1)
        END
```

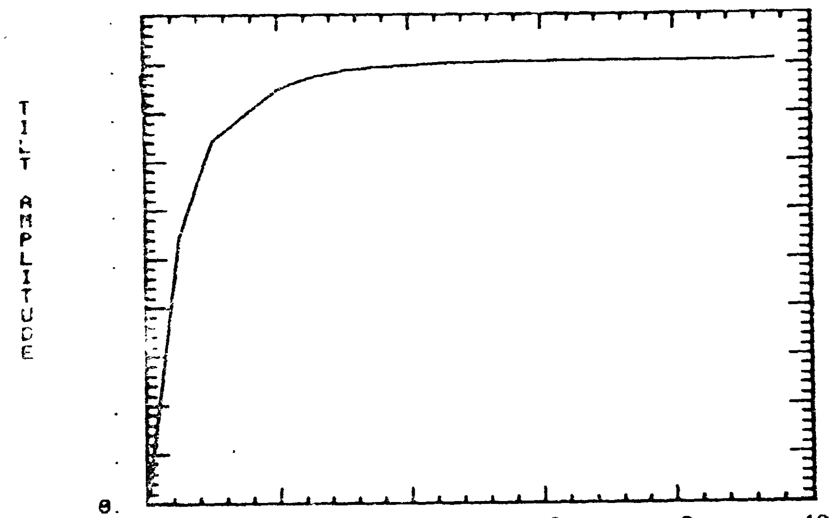
EXAMPLE OF STRTCH

```
^RUN!
1=ZONE EXPANDS, 2=ZONE CONTRACTS
1!
1/2=SLIP INCREMENTED EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
1 1!
0  =INCREMENT CORNERS SEPARATELY
1/2=INCREMENT ALL CORNERS EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
1 1!
'U1>0' = LEFT-LATERAL STRIKE-SLIP
'U3>0' = 'X2>0' SIDE DOWN
U1IN, U1FN, U3IN, U3FN
-1 -1 0 0!
'TRIGGER' OPTION DESIRED?
N!
C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN
0 0 0 0 2 0 1 0!
ENTER STATION COORDINATES--(X1,X2)
1 .5
```



```
!
WRITE PLOT TITLE, 80 CHARACTERS
PROGRAM STRTCH/STRIKE-SLIP = -1 MM.!
MIN/MAX VALUES OF TILT AMPLITUDE =  0.          9.079E-02
SET HORIZONTAL SCALE? Y OR N(=BLANK)
\\\
Y!
MIN/MAX X VALUES
0 10!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
0 .1 !
SKIP PLOT OF TILT AMPLITUDE      ?
```

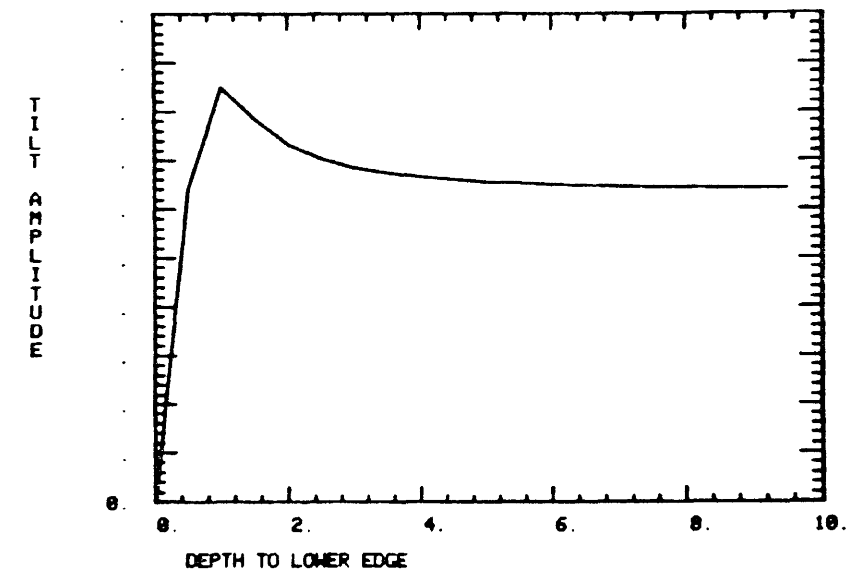
PROGRAM STRTCH/STRIKE-SLIP = -1 MM.
MAXIMUM TILT AMPLITUDE




```
!
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START, 1=SPECIFY NEW DELTAX AND CONTINUE
0!
1=ZONE EXPANDS, 2=ZONE CONTRACTS
1!
1/2=SLIP INCREMENTED EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
1 1!
0  =INCREMENT CORNERS SEPARATELY
1/2=INCREMENT ALL CORNERS EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
1 1!
'U1>0' = LEFT-LATERAL STRIKE-SLIP
'U3>0' = 'X2>0' SIDE DOWN
U1IN, U1FN, U3IN, U3FN
0 0 1 1!
'TRIGGER' OPTION DESIRED?
N!
D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,D3X1FN
0 0 0 0 2 0 1 0!
ENTER STATION COORDINATES--(X1,X2)
1  .5
```

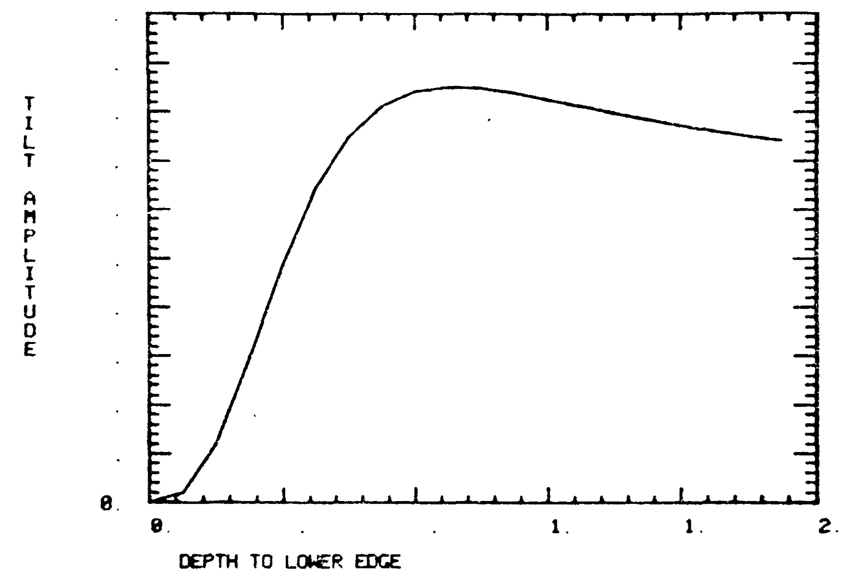
```
!
WRITE PLOT TITLE, 80 CHARACTERS
PROGRAM STRTCH/DIP-SLIP = 1 MM.!
MIN/MAX VALUES OF TILT AMPLITUDE =  0.          4.247E-01
SET HORIZONTAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX X VALUES
0  10!
SET VERTICAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX Y VALUES
0  .5!
SKIP PLOT OF TILT AMPLITUDE      ?
```


PROGRAM STRTCH/DIP-SLIP = 1 MM.
MAXIMUM TILT AMPLITUDE



```
!
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START, 1=SPECIFY NEW DELTAX AND CONTINUE
1!
ENTER NEW DELTAX
.1!
0=RE-START WITH ALL NEW VALUES
1=RE-START WITH PREVIOUS VALUES AND NEW DELTAX
1!
WRITE PLOT TITLE, 80 CHARACTERS
PROGRAM STRTCH/DIP-SLIP=1 MM., DELTAX=.1!
MIN/MAX VALUES OF TILT AMPLITUDE =  0.          4.259E-01
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
0 2!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
0  .5!
SKIP PLOT OF TILT AMPLITUDE      ?
```

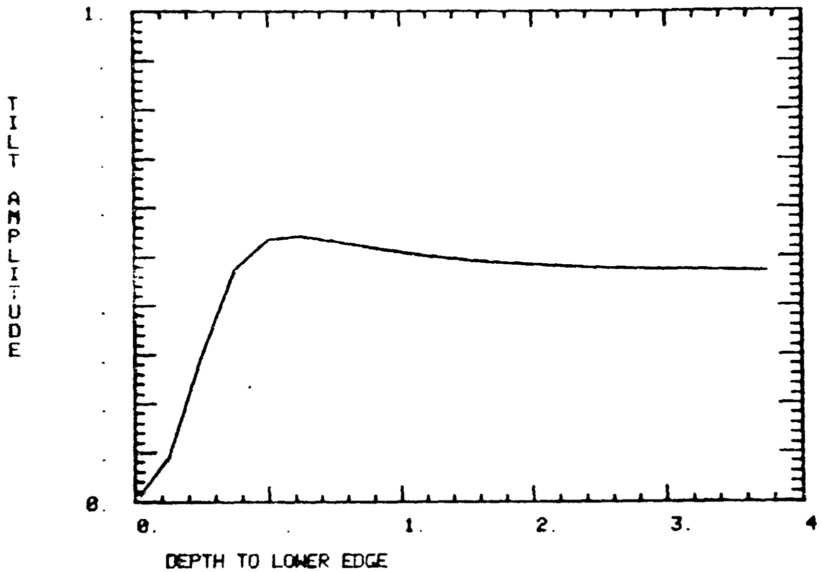
PROGRAM STRTCH/DIP-SLIP=1 MM., DELTAX=.1
MAXIMUM TILT AMPLITUDE



```
!
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START, 1=SPECIFY NEW DELTAX AND CONTINUE
1!
ENTER NEW DELTAX
.2!
0=RE-START WITH ALL NEW VALUES
1=RE-START WITH PREVIOUS VALUES AND NEW DELTAX
0!
1=ZONE EXPANDS, 2=ZONE CONTRACTS
1!
1/2=SLIP INCREMENTED EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
1 1!
0  =INCREMENT CORNERS SEPARATELY
1/2=INCREMENT ALL CORNERS EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
0 0!
1=VARIABLE INCREMENTED EXPONENTIALLY
2=VARIABLE INCREMENTED LINEARLY
D1X1, D1X3, D2X3, D3X1
1 2 1 2!
C1X1, C1X3, C2X3, C3X1
2 1 2 1!
'U1>0' = LEFT-LATERAL STRIKE-SLIP
'U3>0' = 'X2>0' SIDE DOWN
U1IN, U1FN, U3IN, U3FN
0 -5 0 1!
'TRIGGER' OPTION DESIRED?
Y
```

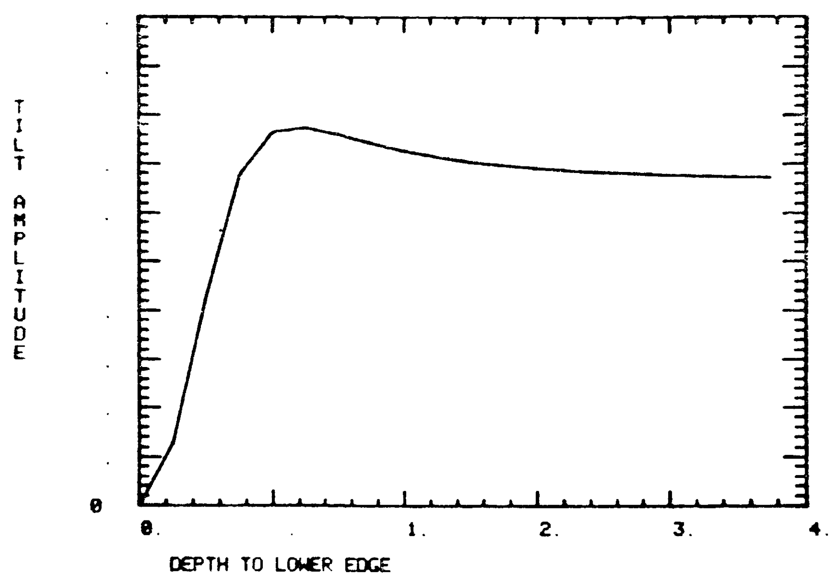
```
!
0 = D(I1) > D(I2),  1 = D(I1) < D(I2)
0!
0 = STRIKE-SLIP/1 = DIP-SLIP ZONE TRIGGERED
1!
SPECIFY I1 AND I2
9 5!
C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN
0 0 0 0 2 0 1 0!
D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,D3X1FN
.5 0 0 .5  1.5 0 1 .5!
ENTER STATION COORDINATES--(X1,X2)
1 .5!
PROGRAM STRTCH/STRIKE-SLIP AND DIP-SLIP!
WRITE PLOT TITLE, 80 CHARACTERS
MIN/MAX VALUES OF TILT AMPLITUDE =  0.                  5.428E-01
MIN/MAX VALUES OF DEPTH          =  0.                  3.000E+00
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
0  4!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
0  1!
SKIP PLOT OF TILT AMPLITUDE      ?
```

PROGRAM STRTCH/STRIKE-SLIP AND DIP-SLIP
MAXIMUM TILT AMPLITUDE



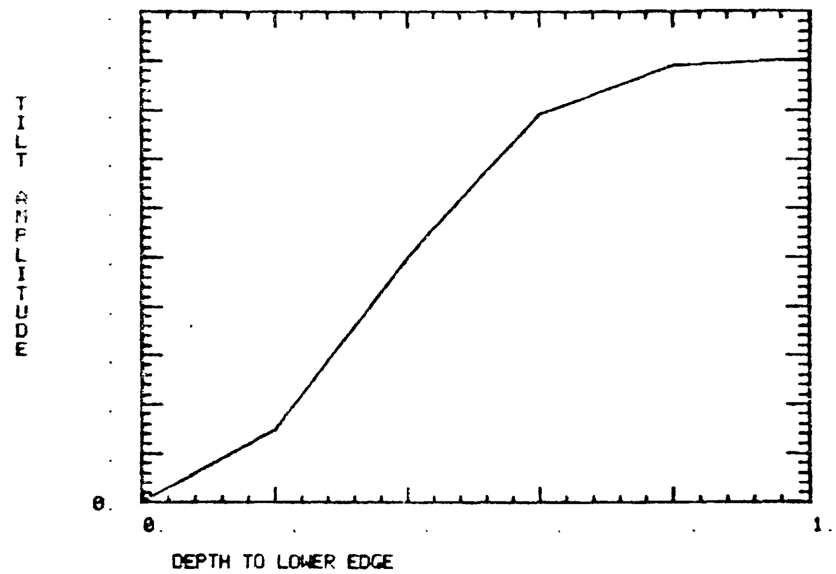

```
 !
1 = NEW PLOT, 2 = RETURN
1!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
0  4!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
0  .7!
SKIP PLOT OF TILT AMPLITUDE      ?
```

PROGRAM STRTCH/STRIKE-SLIP AND DIP-SLIP
MAXIMUM TILT AMPLITUDE



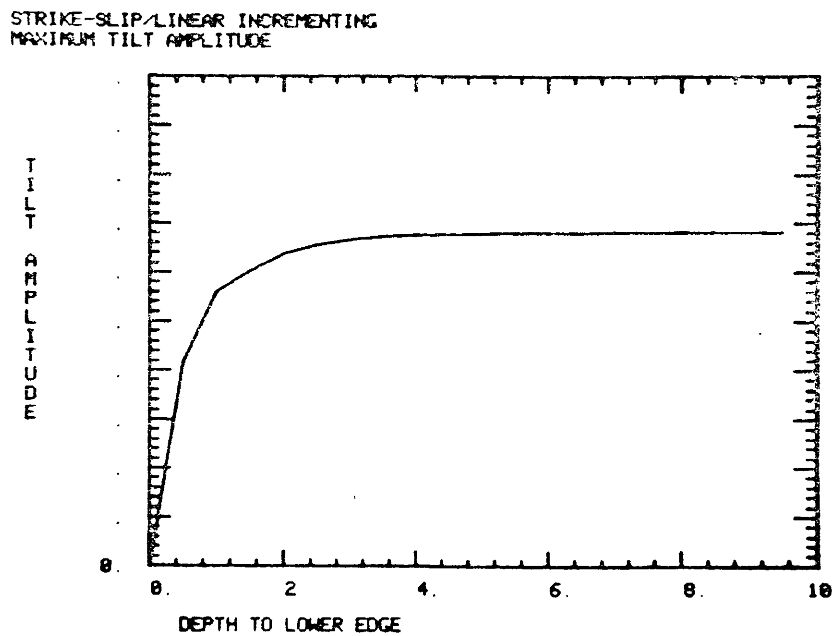

```
 1 = NEW PLOT, 2 = RETURN

1'
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y'
MIN/MAX X VALUES
0  1'
SET VERTICAL SCALE? Y OR N(=BLANK)
Y'
MIN/MAX Y VALUES
0   6'
SKIP PLOT OF TILT AMPLITUDE      ?
```

PROGRAM STRTCH/STRIKE-SLIP AND DIP-SLIP
MAXIMUM TILT AMPLITUDE




```
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START, 1=SPECIFY NEW DELTAX AND CONTINUE
1!
ENTER NEW DELTAX
 .5!
0=RE-START WITH ALL NEW VALUES
1=RE-START WITH PREVIOUS VALUES AND NEW DELTAX
0
```

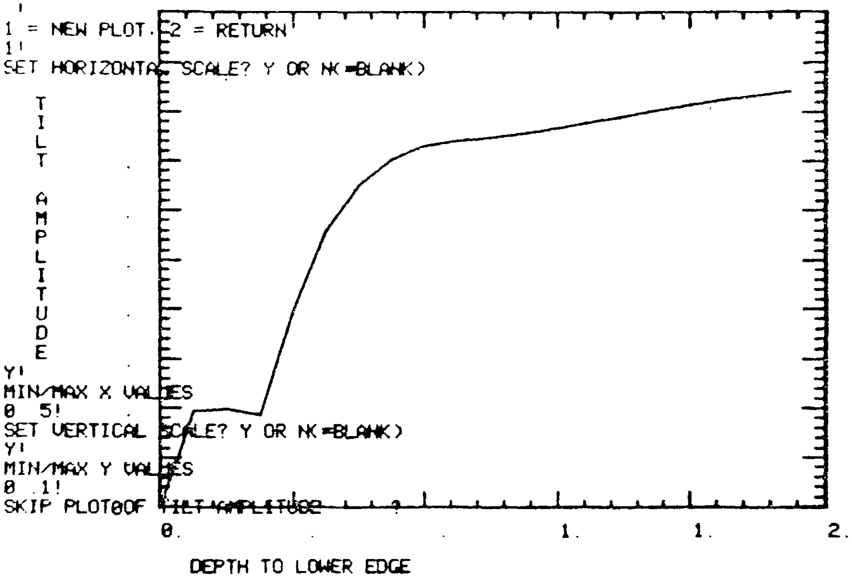
```
1=ZONE EXPANDS, 2=ZONE CONTRACTS
2!
1/2=SLIP INCREMENTED EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
2 2!
0  =INCREMENT CORNERS SEPARATELY
1/2=INCREMENT ALL CORNERS EXPONENTIALLY/LINEARLY
STRIKE-SLIP/DIP-SLIP
2 2!
'U1>0' = LEFT-LATERAL STRIKE-SLIP
'U3>0' = 'X2>0' SIDE DOWN
U1IN, U1FN, U3IN, U3FN
-3 -3 0 0!
'TRIGGER' OPTION DESIRED?
N!
C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN
0 0 0 0 2 0 1 0!
ENTER STATION COORDINATES--(X1,X2)
1 .5!
WRITE PLOT TITLE, 80 CHARACTERS
STRIKE-SLIP/LINEAR INCREMENTING!
MIN/MAX VALUES OF TILT AMPLITUDE =  0.              2.731E-01
MIN/MAX VALUES OF DEPTH          =  0.              9.500E+00
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
0 10!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
0 .4!
SKIP PLOT OF TILT AMPLITUDE      ?
```



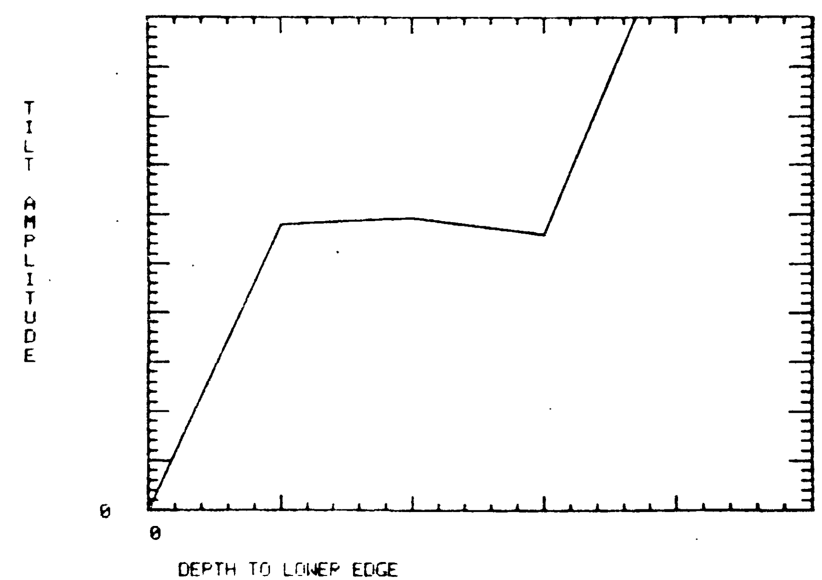
```
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START, 1=SPECIFY NEW DELTAX AND CONTINUE
1!
ENTER NEW DELTAX
.1!
0=RE-START WITH ALL NEW VALUES
1=RE-START WITH PREVIOUS VALUES AND NEW DELTAX
1!
WRITE PLOT TITLE, 80 CHARACTERS
STRIKE-SLIP/LINEAR INCREMENTING/DELTAX=.1!
MIN/MAX VALUES OF TILT AMPLITUDE =  0.            2.526E-01
MIN/MAX VALUES OF DEPTH          =  6.            1.960E+00
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
0 2!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
6   3!
SKIP PLOT OF TILT AMPLITUDE      ?
```

STRIKE-SLIP/LINEAR INCREMENTING/DELTAX= 1
MAXIMUM TILT AMPLITUDE

```
1 = NEW PLOT, 2 = RETURN
1!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX X VALUES
0  5!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
0 .1!
SKIP PLOT0OF TILT AMPLITUDE ?
```



STRIKE-SLIP/LINEAR INCREMENTING DELTAX= 1
MAXIMUM TILT AMPLITUDE




```
1 = NEW PLOT, 2 = RETURN
2!
0=RE-START, 1=SPECIFY NEW DELTAX AND CONTINUE
2!
OK - ^EDIT
```

PROGRAM LISTING FOR STRTCH

```
DELETE(LGO,OUTPUT,STRTCH)
STRTCH.
EXIT.
LIBCOPY(GRAPHIC,TXLGO/RR,TXLGO)
LIBCOPY(JDRAT,NPLGO/RR,NPLGO)
DELETE(LGO,OUTPUT,STRTCH)
RUN76(S)
LINK(F=LGO,F=TXLGO,F=NPLGO,B=STRTCH)
STRTCH.
FIN.
EOR
      PROGRAM STRTCH(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
      COMMON/TVPCOL/TVPUL(8)
      COMMON/TVTUNE/ITUNE(30)
      COMMON/JPLOT/XLT,XRT,YLO,YUP,MAJX,MAJY,KX(2),KY(2),
     1LTITL(8),LU,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
      DIMENSION IFET(8)
      DIMENSION TX(50),TY(50),T(50)
      DIMENSION D(20),C(40),A(30),IT(4),ICRNR(16)
      DIMENSION AMPMAX(50),DEPTH(50)
      DIMENSION THETA1(50,50),THETA2(50,50)
      DIMENSION TAMP(50),TAZM(50)
      CALL FET(5LTAPE7,IFET,8)
      IFET(2)=IFET(2).OR.0000 0010 0000 0000 0000B
      IFET(8)=IFET(8).OR.4000 0000 0000 0000 0000B
      CALL FET(5LTAPE7,IFET,-8)
      KTEST=1 $DELTAX=.5 $CASE=-DELTAX
   67 DO 101 J=1,18
      D(J)=0.  $C(J+18)=0. $C(J)=0.
  101 CONTINUE
      WRITE(7,1)
    1 FORMAT(*1=ZONE EXPANDS, 2=ZONE CONTRACTS*)
      CALL GETNUM(A) $IFLAG=A(1) $WRITE(7,2)
    2 FORMAT(*1/2=SLIP INCREMENTED EXPONENTIALLY/LINEARLY*,/,
     1         *STRIKE-SLIP/DIP-SLIP*)
      CALL GETNUM(A) $KFLAGS=A(1) $KFLAGD=A(2) $WRITE(7,3)
    3 FORMAT(*0  =INCREMENT CORNERS SEPARATELY*,/,
     1         *1/2=INCREMENT ALL CORNERS EXPONENTIALLY/LINEARLY*,/,
     1         *STRIKE-SLIP/DIP-SLIP*)
      CALL GETNUM(A) $LFLAG=A(1) $MFLAG=A(2)
      DO 110 I=1,4
      IT(I)=5
  110 CONTINUE
      DO 4 I=1,16
    4 ICRNR(I)=1
      IF(LFLAG.EQ.0.OR.MFLAG.EQ.0)CALL SLPCRN(LFLAG,MFLAG,ICRNR)
      IF(LFLAG.EQ.2)GOTO 5 $GOTO6
    5 DO 7 I=9,16
    7 ICRNR(I)=2
    6 IF(MFLAG.EQ.2)GOTO8 $GOTO9
    8 DO 10 I=1,8
   10 ICRNR(I)=2
    9 WRITE(7,11) $CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
   11 FORMAT(*'U1>0' = LEFFT-LATERAL STRIKE-SLIP*,/,
     1         *'U3>0' = 'X2>0' SIDE DOWN*,/,
     1         *U1IN, U1FN, U3IN, U3FN*)
      C(35)=A(3) $C(36)=A(4) $WRITE(7,12)
   12 FORMAT(*'TRIGGER' OPTION DESIRED?*)
```

```
      READ(7,13)TRIG $IF(TRIG.EQ.1HY)CALL TRIGGR(IT)
   13 FORMAT(A1)
      IF(C(33).EQ.0..AND.C(34).EQ.0.)GOTO14
      WRITE(7,15) $CALL GETNUM(A) $C(17)=C(19)=A(1)
   15 FORMAT(*C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN*)
      C(18)=C(24)=A(2) $C(20)=C(22)=A(3) $C(21)=C(23)=A(4)
      C(25)=C(27)=A(5) $C(26)=C(32)=A(6) $C(28)=C(30)=A(7)
      C(29)=C(31)=A(8)
   14 IF(C(35).EQ.0..AND.C(36).EQ.0.)GOTO16
      WRITE(7,17) $CALL GETNUM(A) $C(1)=C(3)=A(1) $C(2)=C(8)=A(2)
   17 FORMAT(*D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,D3X1FN*)
      C(4)=C(6)=A(3) $C(5)=C(7)=A(4) $C(9)=C(11)=A(5)
      C(10)=C(16)=A(6) $C(12)=C(14)=A(7) $C(13)=C(15)=A(8)
   16 WRITE(7,18) $CALL GETNUM(A) $X1=A(1) $X2=A(2)
   18 FORMAT(*ENTER STATION COORDINATES--(X1,X2)*)
      X3=0. $NRECPT=50 $N=(.075*NRECPT) $NSLPPT=NRECPT
      N1=N+1 $N2=NRECPT-(.075*NRECPT)-1
      N3=N2+1  $PI=3.1415926 $DEPTHL=CASE
      RNSLPP=NSLPPT $RN1=N1 $RN2=N2 $ONE=1. $ITIME=99999
   75 CONTINUE
      DEPTHL=DEPTHL+DELTAX  $C(28)=DEPTHL  $C(20)=C(22)=C(30)=C(28)
      C(4)=C(12)=C(6)=C(14)=C(28)
      DO  21 J=1,8
      D(J)=C(J)  $D(J+8)=C(J+16)
   21 CONTINUE
      D(17)=C(33)  $D(18)=C(35)
      DO 19 I=1,N
      T(I)=I
      DO 63 K=1,NSLPPT
      CALL CMPTLT(D,X1,X2,X3,T1,T2)
      THETA1(I,K)=T1 $THETA2(I,K)=T2
   63 CONTINUE
   19 CONTINUE
      DO 24 J=1,8
      D(J)=C(J+8)  $D(J+8)=C(J+24)
   24 CONTINUE
      D(17)=C(34)  $D(18)=C(36)
      DO 23 I=N3,NRECPT
      T(I)=I
      DO 64 K=1,NSLPPT
      RK=K $CALL XPNSHL(C(33),C(34),ONE,RNSLPP,RK,Y1)
      CALL XPNSHL(C(35),C(36),ONE,RNSLPP,RK,Y2)
      IF(KFLAGS.NE.1)CALL ALINAR(C(33),C(34),ONE,RNSLPP,RK,Y1)
      IF(KFLAGD.NE.1)CALL ALINAR(C(35),C(36),ONE,RNSLPP,RK,Y2)
      D(17)=Y1  $D(18)=Y2
      CALL CMPTLT(D,X1,X2,X3,T1,T2)
      THETA1(I,K)=T1 $THETA2(I,K)=T2
   64 CONTINUE
   23 CONTINUE
      IFGREC=0
      DO 25 I=N1,N2
      T(I)=I  $RI=I
      DO 65 K=1,NSLPPT
      RK=K
      IF(TRIG.EQ.1HY)GOTO26
      DO 27 J=1,8
      J8=J+8 $A1=C(J) $B=C(J8) $E=C(J8+8) $F=C(J+24)
      CALL XPNSHL(A1,B,RN1,RN2,RI,Y1)
```

```
      CALL XPNSHL(E,F,RN1,RN2,RI,Y2)
      IF(ICRNR(J).EQ.2)CALL ALINAR(A1,B,RN1,RN2,RI,Y1)
      IF(ICRNR(J8).EQ.2)CALL ALINAR(E,F,RN1,RN2,RI,Y2)
      D(J)=Y1 $D(J8)=Y2
   27 CONTINUE
      CALL XPNSHL(C(33),C(34),ONE,RNSLPP,RK,Y1)
      CALL XPNSHL(C(35),C(36),ONE,RNSLPP,RK,Y2)
      IF(KFLAGS.NE.1)CALL ALINAR(C(33),C(34),ONE,RNSLPP,RK,Y1)
      IF(KFLAGD.NE.1)CALL ALINAR(C(35),C(36),ONE,RNSLPP,RK,Y2)
      D(17)=Y1 $D(18)=Y2 $GOTO28
   26 IF(IT(2).EQ.1.AND.KFLAGS.EQ.1)CALL XPNSHL(C(33),C(34),ONE,
     1 RNSLPP,RK,Y)
      IF(IT(2).EQ.1.AND.KFLAGS.NE.1)CALL ALINAR(C(33),C(34), ONE,
     1 RNSLPP,RK,Y)
      IF(IT(2).EQ.0.AND.KFLAGD.EQ.1)CALL XPNSHL(C(35),C(36),ONE,
     1 RNSLPP,RK,Y)
      IF(IT(2).EQ.0.AND.KFLAGD.NE.1)CALL ALINAR(C(35),C(36),ONE,
     1 RNSLPP,RK,Y)
      IF(IT(2).EQ.0)D(18)=Y $IF(IT(2).EQ.1)D(17)=Y
      DO 32 J=1,8
      J8=J+8 $A1=C(J) $B=C(J8) $E=C(J8+8) $F=C(J+24)
      CALL XPNSHL(A1,B,RN1,RN2,RI,Y1)
      CALL XPNSHL(E,F,RN1,RN2,RI,Y2)
      IF(ICRNR(J).EQ.2)CALL ALINAR(A1,B,RN1,RN2,RI,Y1)
      IF(ICRNR(J8).EQ.2)CALL ALINAR(E,F,RN1,RN2,RI,Y2)
      D(J)=Y1 $D(J8)=Y2
   32 CONTINUE
      MIT3=IT(3)  $MIT4=IT(4) $IF(IT(1).EQ.1)GOTO29
      IF(D(MIT3).LT.C(MIT4).AND.IT(2).EQ.0)D(17)=C(33)
      IF(D(MIT3).LT.C(MIT4).AND.IT(2).EQ.1)D(18)=C(35)
      IF(D(MIT3).LT.C(MIT4).AND.IT(2).EQ.1)GOTO33
      IF(D(MIT3).GE.C(MIT4))GOTO35
      DO 40 J=9,16
   40 D(J)=C(J+8)
      GOTO28
   29 IF(D(MIT3).GT.C(MIT4).AND.IT(2).EQ.0)D(17)=C(33)
      IF(D(MIT3).GT.C(MIT4).AND.IT(2).EQ.1)D(18)=C(35)
      IF(D(MIT3).GT.C(MIT4).AND.IT(2).EQ.1)GOTO33
      IF(D(MIT3).LE.C(MIT4))GOTO35
      DO 34 J=9,16
   34 D(J)=C(J+8)
      GOTO28
   33 DO 36 J=1,8
   36 D(J)=C(J)
      GOTO28
   35 IFGREC=IFGREC+1
      IF(IFGREC.EQ.1)ITIME=I
      IF(IFGREC.EQ.1)RITIME=ITIME
      IF(IT(2).EQ.0.AND.KFLAGS.EQ.1)CALL XPNSHL(C(33),C(34),ONE,
     1 RNSLPP,RK,Y)
      IF(IT(2).EQ.0.AND.KFLAGS.NE.1)CALL ALINAR(C(33),C(34),ONE,
     1 RNSLPP,RK,Y)
      IF(IT(2).EQ.1.AND.KFLAGD.EQ.1)CALL XPNSHL(C(35),C(36),ONE,
     1 RNSLPP,RK,Y)
      IF(IT(2).EQ.1.AND.KFLAGD.NE.1)CALL ALINAR(C(35),C(36),ONE,
     1 RNSLPP,RK,Y)
      IF(IT(2).EQ.1)D(18)=Y $IF(IT(2).EQ.0)D(17)=Y
      IF(IT(2).EQ.1)GOTO37
```

```
      DO 38 J=9,16
      A1=C(J+8)  $B=C(J+16) $CALL XPNSHL(A1,B,RITIME,RN2,RI,Y)
      IF(ICRNR(J).EQ.2)CALL ALINAR(A1,B,RITIME,RN2,RI,Y)
 38   D(J)=Y
      GOTO28
 37   DO 39 J=1,8
      E=C(J) $F=C(J+8) $CALL XPNSHL(E,F,RITIME,RN2,RI,Y)
      IF(ICRNR(J).EQ.2)CALL ALINAR(E,F,RITIME,RN2,RI,Y)
 39   D(J)=Y
 28   CALL CMPTLT(D,X1,X2,X3,T1,T2)
      THETA1(I,K)=T1 $THETA2(I,K)=T2
 65   CONTINUE
  25  CONTINUE
      DO 41 I=1,NRECPT
      SUM11=SUM12=SUM21=SUM22=0.
      DO 42 J=1,I
      AT=THETA1(J,I+1-J) $B=THETA2(J,I+1-J) $SUM11=AT+SUM11
      SUM21=B+SUM21
 42   CONTINUE
      LM=I-1 $IF(LM.LT.1)GOTO43
      DO 44 J=1,LM
      AT=THETA1(J,I-J) $B=THETA2(J,I-J) $SUM12=AT+SUM12 $SUM22=B+SUM22
  44  CONTINUE
 43   TX(I)=SUM21-SUM22 $TY(I)=SUM11-SUM12
 41   CONTINUE
      IF(IFLAG.EQ.2)GOTO46
      AR=TX(1) $B=TY(1)
      DO 47 I=1,NRECPT
      TX(I)=TX(I)-AR
 47   TY(I)=TY(I)-B
      GOTO48
  46  DO 49 I=1,NRECPT
      TX(I)=TX(NRECPT)-TX(I)
  49  TY(I)=TY(NRECPT)-TY(I)
 48   CONTINUE
      DO 50 I=1,NRECPT
      TAMP(I)=SQRT((TX(I)**2)+(TY(I)**2))
  50  CONTINUE
      CALL AMINMX(TAMP,NRECPT,TAMPMN,TAMPMX)
      AMPMAX(KTEST)=TAMPMX
      DEPTH(KTEST)=C(28)
      KTEST=KTEST+1
      IF(KTEST.LE.20)GOTO75
      KTEST=20  $DEPTHL=CASE
      LU=7 $LNLGX=1 $LNLGY=1 $NCLX=2 $NCLY=2
      DO 200 KM=4,8
 200  LTITL2(KM)=10H
      LTITL2(1)=10HMAXIMUM TI  $LTITL2(2)=10HLT AMPLITU
      LTITL2(3)=10HDE
      WRITE(7,53) $READ(7,54)(LTITL(I),I=1,8)
 53   FORMAT(*WRITE PLOT TITLE, 80 CHARACTERS*)
 54   FORMAT(8A10)
      MAJX=5 $MAJY=10
      CALL AMINMX(AMPMAX,KTEST,A1,A2)
      CALL AMINMX(DEPTH,KTEST,B1,B2)
      WRITE(7,52)A1,A2,B1,B2
 52   FORMAT(*MIN/MAX VALUES OF TILT AMPLITUDE = *,E10.3,5X,E10.3,/,
     1        *MIN/MAX VALUES OF DEPTH          = *,E10.3,5X,E10.3)
```

```
      68 KTER1=10HTILT AMPLI  $KTER2=10HTUDE
         CALL AGRAPH(AMPMAX,DEPTH,A1,A2,KTEST,KTER1,KTER2)
         WRITE(7,1234) $CALL GETNUM(A) $IF(A(1).EQ.1.)WRITE(7,1235)
   1234  FORMAT(*0=RE-START, 1=SPECIFY NEW DELTAX AND CONTINUE*)
    1235 FORMAT(*ENTER NEW DELTAX*)
         KTEST=1 $IF(A(1).EQ.0.)GOTO67 $IF(A(1).EQ.1.)GOTO1236
         GOTO73
    1236 CALL GETNUM(A)  $DELTAX=A(1) $DEPTHL=CASE=-DELTAX  $KTEST=1
         WRITE(7,1237) $CALL GETNUM(A) $IF(A(1).EQ.0.)GOTO67
   1237  FORMAT(*0=RE-START WITH ALL NEW VALUES*,/,
        1 *1=RE-START WITH PREVIOUS VALUES AND NEW DELTAX*)
         GOTO75
      73 STOP  $END
         SUBROUTINE XPNSHL(P,Q,R1,S1,T,Y)
         S=(T-R1)*6.  $R=S1-R1  $IF(R.EQ.0.)R=1.E-20 $ALPHA=-S/R
         Y=((Q-P)*(1.-EXP(ALPHA)))+P
         RETURN $END
         SUBROUTINE ALINAR(P,Q,R,S,T,Y)
         Y=(((Q-P)*(T-R))/(S-R)) + P
         RETURN  $END
         SUBROUTINE SLPCRN(LFLAG,MFLAG,ICRNR)
         DIMENSION A(20),ICRNR(16)
         WRITE(7,1)
     1   FORMAT(*1=VARIABLE INCREMENTED EXPONENTIALLY*,/,
        1 *2=VARIABLE INCREMENTED LINEARLY*)
         IF(MFLAG.NE.0)GOTO2
         WRITE(7,3) $CALL GETNUM(A) $ICRNR(1)=ICRNR(3)=A(1)
     3   FORMAT(*D1X1, D1X3, D2X3, D3X1*)
         ICRNR(2)=ICRNR(8)=A(2)  $ICRNR(4)=ICRNR(6)=A(3)
         ICRNR(5)=ICRNR(7)=A(4)
     2   IF(LFLAG.NE.0)RETURN
         WRITE(7,4)  $CALL GETNUM(A)  $ICRNR(9)=ICRNR(11)=A(1)
     4   FORMAT(*C1X1, C1X3, C2X3, C3X1*)
         ICRNR(10)=ICRNR(16)=A(2)  $ICRNR(12)=ICRNR(14)=A(3)
         ICRNR(13)=ICRNR(15)=A(4)  $RETURN  $END
         SUBROUTINE TRIGGR(IT)
         DIMENSION A(20),IT(4)
         WRITE(7,1)  $CALL GETNUM(A)  $IT(1)=A(1)  $WRITE(7,2)
     1   FORMAT(*0 = D(I1) > C(I2),  1 = D(I1) < C(I2)*)
      2  FORMAT(*0 = STRIKE-SLIP/1 = DIP-SLIP ZONE TRIGGERED*)
         CALL GETNUM(A) $IT(2)=A(1)  $WRITE(7,3)
     3   FORMAT(*SPECIFY I1 AND I2*)
         CALL GETNUM(A)  $IT(3)=A(1)  $IT(4)=A(2)
         RETURN  $END
         SUBROUTINE AGRAPH(R,T,A1,B,NRECPT,K1,K2)
         COMMON/JPLOT/XLT,XRT,YLO,YUP,MAJX,MAJY,KX(2),KY(2),
        1 LTITL(8),LU,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
         DIMENSION R(50),T(50),A(20),KX(2),KY(2)
         KX(1)=10HDEPTH TO L  $KX(2)=10HOWER EDGE
         XLT=T(1)  $XRT=T(NRECPT)  $KY(1)=K1  $KY(2)=K2
         YLO=A1  $YUP=B
      13 WRITE(7,3)  $READ(7,4)CH  $IF(CH.EQ.1HN.OR.CH.EQ.1H )GOTO5
     3   FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
     4   FORMAT(A1)
         WRITE(7,6)  $CALL GETNUM(A)  $XLT=A(1)  $XRT=A(2)
      6  FORMAT(*MIN/MAX X VALUES*)
     5   WRITE(7,7)  $READ(7,4)CH  $IF(CH.EQ.1HN.OR.CH.EQ.1H )GOTO8
         WRITE(7,9)  $CALL GETNUM(A)  $YLO=A(1)  $YUP=A(2)
```

```
 7    FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
 9    FORMAT(*MIN/MAX Y VALUES*)
 8    AA=YUP $IF(YLO.EQ.AA)YUP=YUP+1.  $IF(YLO.EQ.AA)YLO=YLO-1.
      WRITE(7,10)KY(1),KY(2)
 10   FORMAT(*SKIP PLOT OF *,2A10,*?*)
      READ(7,4)IJVAR
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(R,T,1,NRECPT)
      WRITE(7,12) $CALL GETNUM(A)  $IRS=A(1)  $IF(IRS.EQ.1)GOTO13
 12   FORMAT(*1 = NEW PLOT, 2 = RETURN*)
      RETURN $END
      SUBROUTINE AMINMX(R,NRECPT,B,A)
      DIMENSION R(50)
      AMIDPT=((R(NRECPT)-R(1))/2.) + R(1)
      A = B =AMIDPT
      DO 1 I=1,NRECPT
      IF(R(I).GT.A)A=R(I)
      IF(R(I).LT.B)B=R(I)
  1   CONTINUE
      RETURN $END
      SUBROUTINE CMPTLT(D,X1,X2,X3,T1,T2)
      DIMENSION D(20)
      A1=A2=A3=A4=B1=B2=B3=B4=0.
      DA1=DA2=DA3=DA4=DB1=DB2=DB3=DB4=0.
      U1=D(17) $U3=D(18)
      IF(U1.EQ.0.)GO TO 1
      CALL TILT(D(17),X1,X2,X3,D(11),D(12),A1,B1)
      CALL TILT(D(17),X1,X2,X3,D(9),D(10),A2,B2)
      CALL TILT(D(17),X1,X2,X3,D(13),D(14),A3,B3)
      CALL TILT(D(17),X1,X2,X3,D(15),D(16),A4,B4)
    1 IF(U3.EQ.0.)GO TO 2
      CALL DPSPTL(D(18),X1,X2,X3,D(3),D(4),DA1,DB1)
      CALL DPSPTL(D(18),X1,X2,X3,D(1),D(2),DA2,DB2)
      CALL DPSPTL(D(18),X1,X2,X3,D(5),D(6),DA3,DB3)
      CALL DPSPTL(D(18),X1,X2,X3,D(7),D(8),DA4,DB4)
    2 T1=A1-A2-A3+A4+DA1-DA2-DA3+DA4
      T2=B1-B2-B3+B4+DB1-DB2-DB3+DB4
      RETURN $END
      SUBROUTINE TILT(U1,X1,X2,X3,P1,P3,T1,T2)
      R=SQRT((X1-P1)**2+X2**2+(X3-P3)**2)
      RP=R+P3
      T1=(U1/12.5664)*(X2*(X1-P1)*(R*RP-(R+2.*P3)*(2.*R+P3)))/
     1 (R**3*RP**2)
      T2=(U1/12.5664)*(X2**2*(R*RP-(R+2.*P3)*(2.*R+P3))/(R**3*RP**2)
     1 +(R+2.*P3)/(R*RP))
      RETURN $END
      SUBROUTINE DPSPTL(U3,X1,X2,X3,P1,P3,DT1,DT2)
      R=SQRT(((X1-P1)**2)+(X2**2)+((X3-P3)**2))
      DT1=(U3/6.28318)*((X2*P3)/R)*((1./(R**2))-(1./(((X1-P1)
     1 **2)+(X2**2))))
      DT2=(U3/6.28318)*(((X1-P1)*P3)/((X2**2)+(P3**2)))*(((P3**2)
     1 -(X2**2))/(R*((X2**2)+(P3**2)))+((((X1-P1)**2)+(P3**2))
     1 /(R**3))+((X2**2)+(P3**2))/(R*(((X1-P1)**2)+(X2**2))))
      RETURN $END
      SUBROUTINE GETNUM(R)
      DIMENSION R(1),L(80)
      READ(7,9)L $ I=J=0
    6 J=J+1 $ N=P=S=0 $ M=F=1
    5 I=I+1 $ IF(I.GT.80)RETURN $ D=L(I) $ K=4
```

```
      IF(D.EQ.38)K=2 $ IF(D.GE.27.A.D.LE.36)K=1
      IF(D.EQ.47)K=3 $ K=K+S $ GOTO(1,2,3,5,1,4,3,4)K
    1 N=N*10+D-27                  $ S=4       $ GOTO 5
    2 M=-1                         $ S=4       $ GOTO 5
    3 P=I                          $ S=4       $ GOTO 5
    4 IF(P.NE.0)F=10.**(I-P-1) $ R(J)=N/F*M $ GOTO 6
    9 FORMAT(80R1)
      END
```