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) 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

by

Stuart McHugh

Open-file Report No. 76-484

ABSTRACT

The material in this report can be grouped into two categories: 1) programs that compute tilts produced by a vertically oriented expanding rectangular dislocation loop in an elastic or viscoelastic material and 2) programs that compute the shear stresses, strains, and shear displacements in a three-phase half-space (i.e. a half-space containing a vertical slab). Each section describes the relevant theory, and provides a detailed guide to the operation of the programs. A series of examples is provided at the end of each section.

INTRODUCTION

Programs XPND, XPND01, and XPND02 compute the tilts at the free surface of an elastic medium produced by an expanding strike-slip or dip-slip zone, or both, using the equations in Press (1965). The initial and final positions of each zone, the amount and type of slip, station position, and a series of flags must be specified. In addition, one of the expanding zones may trigger the other at a predetermined position. The associated tilts are quasi-static; that is, the only function of time is to label the position of the zone.

Program MAXVGT computes the quasi-static tilt waveforms, at the surface of a Maxwell or Voigt viscoelastic material, produced by an expanding dislocation loop using the equations in Rosenman and Singh (1973). In operation, the program is quite similar to the XPND programs except that the tilt amplitudes will decay as a function of time after the introduction of a static slip zone. If the slip zone grows (ie., the amount of slip or the position of the zone change in time), the tilt waveshapes seen at a station on the surface will depend upon the source-station geometry and the material's time constants and will no longer be a simple exponential decay. The effect of the time dependent material response on the tilts may be examined by changing the Maxwell or Voigt time constants.

Programs ST3PHS, ST3PSZ, SHRSTN, STNRML, and DBLNRM compute the shear stresses, strains, and shear displacements produced by screw dislocations in a vertical slab in a three-phase half-space using the equations in Chou (1966). Program ST3PHS computes and displays the numerical magnitudes of shear stress, strain, and displacement as a function of distance using a single screw dislocation, and ST3PSZ computes the same quantities for a finite width slip zone

(modeled by two screw dislocations, with antiparallel Burger's vectors, separated by some finite vertical dimension). STNRML and DBLNRM compute and display these quantities normalized by their homogeneous half-space values as a function of distance using a single dislocation and a pair of dislocations respectively. SHRSTN computes and displays the normalized shear stress and strain at a point produced by a single screw dislocation, or the maximum or minimum normalized values along some profile, as a function of shear modulus ratio.

Sections 1, 2, and 3 describe the operation of the XPND group, the MAXVGT program, and the ST3PHS group respectively. Examples of the programs' operation and a listing of each program are provided at the end of each section.

CONTENTS PAGE

XPND, XPND01, XPND02

Documentation for Programs

XPND/XPNDØ1/XPNDØ2

These programs compute the quasi-static tilt change associated with an expanding vertically oriented rectangular dislocation loop embedded in an elastic half-space using the equations for a static dislocation loop in Press (1965). The input consists of the initial and final slip zone coordinates, slip magnitudes, station coordinates, and a series of flags that allow various options to be selected. The output consists of the tilt component amplitude, total tilt amplitude, and tilt azimuth versus time; progr m XPNDØ1 also outputs the position of two of the slip zone corners versus time, the slip (both strike-slip and dip-slip) versus time, and the slip area versus time (thereby allowing the operator to monitor the program).

Program Access and Use

The programs are intended for use on the LBL 6600B or 6600C machines and the Tektronix (4010-1) terminal and will automatically link to the appropriate plotting subroutines for interactive graphics. The programs are stored on the library MCHUGH as subsets XPND and XPNDØ1. After logging into the Tektronix (with approximately 70K of core), the programs may be accessed using:

↑LOAD,XPND,MCHUGH

or ↑LOAD,XPNDØ1,MCHUGH

Once loaded, they are ready for use so that an ↑RUN command may directly follow the ↑LOAD command.

Geometry

The input and output geometry is the same as in Press (1965). The x1 axis is coincident with the strike of the fault, the x2 axis is the horizontal axis perpendicular to x1, and the x3 axis is vertical with its positive direction into the medium. In these programs x1 increases to the northwest and x2 increases to the northeast. The angle between x1 and north is assumed to be 45 degrees (figure 1).

The slip zone is at all times rectangular such that a vector drawn from corner #4 to corner #1 points in the x1>0 direction. The corners are labelled clockwise from corner #1 (figure 2). There are no restrictions on the initial or final positions of the zone once the above constraints have been applied (although probably all "practical" problems will require that the slip zone as it expands will always enclose all previous slip zones). If the initial slip zone configuration is not a line or a point, the slip zone will appear to be expanding away from a "hole" in the slip distribution (App. A, p. A13).

Strike-slip or dip-slip or both may be selected. The slip is held constant over the zone at any instant, but may vary in time either linearly or exponentially. The strike-slip and dip-slip zones need not coincide and in addition use of the 'TRIGGER' option allows the growth of one zone to be initiated only when the other zone has reached a pre-specified point (App. B).

The instrument or station position is restricted to the surface, but there are no other constraints on its location.

Procedure and Input:

The computer will automatically request the information it needs. The first response after the ↑RUN command is used is:

a) 1 = Zone expands

2 = Zone contracts

If 1 is entered the zone grows from its initial to its final position; a 2 reverses the growth (i.e., the zone will "collapse" inward or "shrink").

Then:

b) 1 = strike-slip component incremented exponentially

2 = dip-slip component incremented exponentially

3 = both slip components incremented exponentially

4 = neither slip component incremented exponentially

If a 1, 2, or 3 is entered, the appropriate slip component is incremented in a $(1 - e^{-t/\tau})$ fashion from its initial to final slip value; otherwise the slip is incremented in a linear fashion. The time constant, τ, is internally set such that the exponential has decayed to 0.002 when the slip zone coordinates have reached their final value. If the 'TRIGGER' option is used, the one slip component (i.e., corresponding to the triggered zone) remains at zero until that zone's growth is initiated; the time constant of the triggered zone is set so that the slip in this zone has reached 99.998% of its final value when the zone's growth stops.

c) Do you wish the 'TRIGGER' option

'Yes' or 'No'

1) If 'yes' is entered, the response is:

Specify "TRIGGER' parameters in the form

D(M1) .GE. C(M2)

(".GE." is equivalent to "greater than or equal to")

D(M1) and C(M2) are explained in Appendixes A and B. Once M1 and M2 are entered, the response is:

Specify which slip component is to be = 0.

when D (M1) .LT. C(M2)

Strike-slip = S, Dip-slip = D

(".LT." is equivalent to "less than").

After an S or D is entered, the program proceeds to part d below.

2) If 'no' is entered, the steps in section c-1 are skipped and the program moves to part d.

d) Specify initial and final slip values

Right-lateral strike-slip is < 0.

Dip-slip > 0. for 'X2 > 0. side' down

Format = U1IN, U1FN, U3IN, U3FN

The initial (IN) and final (FN) values of the strike-slip (U1) and/or dip-slip (U3) components are entered in the order given above. If both U1IN and U1FN are zero or U3IN and U3FN are zero the computer will not ask for that zone's coordinates.

e) Theta = angle between strike of fault and north = 45.000 degrees

No action required. If it is desired to change theta, line 47 (XPND) or 48 (XPNDØ1) in the main program must be altered before running the program.

f) Either part 1 or part 2 or both (below) will be activated depending upon whether or not both initial and final slip values were zero as discussed in part d.

1) Specify initial and final coordinates

of dip-slip zone corners as indicated

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

2) Specify initial and final values

of strike-slip zone coordinates as indicated

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

The appropriate initial (IN) and final (FN) positions (X1,X3) of the corners (D1, D2, D3, C1, C2, C3) must be entered.

g) Coordinates of station (X1, X2)

The instrument location in the X1-X2 plane must be specified.

h) This section only occurs when running program XPNDØ1.

Specify 2 corners of dislocation surface for display

1 = D1X2 = D2X1 2 = D1X3 = D4X3

4 = D2X3 = D3X3 5 = D3X1 = D4X1

9 = C1X1 = C2X1 10 = C1X3 = C4X3

12 = C2X3 = C3X3 13 = C3X1 = C4X1

Entering the numbers corresponding to the 2 corners will cause the position of those corners as a function of time to be displayed. Failure to enter the 2 numbers will terminate the program.

Output:

i) 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

Entering 0 will cause the program to start over, 1 allows the program to continue.

j) Write plot title, 80 characters

Enter any alpha-numeric title.

k) The plots are automatically labelled. The graphs occur in the following sequence:

1) east-west component versus time
2) north-south component versus time
3) total tilt amplitude versus time
4) tilt azimuth versus time

XPNDØ1

5) U1 component of slip versus time
6) U3 component of slip versus time
7) first corner's position versus time
8) second corner's position versus time
9) area of strike-slip zone versus time
10) area of dip-slip zone versus time

In each case the computer will ask for instructions for scaling the plots:

1) set horizontal scale? Y or N (= blank)

If Y (yes) is entered, the computer responds:

min/max X values

After entering the appropriate minimum and maximum X values the program moves to part 2 below.

If N (no) or blank (space) is entered, the program moves immediately to part 2.

2) Set vertical scale? Y or N (= blank)

If Y is entered, the computer responds:

MIN/MAX Y values

The appropriate values must be entered, and the program moves to part 3. If N or blank is entered, the program moves immediately to part 3.

3) Skip plot of EW tilt

Entering Y (yes) causes the plot to be skipped, entering N or blank causes the plot to appear on the CRT.

If no scaling is indicated (N or blank entered for parts 1 and 2), the upper and lower bounds on the horizontal and vertical axes will be the minimum and maximum values computed by the program. For plots other than the east-west component, part 3 changes appropriately.

After the plot is displayed, entering a blank (space) causes the computer to respond:

$\emptyset$ = Re-start, 1 = New plot

Entering zero causes the program to start over, a 1 initiates a new plot (and re-scaling if desired), and any other number allows the program to move to the next plot.

Once the plotting is finished, or if a 're-start' is chosen, the computer prints out the C array (i.e., the initial and final values of the zones' coordinates and the slip - App. A), and the various flags and time constants selected.

The format is as follows:

C(1) C(2) C(3) C(4)

C(5)

⋮

C(33) . . . . . C(36)

IFLAG IKFLAG LFLAG IJKL NUM NEND M1

M2 M3 KTIME K7 K8 TAU RTIME

The meaning of the flags are explained in Appendix C.

Then the computer responds:

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 vlaue only

6 = Re-start with new tiltmeter coordinates only

7 = Stop

Entering the number corresponding to the option desired allows the program to be restarted either with all new values (1), some subset of new values (2 through 6), or terminates the program (7). If options 1 through 6 are selected, the computer will ask for the information it needs as before.

XPNDØ2

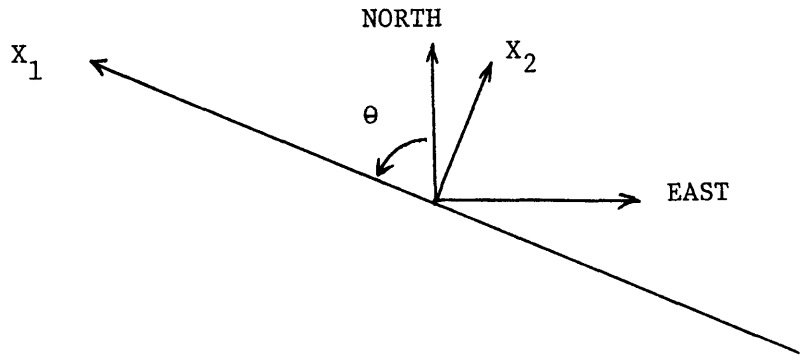
This program operates in the same fashion as, and requires responses identical to, XPNDØ1. However instead of the TRIGGER option responding 'D(M1) > C(M2)' it will ask for M1 and M2 such that 'D(M1) < C(M2)'. This modification allows the triggered zone to have coordinates that are less than the coordinates of the triggering zone such as in figure A1.

References

Press, Frank - Displacements, Strains, and Tilts at Teleseismic Distances - J. Geophy. Res., 70, 2395-2412, 1965.

FIGURES AND APPENDICES FOR

XPND PROGRAMS



θ = theta - assumed to be 45.°

Figure 1

Geometry assumed in Programs XPND/XPNDØ1

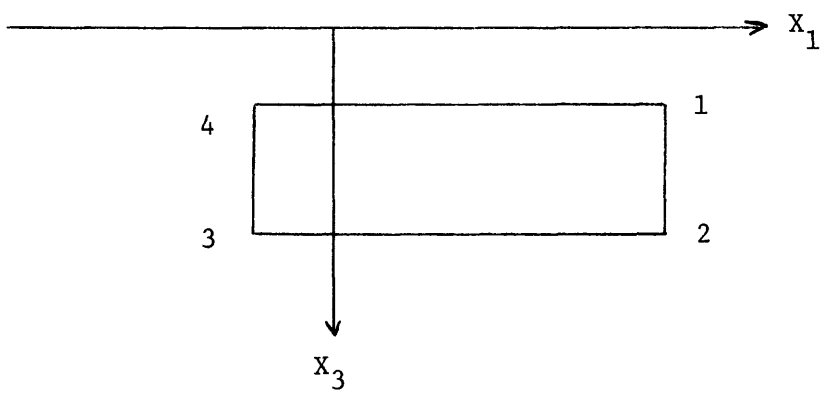


Figure 2

Geometry and notation used in Programs XPND/XPNDØ1

APPENDIX A

Array assignments:

C array

Dip-slip zone coordinates		strike-slip zone coordinates
C(1) = D1X1IN		C(17) = C1X1IN
C(2) = D1X3IN		C(18) = C1X3IN
C(3) = D2X1IN	initial	C(19) = C2X1IN
C(4) = D2X3IN	values	C(20) = C2X3IN
C(5) = D3X1IN		C(21) = C3X1IN
C(6) = D3X3IN		C(22) = C3X3IN
C(7) = D4X1IN		C(23) = C4X1IN
C(8) = D4X3IN		C(24) = C4X3IN
C(9) = D1X1FN		C(25) = C1X1FN
C(10)= D1X3FN		C(26) = C1X3FN
C(11)= D2X1FN	final	C(27) = C2X1FN
C(12)= D2X3FN	values	C(28) = C2X3FN
C(13)= D3X1FN		C(29) = C3X1FN
C(14)= D3X3FN		C(30) = C3X3FN
C(15)= D4X1FN		C(31) = C4X1FN
C(16)= D4X3FN		C(32) = C4X3FN
C(33) = U1IN		C(35) = U3IN
C(34) = U1FN		C(36) = U3FN

D Array

(working array)

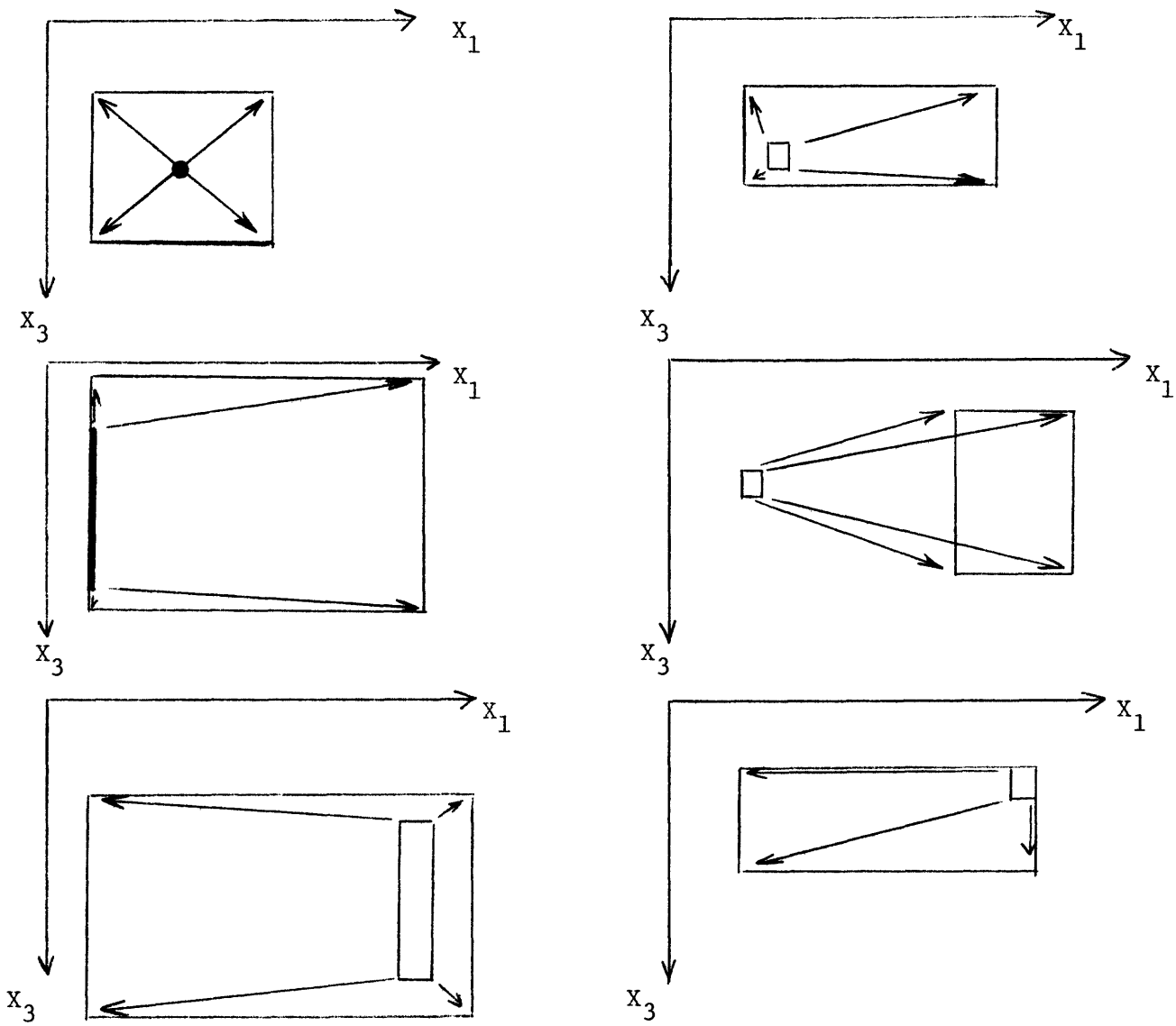
D(1) = D1X1	D(9) = C1X1
D(2) = D1X3	D(10)= C1X3
D(3) = D2X1	D(11)= C2X1
D(4) = D2X3	D(12)= C2X3
D(5) = D3X1	D(13)= C3X1
D(6) = D3X3	D(14)= C3X3
D(7) = D4X1	D(15)= C4X1
D(8) = D4X3	D(16)= C4X3

D(17) = U1 D(18) = U3

The coordinates are incremented linearly:

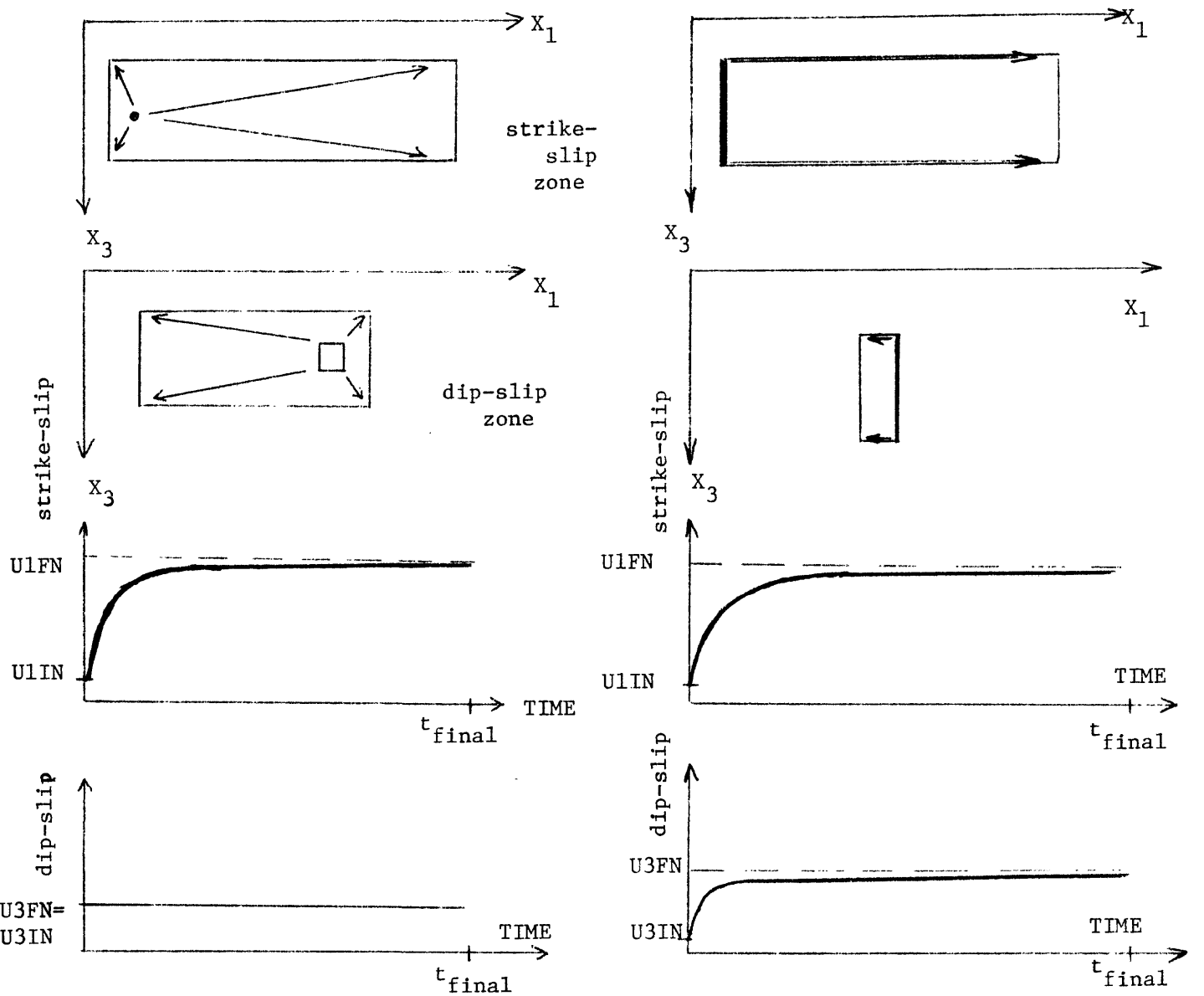
ie. (initial) + [(final-initial)*(time)]

Examples of slip zone expansion:



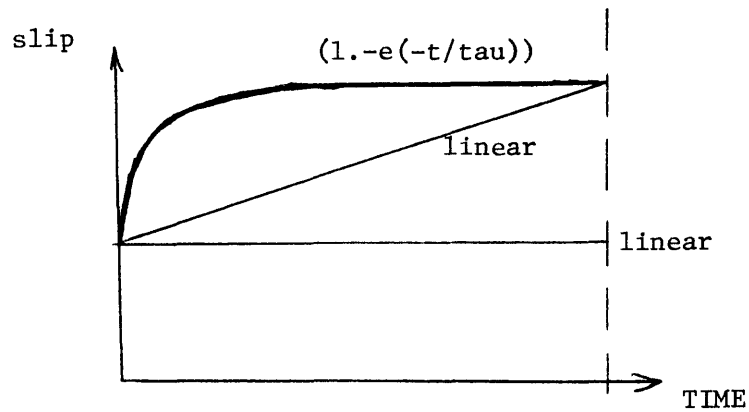
If the contraction option is entered the zone's growth will be the reverse of that shown above.

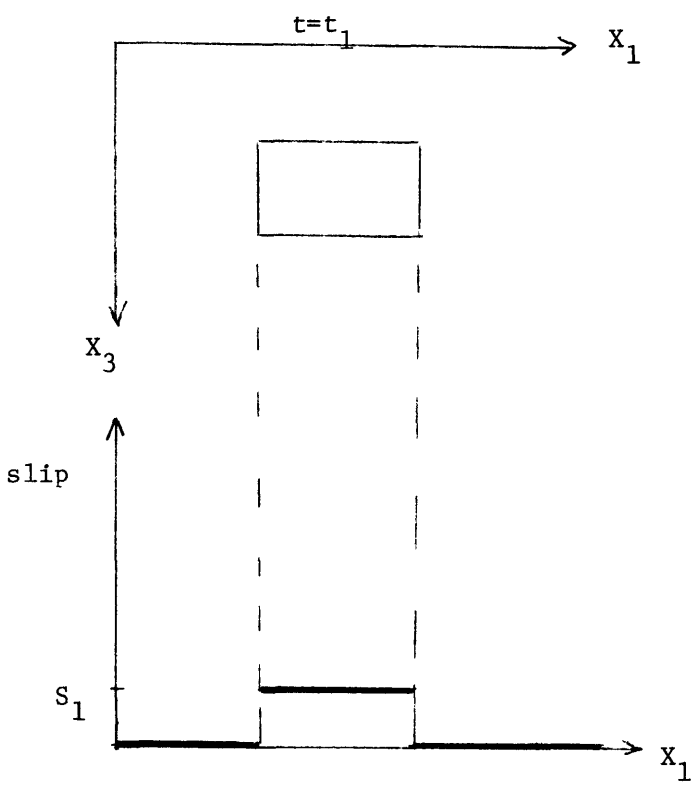
If both dip-slip and strike-slip are present, and if the two zones start growing at the same instant (if not, the 'TRIGGER' option must be used), then two of the various possibilities may be as shown below:

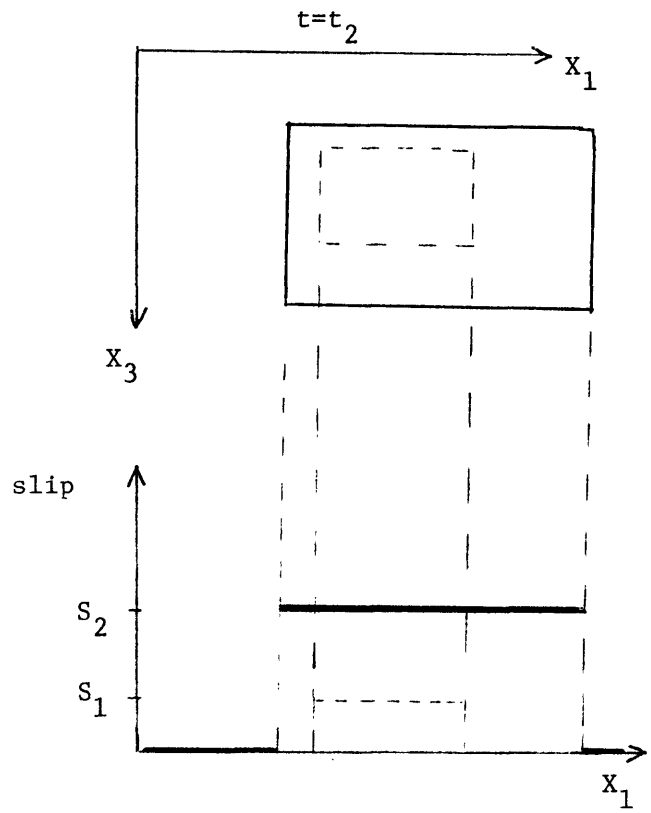


Note that the dip-slip and strike-slip zones need not be spatially coincident; it is only necessary that they start growing at the same instant.

The slip magnitude is constant over the zone at each instant, but may change in time as shown:





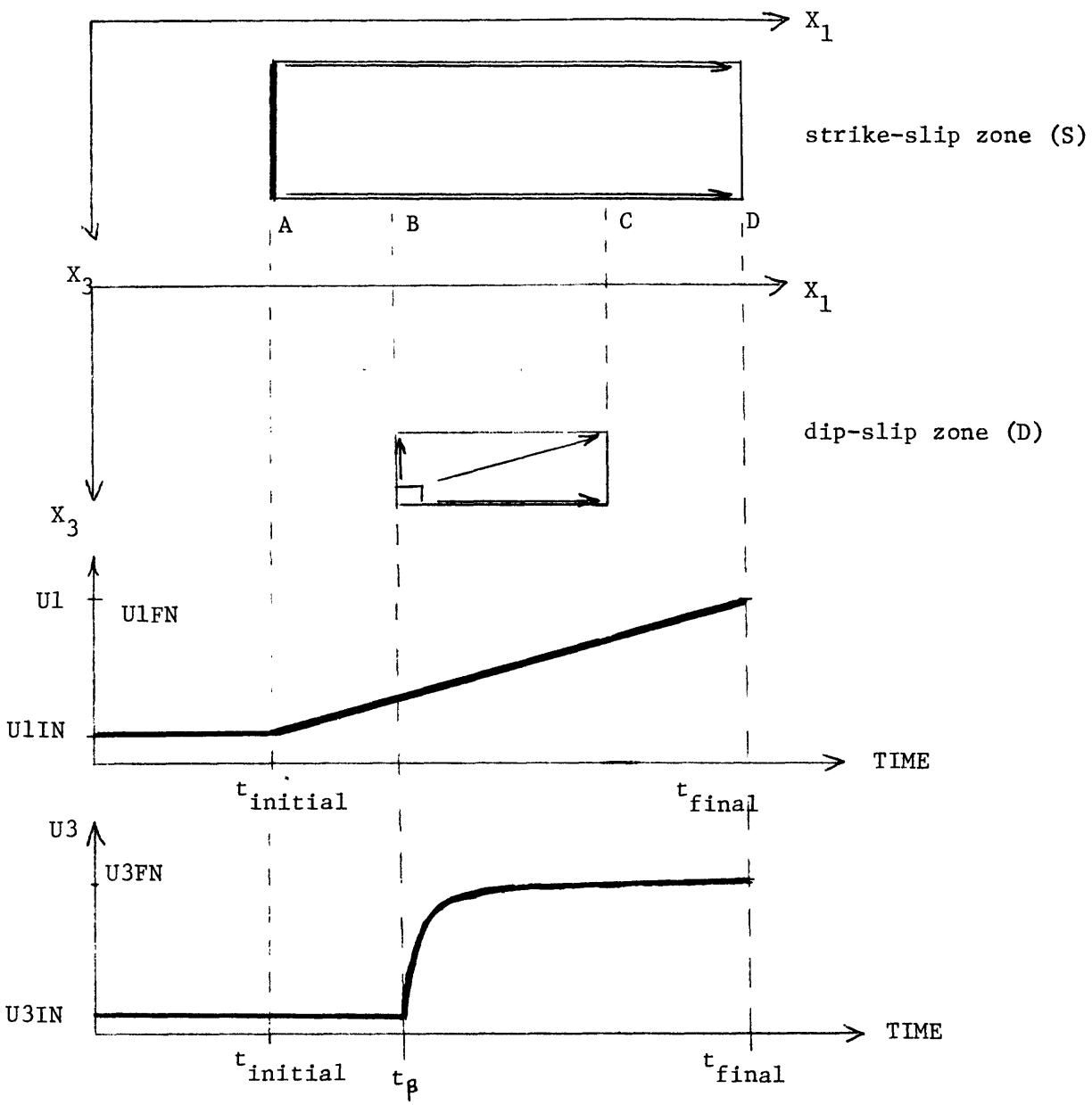


APPENDIX B

Trigger Option

To simulate a growing slip zone that has both dip-slip and strike-slip components, two sub zones are specified: one that is pure strike-slip and one that is pure dip-slip. If the two subzones are not temporally coincident, the growth of one must be tied to, or initiated by, the growth of the other. The 'TRIGGER' option allows one zone to trigger the expansion of the other.

For example:


X1
strike-slip zone (S)
A
B
C
D
X3
X1
dip-slip zone (D)
X3
U1
U1FN
U1IN
TIME
t initial
t final
U3
U3FN
U3IN
TIME
t initial
tβ
t final

In this example the strike-slip zone (S) expands such that corners 3 and 4 are stationary and 1 and 2 move to position D. When C1X1 or C2X1 is equal to or greater than D1X1 or D2X1 (as happens at time = t_β), slip starts on the dip-slip zone (D) and the dip-slip zone expands. The position of C1 (or C2) determines whether or not the dip-slip zone is growing. Therefore in this case, M1 = 9 and M2 = 1 because no slip on the D zone occurs until C1X1 $\geq$ D1X1IN (i.e. D(9) $\geq$ C(1)), and a 'D' is entered so that the dip-slip component is zero until D(9) $\geq$ C(1). Other cases can be generated using the options and geometries discussed previously.

APPENDIX C

Definitions of Some Parameters

IFLAG - this flag determines whether the zone expands or contracts

LFLAG - determines whether the slip increases linearly or exponentially

NUM = total number of points used in plotting

IJKL = index equivalent to initial position of zone

= (.075 * NUM) + 1.0

NEND = index equivalent to final position of zone

= NUM -IJKL

(NEND and IJKL are used for scaling the time axis).

TAU = time constant in exponential = -(NEND-IJKL)/6.

KTIME = index equivalent to the amount of time that triggered zone has in which to expand

= NEND $-I_{trigger}$

RTIME = index equivalent to time that zone is triggered

= $I_{trigger}$

K7 = NEND - 1)
) used for scaling the time axis
K8 = NEND + 1)

M1)
) defined by trigger option : D(M1) $\geq$ C(MZ)
M2)

M3 - defined by the requirement that D(M3) = 0.0 when D(M1) < C(M2).

Therefore M3 = 17 if 'S' is entered in the trigger option, and M3 = 18 if 'D' is entered.

IKFLAG: determines whether the program is restarted (Ø) after a plot, a new plot is displayed (1), or the program continues (any other number).

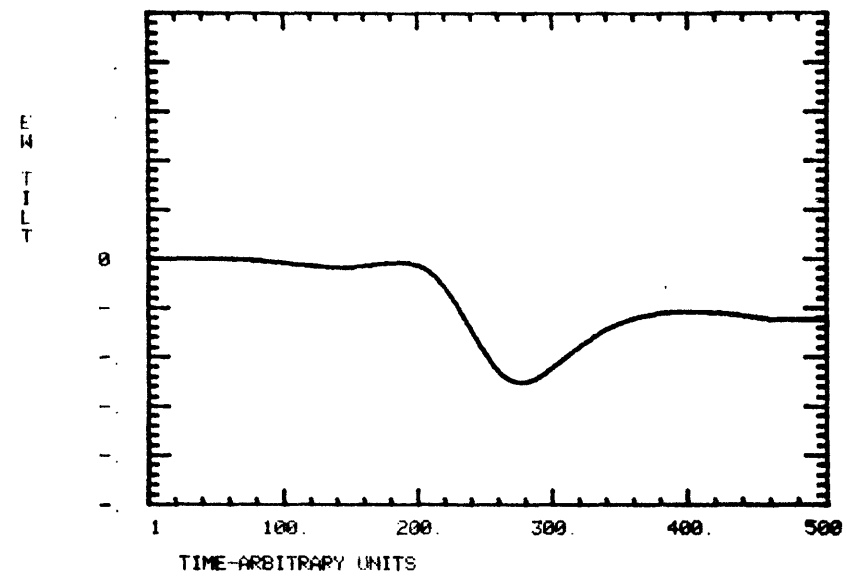
EXAMPLE FROM XPND

```
1=ZONE EXPANDS
2=ZONE CONTRACTS
1!
1=STRIKE SLIP COMPONENT INCREMENTED EXPONENTIALLY
2=DIP SLIP COMPONENT INCREMENTED EXPONENTIALLY
3=BOTH SLIP COMPONENTS INCREMENTED EXPONENTIALLY
4 NEITHER SLIP COMPONENT INCREMENTED EXPONENTIALLY
3
DO YOU WISH THE 'TRIGGER' OPTION
'YES' OR 'NO'
YES!
SPECIFY 'TRIGGER' PARAMETERS IN THE FORM
D(M1) ,E (M2)
4 ,1!
SPECIFY WHICH SLIP COMPONENT IS TO BE =0
WHEN D(M1) .LT. E(M2)
STRIKE-SLIP=S, DIP-SLIP=D
D!
SPECIFY INITIAL AND FINAL SLIP VALUES,
RIGHT-LATERAL STRIKE-SLIP IS <0
DIP-SLIP >0, FOR 'X2>0, SIDE' DOWN
FORMAT=U1IN,U1FN,U3IN,U2FN
0 -3 0 .5!
SPECIFY INITIAL AND FINAL COORDINATES
OF DIP-SLIP ZONE CORNERS AS INDICATED
D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,D3X1FN
 9 0 1 9 1.1 0 1 9!
SPECIFY INITIAL AND FINAL VALUES
OF STRIKE-SLIP ZONE COORDINATES AS INDICATED
C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN C1X3FN,C2X3FN,C3X1FN
0 0 0 0 2 0 1 0
```



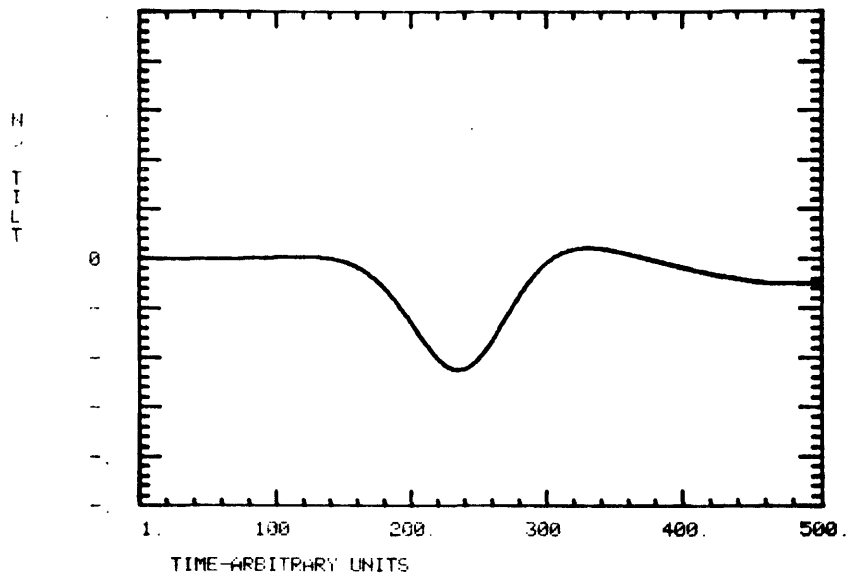
```
!
COORDINATES OF STATION (X1,X2)
1 3?!
THETA=ANGLE BETWEEN STRIKE OF FAULT
AND NORTH=    45.000 DEGREES
MIN./MAX. VALUES OF EW COMPONENT  -2.513E-01   0.
MIN./MAX. VALUES OF NS COMPONENT  -2.240E-01   2.220E-02
MIN./MAX. VALUES OF AMPLITUDE      0.          2.824E-01
MIN./MAX  VALUES OF AZIMUTH        0.        288.503
(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=KE-START, 1=CONTINUE
1!
WRITE PLOT TITLE, 80 CHARACTERS
EXPANDING SLIP ZONES!
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?
```

EXPANDING SLIP ZONES



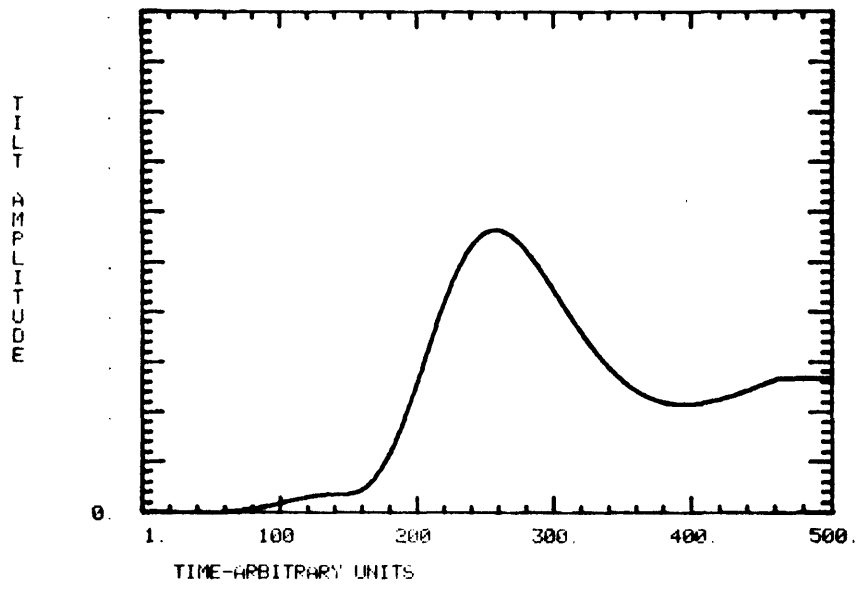

```
0=RE-START, 1=NEW PLOT
2
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y
SET VERTICAL SCALE? Y OR N(=BLANK)
Y
MIN/MAX Y VALUES
-.5   .5
SKIP PLOT OF NS TILT
```

EXPANDING SLIP ZONES

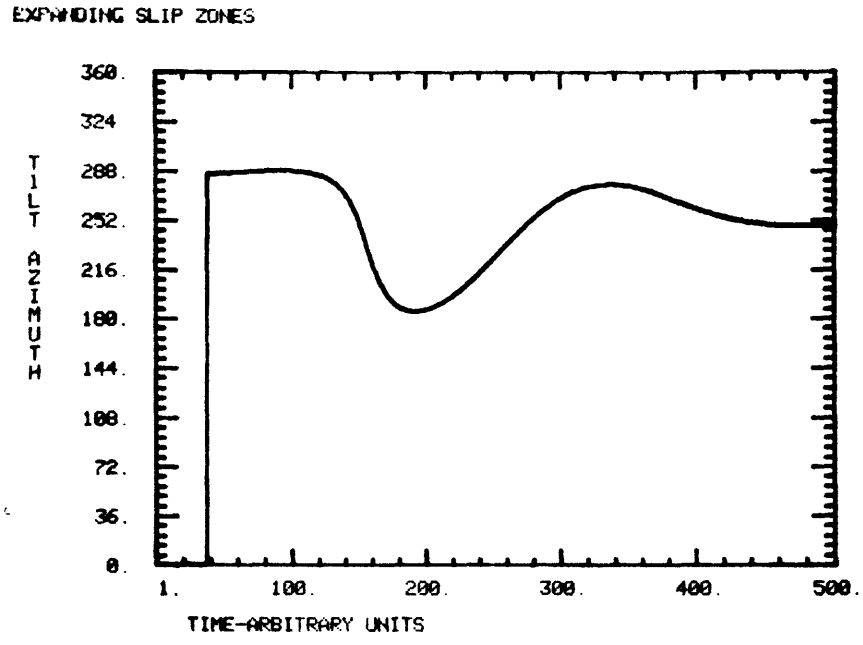


!
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN,MAX Y VALUES
0 .5!
SKIP PLOT OF TILT AMPLITUDE?

EXPANDING SLIP ZONES




```
0=RESTART, 1=NEW PLOT
2
SET HORIZONTAL SCALE? X (OR N=BLANK)
X
SET VERTICAL SCALE? Y (OR N=BLANK)
Y
MIN,MAX Y VALUES
0, 360
SLIP PLOT OF TILT AZIMUTH
```



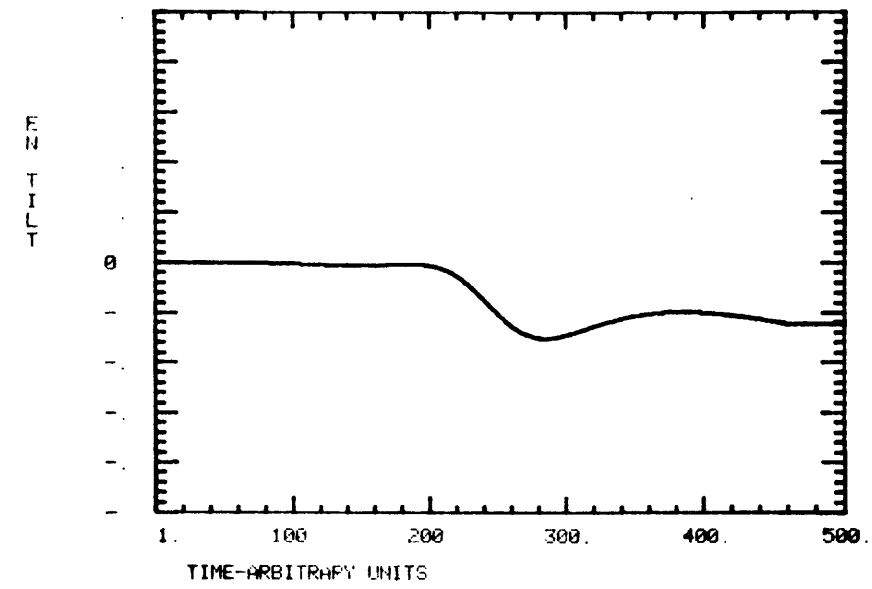

```
        .90          0.           .90          1.00
        .90          1.00         .90          0
       1.10          0           1.10          1.00
        .90          1.00         .90          0
          0          0              0          0.
          0.         0.             0          0.
       2.00          0.          2.00          1.00
          0.         1.00           0.         0
          0         -3.00           0.          .50

          1          2              3          38          500          462          9
          1         18            233         37          463       -70.667     229.000
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
OK - ^EDIT
```

EXAMPLE FROM XPND01

```
1=ZONE LENGTHWISE
2=ZONE POINTWISE
(
1=STRIKE-SLIP COMPONENT INCREMENTED EXPONENTIALLY
2=DIP-SLIP COMPONENT INCREMENTED EXPONENTIALLY
3=BOTH SLIP COMPONENTS INCREMENTED EXPONENTIALLY
4=NEITHER SLIP COMPONENT INCREMENTED EXPONENTIALLY
2!
DO YOU WISH THE 'TRIGGER' OPTION?
'YES' OR 'NO'
NO!
SPECIFY 'TRIGGER' PARAMETERS IN THE FORM
(M1, GE (M2)
-1!
SPECIFY WHICH SLIP COMPONENT IS TO BE =0.
WHEN (M1) LT (M2)
STRIKE-SLIP=S, DIP-SLIP=D
D!
SPECIFY INITIAL AND FINAL SLIP VALUES.
RIGHT-LATERAL STRIKE-SLIP IS <0
DIP-SLIP >0 FOR 'HEAD' SIDE DOWN
FORMAT=U1IN,U1FN,U3IN,U3FN
-3 0 5!
SPECIFY INITIAL AND FINAL COORDINATES
OF DIP-SLIP ZONE CORNERS AS INDICATED
D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,D3X1FN
-.9 1 .9 1.1 0 1 9!
SPECIFY INITIAL AND FINAL VALUES
OF STRIKE-SLIP ZONE COORDINATES AS INDICATED
C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN
0 3 0 0 2 0 1 0
```

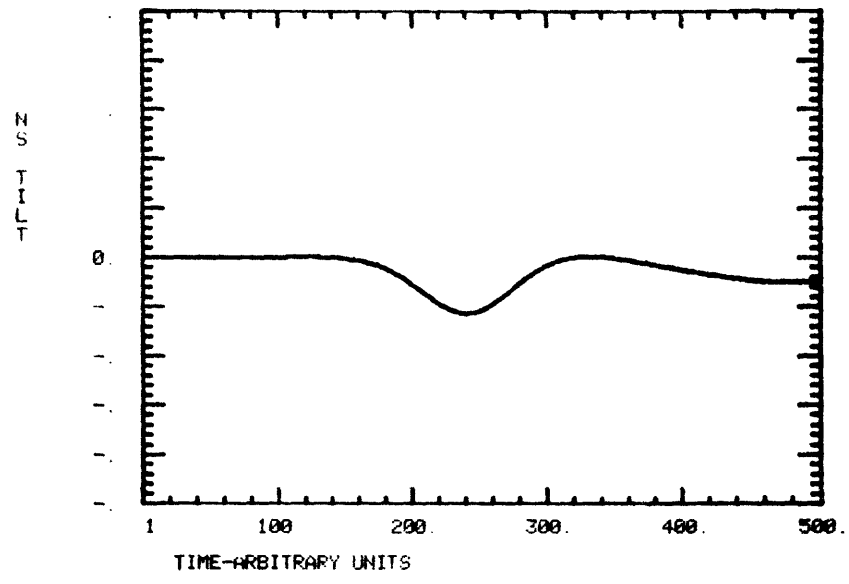


```
(
COORDINATES OF STATION (X1,X2)
1 37!
THETA=ANGLE BETWEEN STRIKE OF FAULT
AND NORTH= 45.000
SPECIFY 2 CORNERS OF DISLOCATION SURFACE FOR DISPLAY
1=D1X1=D2X1 2=D1X3=D4X3
4=D2X3=D3X3 5=D3X1=D4X1
9=C1X1=C2X1 10=C1X3=C4X3
11=C2X3=C3X3 13=C3X1=C4X1
' 10!
MIN/MAX VALUES OF EW COMPONENT -1.524E-01 0.
MIN/MAX VALUES OF NS COMPONENT -1.132E-01 2.471E-03
MIN/MAX VALUES OF AMPLITUDE 0 1.617E-01
MIN/MAX VALUES OF AZIMUTH 0. 298.503
(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
XPND01/EXPANDING SLIP ZONES!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
!
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN/MAX Y VALUES
-5\5
-5 5!
SKIP PLOT OF EW TILT?
```

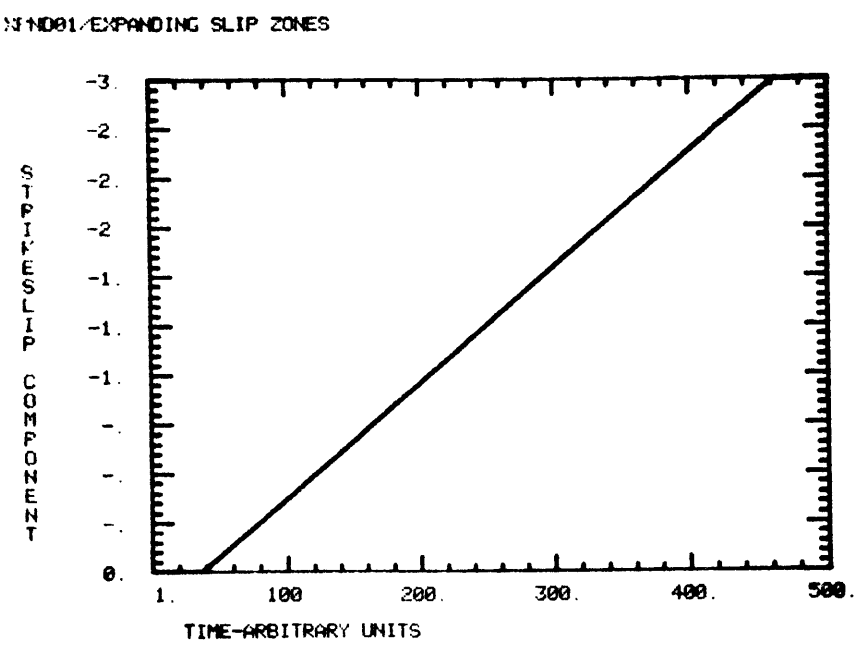


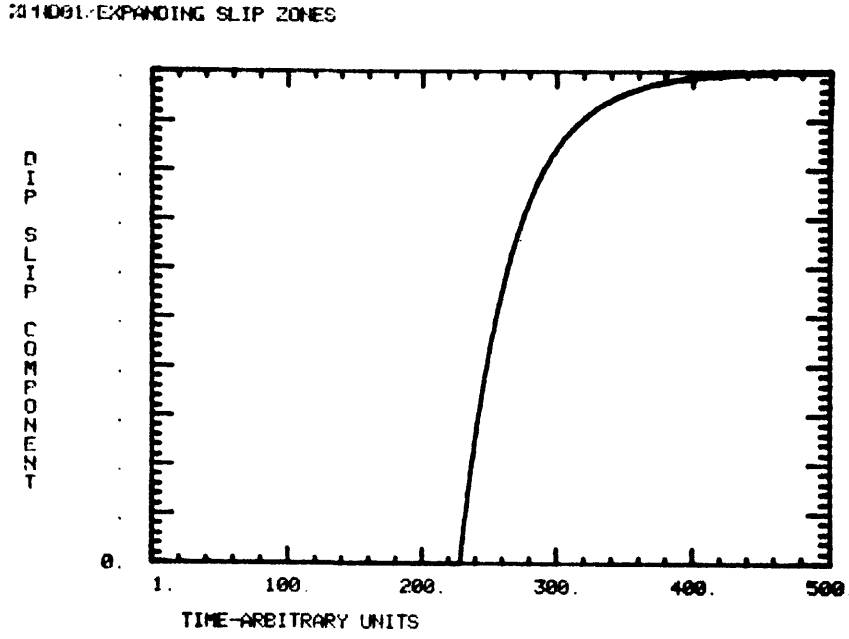

```
TO PRE-START, 1=NEW PLOT
?1
SET HORIZONTAL SCALE? Y OR N(=BLANK)
?
SET VERTICAL SCALE? Y OR N(=BLANK)
?Y
MIN,MAX Y VALUES
-.5  .5
SLIP PLOT OF NS TILT?
```

%T1D01/EXPANDING SLIP ZONES



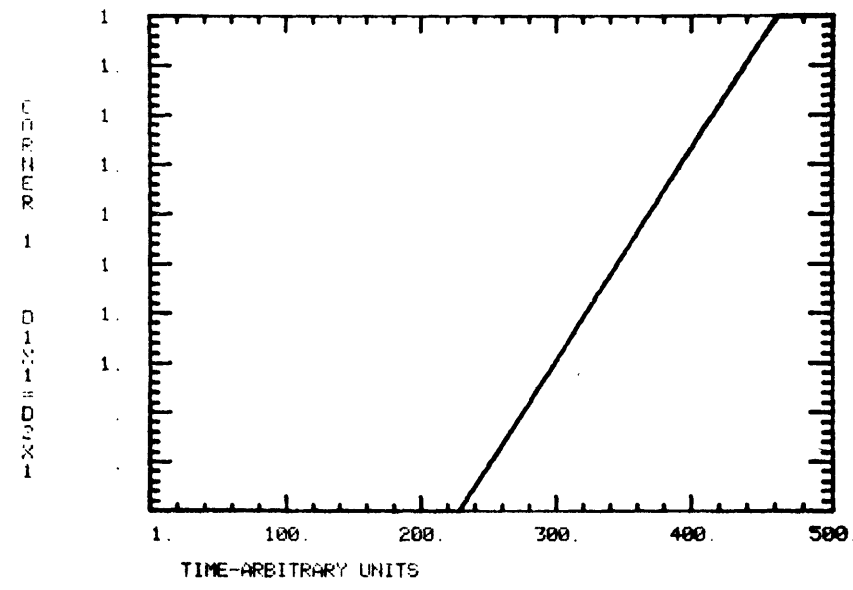
0=RE-START, 1=NEW PLOT
?1
SET HORIZONTAL SCALE? Y OR N=BLANK)
?
SET VERTICAL SCALE? Y OR N=BLANK)
?
SKIP PLOT OF TILT AMPLITUDE?
?1
0=RE-START, 1=NEW PLOT
?1
SET HORIZONTAL SCALE? Y OR N=BLANK)
?
SET VERTICAL SCALE? Y OR N=BLANK)
?
SKIP PLOT OF TILT AZIMUTH
?1
0=RE-START, 1=NEW PLOT
?1
SKIP PLOT OF STRIKE SLIP COMPONENT?

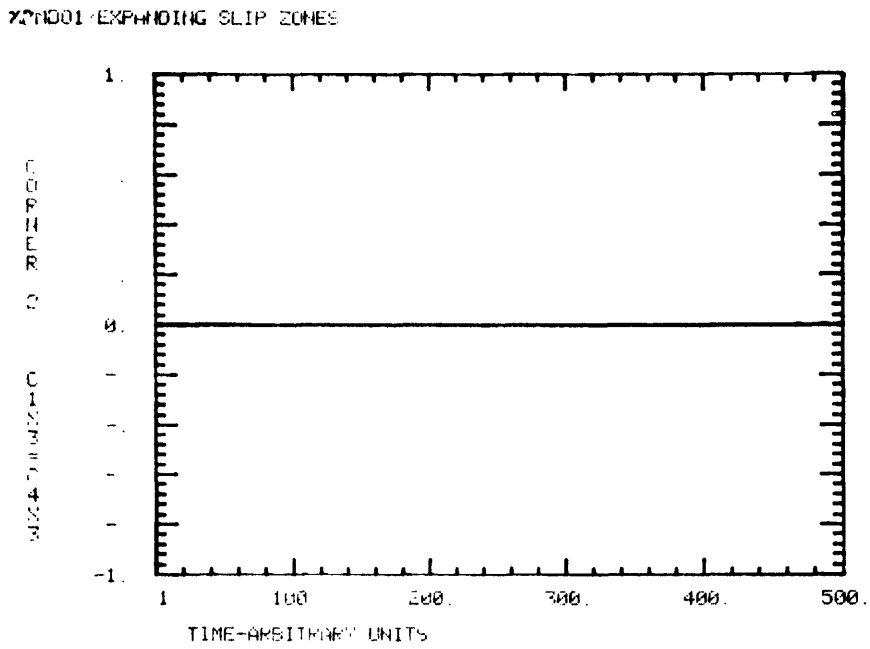

:1ND01/EXPANDING SLIP ZONES
STRIKE SLIP COMPONENT
-3.
-2.
-2.
-2
-1.
-1.
-1.
-.
-.
-.
0.
1.
100
200.
300.
400.
500.
TIME-ARBITRARY UNITS


:11D01/EXPANDING SLIP ZONES
DIP SLIP COMPONENT
0.
1.
100.
200.
300.
400.
500.
TIME-ARBITRARY UNITS

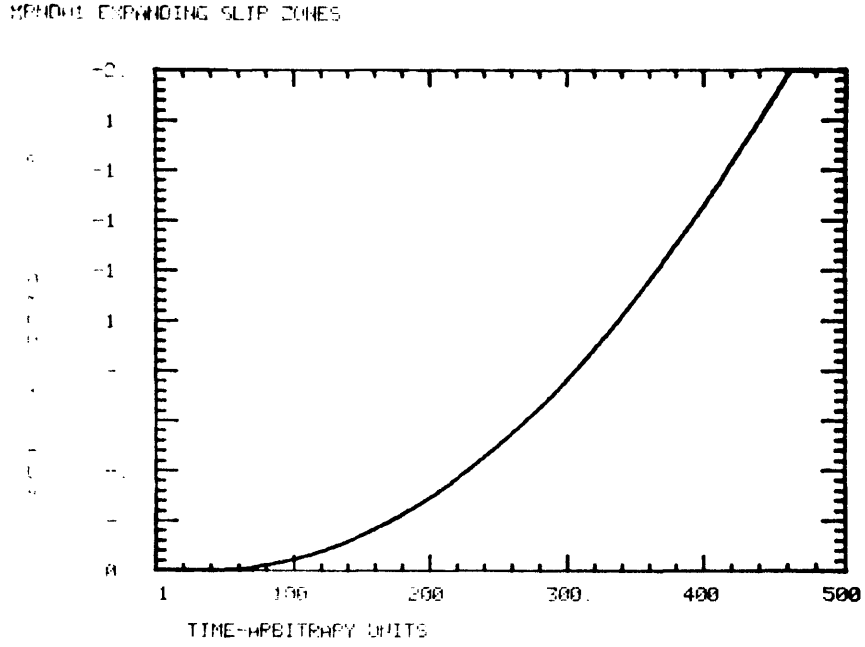
0 START
2
SKIP PLOT OF CORNER 1?

XPNDU1-EXPANDING SLIP ZONES




1.
0.
-1.
1
100
200.
300.
400.
500.
TIME-ARBITRARY UNITS

A40


1
-1
-1
-1
1
1
100
200
300.
400
500.
TIME-ARBITRARY UNITS

A41

```
          .90          0.            .90          1.00
          .90          1.00          .90          0.
         1.10          0.           1.10          1.00
          .90          1.00          .90          0.
         0.            0.           0.            0.
         0.            0.           0.            0.
         2.00          0            2.00          1.00
         0.            1.00         0.            0.
         0.           -3.00         0.             .50

          1          2          2         38        500        462          9
          1         18        233         37        463    -70.667    229.000
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
OK -- ^EDIT
```

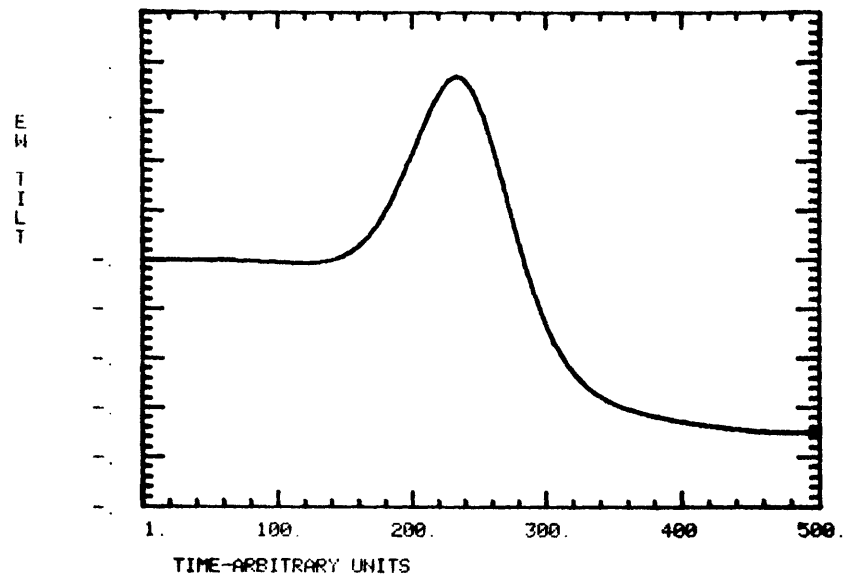
EXAMPLE FROM XPNDO2

```
1=ZONE E
2=ZONE
!!
1=S
2=D
3=B
4=N
3
DO YOU WISH THE "STRIKE" OPTION
"1" OR "NO"
1
IF "TRIGGER" PARAMETERS IN THE FORM
(M1 LT COMZ
1
SPECIFY WHICH SLIP COMPONENT IS TO BE =0.
WHEN D(M1 OF C(M2
ENTER SLIP=
2!
SPECIFY INITIAL AND FINAL SLIP VALUES
RIGHT-LATERAL STRIKE-SLIP IS >0
DIP-SLIP >0  FOR "UP"-0  "SIDE" DOWN
FORMAT=U1IN,U1FN,U3IN,U3FN
0   3  0  1!
SPECIFY INITIAL AND FINAL COORDINATES
OF DIP-SLIP ZONE CORNERS AS INDICATED
D1.C1IN,D1X3IN,D2X3IN,D2X1IN,D1X1FN D1X3FN,D2X3FN,D3X1FN
1 1  0  1  1 1  1.1  0  1   9!
SPECIFY INITIAL AND FINAL VALUES
OF STRIKE-SLIP ZONE COORDINATES AS INDICATED
C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN
2 0 0 2 2 0 1 0
```



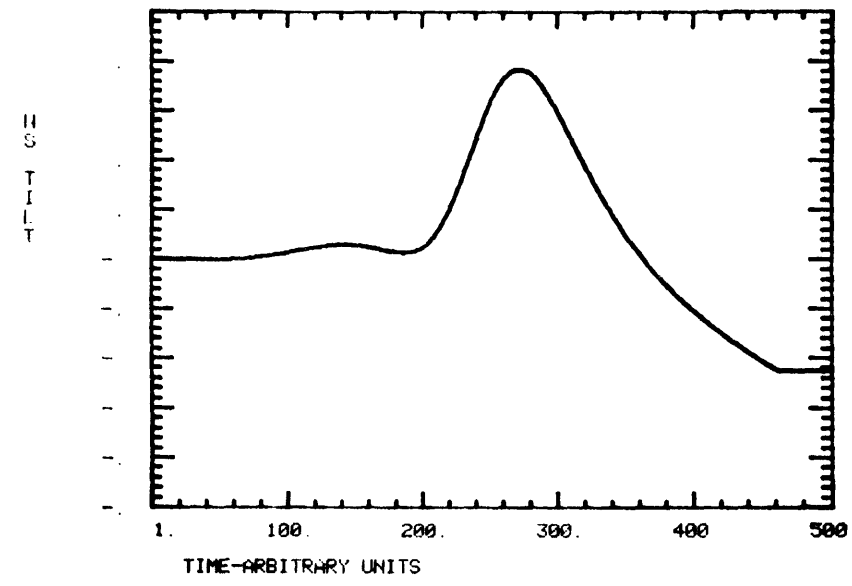
```
COORDINATE OF STATION  1  =
1   .37
THETA=ANGLE BETWEEN STRIKE OF FAULT
AND NORTH:    4
SPEC    1    CORNERS OF RECTANGULAR ZONE FOR DISPLAY
1=D1  1=D1X1          2=D1  X4=D4X3
4=D2  3=D2X3          5=D2X1=D3X1
9=C1X1=0,2=1          10=11,12=,4X2
12=C...               1=%,2,3=4,
5   13!
MIN  MAX  VALUES OF EW COMPONENT   -2.101E-01    2.226E-01
MIN  MAX  VALUES OF NS COMPONENT   -1.358E-01    2.309E-01
MIN  MAX  VALUES OF AMPLITUDE       0.           2.663E-01
MIN  MAX  VALUES OF AZIMUTH         0.         359.379
 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
'PND02-EXPANDING  SLIP  ZONES!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
 !
SET VERTICAL SCALE? Y OR N(=BLANK)
Y!
MIN-MAX Y VALUES
-.3   .3!
SKIP PLOT OF EW TILT?
```

XP:DD2/E:EXPANDING SLIP ZONES

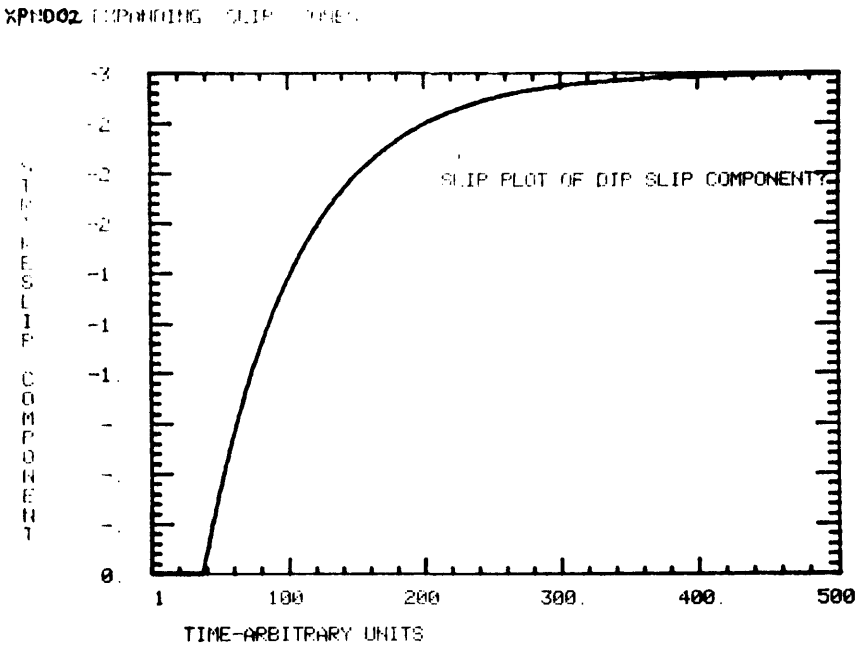


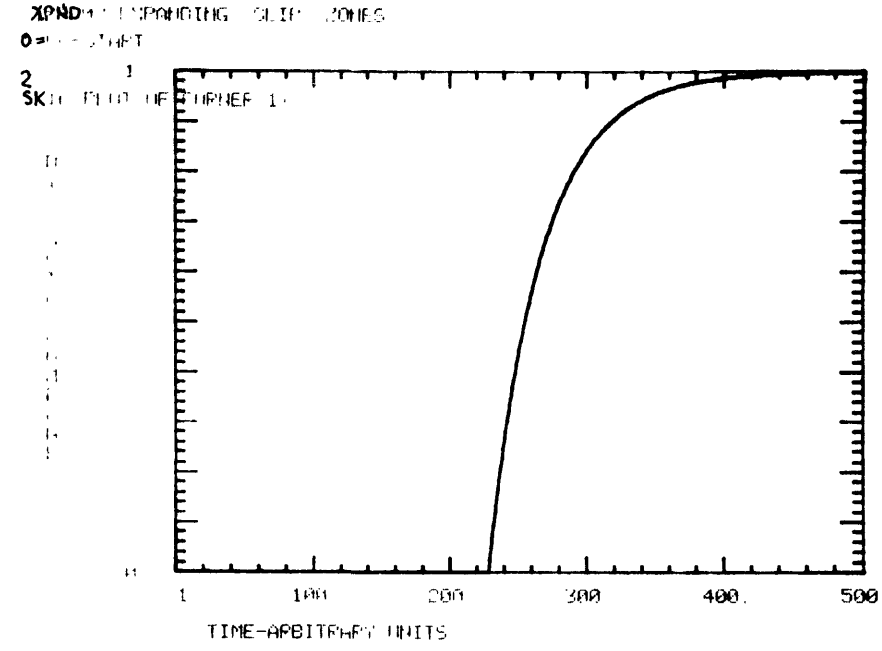
!
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
!
SET VERTICAL SCALE? Y OR N(=BLANK)
?!
MIN/MAX Y VALUES
-3 3!
SKIP PLOT OF NS TILT?

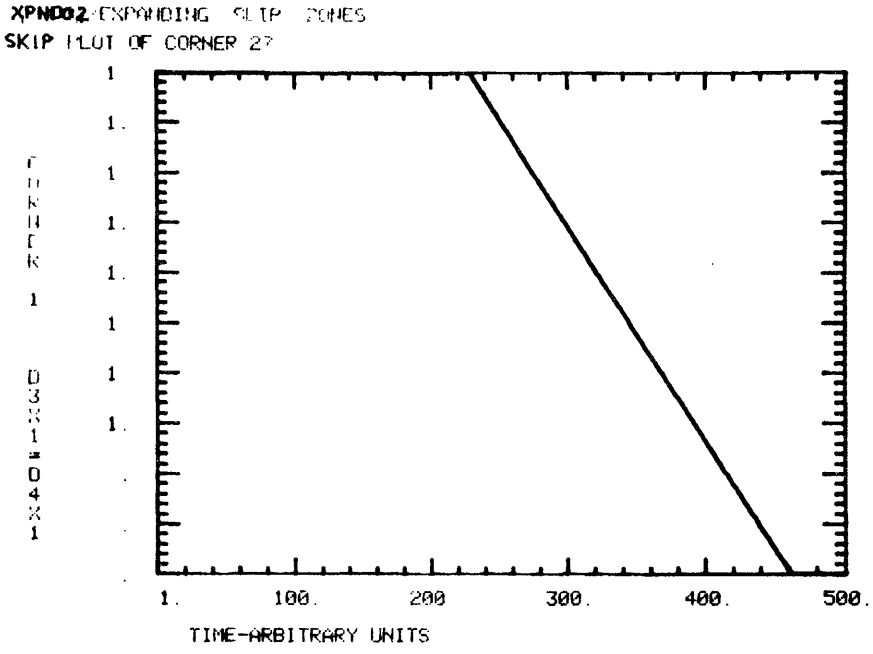
XPMD32/EXPANDING SLIP ZONES



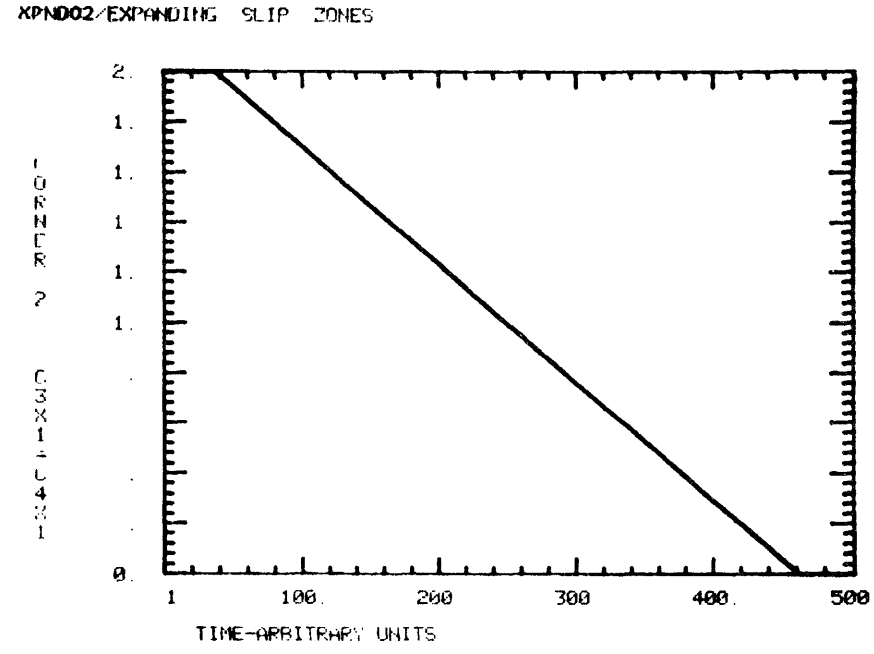
0=RE-START, 1=NEW PLOT
2!
SET HORIZONTAL SCALE? Y OR N(=BLANK)
SET VERTICAL SCALE? Y OR N(=BLANK)
SKIP PLOT OF TILT AMPLITUDE?
0=RE-START, 1=NEW PLOT
SET HORIZONTAL SCALE? Y OR N(=BLANK)
SET VERTICAL SCALE? Y OR N(=BLANK)
SKIP PLOT OF TILT AZIMUTH?
0=RE-START, 1=NEW PLOT
SKIP PLOT OF STRIKE SLIP COMPONENT?


SLIP PLOT OF DIP SLIP COMPONENT
TIME-ARBITRARY UNITS
1
100
200
300.
400.
500.
0.


TIME-ARBITRARY UNITS
1
100
200
300
400.
500


XPND02/EXPANDING SLIP ZONES
SKIP PLOT OF CORNER 2?
1
1.
1
1.
1
1.
1
1
1.
1.
100.
200
300.
400.
500.
TIME-ARBITRARY UNITS

A51


XPND02/EXPANDING SLIP ZONES
2.
1.
1.
1
1.
1.
0.
1
100.
200
300
400.
500.
TIME-ARBITRARY UNITS

A52

```
 !
SKIP PLOT OF STRIKE SLIP AREA?
Y!
SKIP PLOT OF DIP SLIP AREA?
Y!
        1.10         0.          1.10         1.00
        1.10         1.00        1.10         0.
        1.10         0.          1.10         1.00
         .90         1.00         .90         0.
        2.00         0.          2.00         0.
        2.00         0.          2.00         0.
        2.00         0.          2.00         1.00
        0.           1.00        0.           0.
        0.          -3.00        0.           1.00

          1          2          3          38        500        462          13
          1         18        233          37        463   -70.667    229.000
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
```

LISTING OF XPND

```
24 MAY 76    10.52.59          MCHUGH .XPND

  1  DELETE(LGO,OUTPUT,XPND)
  2  XPND.
  3  CXIT.
  4  LIBCOPY(GRAPHIC,TXLGO/RR,TXLGO)
  5  LIBCOPY(JDRAT,NPLGO/RR,NPLGO)
  6  DELETE(LGO,OUTPUT,XPND)
  7  RUN76(S)
  8  LINK(F=LGO,F=TXLGO,F=NPLGO,B=XPND)
  9  XPND.
 10  FIN.
 11  EOR
 12        PROGRAM XPND(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
 13        COMMON/TVPCOL/TVPOL(8)
 14        COMMON/TVTONE/ITONE(30)
 15        COMMON/JPLOT/XLT,XRT,YLO,YLP,MAJX,MAJY,KX(2),KY(2),
 16       1LTITL(8),LU,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
 17        DIMENSION IFET(8)
 18        DIMENSION TEW(500),TNS(500),TAMP(500),TAZM(500),T(500)
 19        DIMENSION D(20),C(40),A(30)
 20        DATA KTIME,RTIME,KCHECK,M1,M2,M3,AAA/1.,1.,1.,1.,1.,1.,-1.234E-30/
 21        CALL FET(5LTAPE7,IFET,8)
 22        IFET(2)=IFET(2).OR.0000 0010 0000 0000 0000B
 23        IFET(8)=IFET(8).OR.4000 0000 0000 0000 0000B
 24        CALL FET(5LTAPE7,IFET,-8)
 25     20 CONTINUE
 26        MSTART=9 $WRITE(7,12)
 27     12 FORMAT(*1=ZONE EXPANDS*,/,
 28       1 *2= ZONE CONTRACTS*,
 29        CALL GETNUM (A) $IFLAG=A(1) $WRITE(7,15)
 30     15 FORMAT(*1=STRIKE-SLIP COMPONENT INCREMENTED EXPONENTIALLY*,/,
 31       1 *2=DIP-SLIP COMPONENT INCREMENTED EXPONENTIALLY*,/,
 32       1 *3=BOTH SLIP COMPONENTS INCREMENTED EXPONENTIALLY*,/,
 33       1 *4=NEITHER SLIP COMPONENT INCREMENTED EXPONENTIALLY*)
 34        CALL GETNUM (A) $IFLAG=A(1)
 35        CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK) $CALL WRITE1
 36        CALL GETNUM (A) $C(33)=A(1) $C(34)=A(2) $C(35)=A(3) $C(36)=A(4)
 37        DO 13 I=1,32
 38     13 C(I)=0.
 39    122 IF(C(35) .EQ. 0. .AND. C(36) .EQ. 0.) GO TO 100
 40        CALL WRITE2(C)
 41    100 IF(MSTART .EQ. 3) GO TO 123
 42        IF(C(33) .EQ. 0. .AND. C(34) .EQ. 0.) GO TO 101
 43        CALL WRITE3(C)
 44    101 IF(MSTART .EQ. 2) GO TO 123
 45        WRITE(7,4)
 46     4  FORMAT(*COORDINATES OF STATION (X1,X2)*)
 47        CALL GETNUM (A) $X1=A(1) $X2=A(2)  $THETA=45.
 48    123 WRITE(7,7)THETA
 49      7 FORMAT(*THETA=ANGLE BETWEEN STRIKE OF FAULT*,/,
 50       1 *AND NORTH=*,F10.3,* DEGREES*)
 51        X3=0.  $NUM=500  $IJKL=(.875*NUM)+1.
 52        IF(C(35).EQ.0..AND.C(36).EQ.0.)GO TO 151
 53        IF(C(33).EQ.0..AND.C(34).EQ.0.)GO TO 152
 54        GO TO 153
 55    151 DO 154 I=1,16
 56    154 C(I)=0.
 57        GO TO 153
 58    152 DO 155 I=16,32
```

```
 59     155 C(I)=0.
 60     153 CONTINUE
 61         NEND=NUM-IJKL  $TAU=-(NEND-IJKL)/6.
 62         DO 103 I=IJKL,NEND
 63         DO 104 K=1,16
 64         IF(K.LE.8)L1=K    $IF(K.LE.8)L2=K+8
 65         IF(K.GT.8.AND.K.LE.16)L1=K+8
 66         IF(K.GT.8.AND.K.LE.16)L2=K+16
 67     104 D(K)=(((C(L2)-C(L1))/(NEND-IJKL))*(I-IJKL))+C(L1)
 68         IF(LFLAG.EQ.1.OR.LFLAG.EQ.3)D(17)=((1.-EXP((I-IJKL)/TAU))*(C(34)-C
 69        1(33)))+C(33)
 70         IF(LFLAG.EQ.2.OR.LFLAG.EQ.3)D(18)=((1.-EXP((I-IJKL)/TAU))*(C(36)-C
 71        1(35)))+C(35)
 72         IF(LFLAG.EQ.1)D(18)=(((C(36)-C(35))/(NEND-IJKL))*(I-IJKL))+C(35)
 73         IF(LFLAG.EQ.2)D(17)=(((C(34)-C(33))/(NEND-IJKL))*(I-IJKL))+C(33)
 74         IF(LFLAG.EQ.4)D(17)=(((C(34)-C(33))/(NEND-IJKL))*(I-IJKL))+C(33)
 75         IF(LFLAG.EQ.4)D(18)=(((C(36)-C(35))/(NEND-IJKL))*(I-IJKL))+C(35)
 76         IF(TRIGGR.NE.3HYES)GO TO 156
 77         IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2))GO TO 114
 78         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2))KCHECK=KCHECK+1
 79         IF(KCHECK.EQ.1)KTIME=NEND-I
 80         IF(KCHECK.EQ.1)RTIME=I  $IF(M3.EQ.18)GOTO 180
 81         DO 181 K=9,16
 82         L1=K+8   $L2=K+16
 83     181 D(K)=(((C(L2)-C(L1))/(KTIME))*(I-RTIME))+C(L1)
 84         GO TO 114
 85     180 DO 182 K=1,8
 86         L1=K    $L2=K+8
 87     182 D(K)=(((C(L2)-C(L1))/(KTIME))*(I-RTIME))+C(L1)
 88     114 IF(TRIGGR.EQ. 3HYES.AND.D(M1).LT.C(M2))D(M3)=0.
 89         IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.M3.EQ.17)GO TO 9
 90         IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.M3.EQ.18)GO TO 10
 91         GO TO 11
 92       9 DO 16 M=9,16
 93      16 D(M)=C(M+8) $GOTO11
 94      10 DO 18 M=1,8
 95      18 D(M)=C(M)
 96      11 IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.C(M3).EQ.0.)GO TO 156
 97         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2) .AND.LFLAG.EQ.1
 98        1 .AND.M3.EQ.17)D(M3)=((1.-EXP((I-RTIME)/
 99        1 (-KTIME/6.)))*(C(34)-C(33)))+C(33)
100         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.3
101        1 .AND. M3.EQ.17)D(M3)=((1.-EXP((I-RTIME)/(-KTIME/6.)))*(C(34)-C(33
102        1)))+C(33)
103         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.2
104        1 .AND.M3.EQ.18)D(M3)=((1.-EXP((I-RTIME)/
105        1 (-KTIME/6.)))*(C(36)-C(35)))+C(35)
106         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.3.AND.
107        1 M3.EQ.18)D(M3)=((1.-EXP((I-RTIME)/(-KTIME/6.)))*(C(36)-C(35)))+C(
108        135)
109         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.1
110        1 .AND.M3.EQ.18)D(M3)=(((C(36)-C(35))/
111        1 (KTIME))*(I-RTIME))+C(35)
112         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.4.AND.M3.EQ.18)
113        1 D(M3)=(((C(36)-C(35))/(KTIME))*(I-RTIME))+C(35)
114         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.2
115        1 .AND.M3.EQ.17) D(M3)=(((C(34)-C(33))/
116        1 (KTIME))*(I-RTIME))+C(33)
```

```
      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.M3.EQ.17.AND.LFLAG.EQ.4)
     1 D(M3)=(((C(34)-C(33))/(KTIME))*(I-RTIME))+C(33)
  156 CALL CMPTLT(D,X1,X2,X3,T1,T2)
      IF(D(1).NE.D(3).OR.D(5).NE.D(7).OR.D(2).NE.D(8)
     1 .OR.D(4).NE.D(6).OR.D(9).NE.D(11).OR.D(13).NE.D(15)
     1 .OR.D(10).NE.D(16).OR.D(12).NE.D(14))WRITE(7,67)
   67 FORMAT(*CORNERS NOT INCREMENTING PROPERLY*)
      T(I)=I $TEW(I)=T2
  103 TNS(I)=T1
      K7=IJKL-1
      DO 106 I=1,K7
      TEW(I)=TEW(IJKL) $T(I)=I
  106 TNS(I)=TNS(IJKL)
      K8=NEND+1
      DO 107 I=K8,NUM
      T(I)=I $TEW(I)=TEW(NEND)
  107 TNS(I)=TNS(NEND)
      IF(IFLAG.LE. 2)GO TO 108
      A=TEW(1)    $B=TNS(1)
      DO 109 I=1,NUM
      TEW(I)=TEW(I)-A
  109 TNS(I)=TNS(I)-B
      GO TO 110
  108 DO 111 I=1,NUM
      TEW(I)=TEW(NUM)-TEW(I)
  111 TNS(I)=TNS(NUM)-TNS(I)
  110 CT=COS(THETA*.01745)
      ST=SIN(THETA *.01745)
      DO 21 I=1,NUM
      B=TEW(I)  $A=TNS(I)
      TNS(I)=A*CT+B*ST
      TEW(I)=-A*ST+B*CT
      TAMP(I)=SQRT((TEW(I)**2)+(TNS(I)**2))
      IF(TNS(I).EQ.0.)TNS(I)=1.E-20
      TAZM(I)=(ATAN(TEW(I)/TNS(I)))*(180./3.1415926)
      IF(TNS(I).LT.0.)TAZM(I)=TAZM(I)+180.
      IF(TAZM(I).LT.0.)TAZM(I)=TAZM(I)+360.
   21 IF(TAZM(I).GT.360.)TAZM(I)=TAZM(I)-360.
      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 160 I=1,NUM
      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)
  160 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 *0=RE-START, 1=CONTINUE*)
      CALL GETNUM(A) $IKFLAG=A(1)
      IF(IKFLAG.EQ. 0) GO TO 113
      LU=7  $LNLGX=1  $LNLGY=1 $NCLX=2  $NCLY=2
      WRITE(7,199)
  199 FORMAT(*WRITE PLOT TITLE, 80 CHARACTERS*)
      READ(7,200)(LTITL(JM),JM=1,8)
  200 FORMAT(8A10)
      KX(1)=10HTIME-ARBIT
      KX(2)=10HRARY UNITS
      KY(1)=10H EW TILT
      KY(2)=10H            $MAJX=5 $MAJY=10 $LTITL2(1)=1
  903 XLT=T(1)$XRT=T(NUM)  $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(A)  $XLT=A(1)   $XRT=A(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(A)  $YLO=A(1)   $YUP=A(2)
   74 IF(TEWMIN.EQ.TEWMAX)YLO=YLO-1.
      IF(TEWMIN.EQ.TEWMAX)YUP=YUP+1.
      WRITE(7,1)
    1 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,NUM)
      WRITE(7,19) $CALL GETNUM(A)$IKFLAG=A(1)
      IF(IKFLAG.EQ.0)GOTO113$IF(IKFLAG.EQ.1)GOTO903
  900 KY(1)=10H NS TILT     $XLT=T(1)$XRT=T(NUM)
      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(A)$XLT=A(1)$XRT=A(2)
  710 WRITE(7,73)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO740
      WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(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,NUM)
      WRITE(7,19)
   19 FORMAT(*0=RE-START, 1=NEW PLOT*)
      CALL GETNUM(A) $IKFLAG=A(1)
      IF(IKFLAG.EQ.0)GO TO 113 $IF(IKFLAG.EQ.1)GOTO900
  911 KY(1)=10HTILT AMPLI  $XLT=T(1) $XRT=T(NUM)
      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(A) $XLT=A(1)$XRT=A(2)
  741 WRITE(7,73)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO742
      WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(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) $READ(7,66)IJVAR
  912 FORMAT(*SKIP PLOT OF TILT AMPLITUDE?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAMP,T,1,NUM)
      WRITE(7,19)$CALL GETNUM(A)$IKFLAG=A(1)
      IF(IKFLAG.EQ.0)GOTO113$IF(IKFLAG.EQ.1)GOTO911
  913 KY(1)=10HTILT AZIMU  $XLT=T(1)$XRT=T(NUM)
      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(A)$XLT=A(1)$XRT=A(2)
  743 WRITE(7,73)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO744
      WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(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,NUM)
      WRITE(7,68)(C(I),I=1,36)
   68 FORMAT(9(4F12.2,/))
      WRITE(7,69)IFLAG,IKFLAG,LFLAG,IJKL,NUM,NENE,M1,M2,M3,KTIME,K7,K8,
     1 TAU,RTIME
   69 FORMAT(7I10,/,5I10,2F10.3)
  113 WRITE(7,115)
  115 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*)
      CALL GETNUM(A) $MSTART=A(1)
      IF(MSTART.EQ.2)GOTO116 $IF(MSTART.EQ.3)GOTO117
      IF(MSTART.EQ.4)GOTO118 $IF(MSTART.EQ.5)GOTO119
      IF(MSTART.EQ.6)GOTO120 $IF(MSTART.EQ.7)GOTO630
      GO TO 20
  116 WRITE (7,12)
      CALL GETNUM(A) $IFLAG=A(1)  $WRITE(7,15)
      CALL GETNUM(A) $LFLAG=A(1)
      WRITE(7,23)
   23 FORMAT(*U1IN,U1FN*)
      CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
      CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
      GO TO 100
  117 WRITE(7,12)
      CALL GETNUM(A) $IFLAG=A(1)
      WRITE(7,15)
      CALL GETNUM(A) $LFLAG=A(1)
      WRITE(7,25)
   25 FORMAT(*U3IN,U3FN*)
      CALL GETNUM(A) $C(35)=A(1) $C(36)=A(2)
```

```
291         CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
292         GO TO 122
293     118 WRITE(7,12)
294         CALL GETNUM(A) $IFLAG=A(1)
295         WRITE(7,15)
296         CALL GETNUM(A) $LFLAG=A(1)
297         WRITE(7,23)
298         CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
299         CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
300         GO TO 123
301     119 WRITE(7,12)
302         CALL GETNUM(A) $IFLAG=A(1)
303         WRITE(7,15)
304         CALL GETNUM(A) $LFLAG=A(1)
305         WRITE(7,25)
306         CALL GETNUM(A) $C(35)=A(1) $C(36)=A(2)
307         CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
308         GO TO 123
309     120 WRITE(7,4)
310         CALL GETNUM(A) $X1=A(1) $X2=A(2)
311         CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
312         GO TO 123
313     630 STOP
314         END
315         SUBROUTINE CMPTLT(D,X1,X2,X3,T1,T2)
316         DIMENSION D(20)
317         A1=A2=A3=A4=B1=B2=B3=B4=0.
318         DA1=DA2=DA3=DA4=DB1=DB2=DB3=DB4=0.
319         U1=D(17)
320         U3=D(18)
321         IF(U1.EQ.0.)GO TO 1
322         CALL TILT(D(17),X1,X2,X3,D(11),D(12),A1,B1)
323         CALL TILT(D(17),X1,X2,X3,D(9),D(10),A2,B2)
324         CALL TILT(D(17),X1,X2,X3,D(13),D(14),A3,B3)
325         CALL TILT(D(17),X1,X2,X3,D(15),D(16),A4,B4)
326       1 IF(U3.EQ.0.)GO TO 2
327         CALL DPSPTL(D(18),X1,X2,X3,D(3),D(4),DA1,DB1)
328         CALL DPSPTL(D(18),X1,X2,X3,D(1),D(2),DA2,DB2)
329         CALL DPSPTL(D(18),X1,X2,X3,D(5),D(6),DA3,DB3)
330         CALL DPSPTL(D(18),X1,X2,X3,D(7),D(8),DA4,DB4)
331       2 T1=A1-A2-A3+A4+DA1-DA2-DA3+DA4
332         T2=B1-B2-B3+B4+DB1-DB2-DB3+DB4
333         RETURN
334         END
335         SUBROUTINE TILT(U1,X1,X2,X3,P1,P3,T1,T2)
336         R=SQRT((X1-P1)**2+X2**2+(X3-P3)**2)
337         RP=R+P3
338         T1=(U1/12.5664)*(X2*(X1-P1)*(R*RP-(R+2.*P3)*(2.*R+P3)))/
339        1 (R**3*RP**2)
340         T2=(U1/12.5664)*(X2**2*(R*RP-(R+2.*P3)*(2.*R+P3))/(R**3*RP**2)
341        1 +(R+2.*P3)/(R*RP))
342         RETURN
343         END
344         SUBROUTINE DPSPTL(U3,X1,X2,X3,P1,P3,DT1,DT2)
345         R=SQRT(((X1-P1)**2)+(X2**2)+((X3-P3)**2))
346         DT1=(U3/6.28318)*((X2*P3)/R)*((1./(R**2))-(1./(((X1-P1)
347        1 **2)+(X2**2))))
348         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 WRITE1
       WRITE(7,1)
     1 FORMAT(*SPECIFY INITIAL AND FINAL SLIP VALUES,*,/,
     1 *RIGHT-LATERAL STRIKE-SLIP IS <0.*,/,
     1 *DIP-SLIP >0. FOR *X2>0. SIDE* DOWN*,/,
     1 *FORMAT=U1IN,U1FN,U3IN,U3FN*)
       RETURN
       END
       SUBROUTINE WRITE2(C)
       DIMENSION C(40),A(30)
       WRITE(7,3)
     3 FORMAT(*SPECIFY INITIAL AND FINAL COORDINATES*,/,
     1 *OF DIP-SLIP ZONE CORNERS AS INDICATED.*,/,
     1 *D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,D3X1FN*)
       CALL GETNUM(A) $C(1)=A(1) $C(2)=A(2) $C(4)=A(3) $C(5)=A(4)
       C(9)=A(5) $C(10)=A(6) $C(12)=A(7) $C(13)=A(8)
       C(3)=C(1) $C(7)=C(5) $C(6)=C(4) $C(8)=C(2)
       C(11)=C(9) $C(16)=C(10) $C(14)=C(12) $C(15)=C(13)
       RETURN
       END
       SUBROUTINE WRITE3(C)
       REAL A(30)
       DIMENSION C(40)
       WRITE(7,8)
     8 FORMAT(*SPECIFY INITIAL AND FINAL VALUES *,/,
     1 *OF STRIKE-SLIP ZONE COORDINATES AS INDICATED*,/,
     1 *C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN*)
       CALL GETNUM(A) $C(17)=A(1) $C(18)=A(2) $C(20)=A(3) $C(21)=A(4)
       C(25)=A(5) $C(26)=A(6) $C(28)=A(7) $C(29)=A(8)
       C(19)=C(17) $C(22)=C(20) $C(23)=C(21) $C(24)=C(18)
       C(27)=C(25) $C(30)=C(28) $C(31)=C(29) $C(32)=C(26)
       RETURN
       END
       SUBROUTINE WRITE4(TRIGGR,M1,M2,M3,KCHECK)
       REAL A(30)
       M1=100 $M2=100 $M3=100
       WRITE(7,150)
   150 FORMAT(*DO YOU WISH THE *TRIGGER* OPTION?*,/,
     1 **YES* OR *NO **)
       READ(7,151)TRIGGR
   151 FORMAT(A3)
       KCHECK=0
       IF(TRIGGR.EQ.3HNO )GO TO 160
       WRITE(7,157)
   157 FORMAT(*SPECIFY *TRIGGER* PARAMETERS IN THE FORM*,/,
     1 *D(M1).GT.C(M2)*)
       CALL GETNUM(A) $M1=A(1) $M2=A(2)
       WRITE(7,158)
   158 FORMAT(*SPECIFY WHICH SLIP COMPONENT IS TO BE =0.*,/,
     1 *WHEN D(M1).LT.C(M2)*,/,
     1 *STRIKE-SLIP=S, DIP-SLIP=D*)
       READ(7,159)SLIP
   159 FORMAT(A1)
       IF(SLIP.EQ.1HS)M3=17
```

```
24 MAY 76    10.53.01          MCHUGH .XPND01

  1  DELETE(LGO,OUTPUT,XPND)
  2  XPND.
  3  CXIT.
  4  LIBCOPY(GRAPHIC,TXLGO/RR,TXLGO)
  5  LIBCOPY(JDRAT,NPLGO/RR,NPLGO)
  6  DELETE(LGO,OUTPUT,XPND)
  7  RUN76(S)
  8  LINK(F=LGO,F=TXLGO,F=NPLGO,B=XPND)
  9  XPND.
 10  FIN.
 11  EOR
 12        PROGRAM XPND(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
 13        COMMON/TVPCOL/TVPUL(8)
 14        COMMON/TVTUNE/ITUNE(30)
 15        COMMON/JPLOT/XLT,XRT,YLO,YUP,MAJX,MAJY,KX(2),KY(2),
 16       1LTITL(8),LU,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
 17        DIMENSION IFET(8)
 18        DIMENSION TEW(500),TNS(500),TAMP(500),TAZM(500),T(500)
 19        DIMENSION SLIPU1(500),SLIPL3(500),C4X1(500),D4X1(500)
 20        DIMENSION B(20),C(40),A(30),AREAS(500),AREAD(500)
 21        DATA KTIME,RTIME,KCHECK,M1,M2,M3,AAA/1.,1.,1.,1.,1.,1.,-1.234E-30/
 22        CALL FET(SLTAPE7,IFET,8)
 23        IFET(2)=IFET(2).OR.0000 0010 0000 0000 0000B
 24        IFET(8)=IFET(8).OR.4000 0000 0000 0000 0000B
 25        CALL FET(SLTAPE7,IFET,-8)
 26     20 CONTINUE
 27        MSTART=9 $WRITE(7,12)
 28     12 FORMAT(*1=ZONE EXPANDS*,/,
 29       1 *2=ZONE CONTRACTS*)
 30        CALL GETNUM (A) $IFLAG=A(1) $WRITE(7,15)
 31     15 FORMAT(*1=STRIKE-SLIP COMPONENT INCREMENTED EXPONENTIALLY*,/,
 32       1 *2=DIP-SLIP COMPONENT INCREMENTED EXPONENTIALLY*,/,
 33       1 *3=BOTH SLIP COMPONENTS INCREMENTED EXPONENTIALLY*,/,
 34       1 *4=NEITHER SLIP COMPONENT INCREMENTED EXPONENTIALLY*)
 35        CALL GETNUM (A) $LFLAG=A(1)
 36        CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK) $CALL WRITE1
 37        CALL GETNUM (A) $C(33)=A(1) $C(34)=A(2) $C(35)=A(3) $C(36)=A(4)
 38        DO 13 I=1,32
 39     13 C(I)=0.
 40    122 IF(C(35) .EQ. 0. .AND. C(36) .EQ. 0.) GO TO 100
 41        CALL WRITE2(C)
 42    100 IF(MSTART .EQ. 3) GO TO 123
 43        IF(C(33) .EQ. 0. .AND. C(34) .EQ. 0.) GO TO 101
 44                                    CALL WRITE3(C)
 45    101 IF(MSTART .EQ. 2) GO TO 123
 46        WRITE(7,4)
 47      4 FORMAT(*COORDINATES OF STATION (X1,X2)*)
 48        CALL GETNUM (A) $X1=A(1) $X2=A(2)  $THETA=45.
 49    123 WRITE(7,7)THETA
 50      7 FORMAT(*THETA=ANGLE BETWEEN STRIKE OF FAULT*,/,
 51       1 *AND NORTH=*,F10.3, DEGREES*)
 52        X3=0.  $NUM=500  $IJKL=(.075*NUM)+1.
 53        WRITE(7,6)
 54      6 FORMAT(*SPECIFY 2 CORNERS OF DISLOCATION SURFACE FOR DISPLAY*,/,
 55       1 *1=D1X1=D2X1          2=D1X3=D4X3*,/,
 56       1 *4=D2X3=D3X3          5=D3X1=D4X1*,/,
 57       1 *9=C1X1=C2X1         10=C1X3=C4X3*,/,
 58       1 *12=C2X3=C3X3        13=C3X1=C4X1*)
```

```
      CALL GETNUM(A) $MVAR1=A(1) $MVAR2=A(2)
      IF(C(35).EQ.0..AND.C(36).EQ.0.)GO TO 151
      IF(C(33).EQ.0..AND.C(34).EQ.0.)GO TO 152
      GO TO 153
151   DO 154 I=1,16
154   C(I)=0.
      GO TO 153
152   DO 155 I=16,32
155   C(I)=0.
153   CONTINUE
      NEND=NUM-IJKL  $TAU=-(NEND-IJKL)/6.
      DO 103 I=IJKL,NEND
      DO 104 K=1,16
      IF(K.LE.8)L1=K   $IF(K.LE.8)L2=K+8
      IF(K.GT.8.AND.K.LE.16)L1=K+8
      IF(K.GT.8.AND.K.LE.16)L2=K+16
104   D(K)=(((C(L2)-C(L1))/(NEND-IJKL))*(I-IJKL))+C(L1)
      IF(LFLAG.EQ.1.OR.LFLAG.EQ.3)D(17)=((1.-EXP((I-IJKL)/TAU))*(C(34)-C
     1(33)))+C(33)
      IF(LFLAG.EQ.2.OR.LFLAG.EQ.3)D(18)=((1.-EXP((I-IJKL)/TAU))*(C(36)-C
     1(35)))+C(35)
      IF(LFLAG.EQ.1)D(18)=(((C(36)-C(35))/(NEND-IJKL))*(I-IJKL))+C(35)
      IF(LFLAG.EQ.2)D(17)=(((C(34)-C(33))/(NEND-IJKL))*(I-IJKL))+C(33)
      IF(LFLAG.EQ.4)D(17)=(((C(34)-C(33))/(NEND-IJKL))*(I-IJKL))+C(33)
      IF(LFLAG.EQ.4)D(18)=(((C(36)-C(35))/(NEND-IJKL))*(I-IJKL))+C(35)
      IF(TRIGGR.NE.3HYES)GO TO 156
      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2))GO TO 114
      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2))KCHECK=KCHECK+1
      IF(KCHECK.EQ.1)KTIME=NEND-I
      IF(KCHECK.EQ.1)RTIME=I $IF(M3.EQ.18)GOTO 180
      DO 181 K=9,16
      L1=K+8  $L2=K+16
181   D(K)=(((C(L2)-C(L1))/(KTIME))*(I-RTIME))+C(L1)
      GO TO 114
180   DO 182 K=1,8
      L1=K   $L2=K+8
182   D(K)=(((C(L2)-C(L1))/(KTIME))*(I-RTIME))+C(L1)
114   IF(TRIGGR.EQ. 3HYES.AND.D(M1).LT.C(M2))D(M3)=0.
      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.M3.EQ.17)GO TO 9
      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.M3.EQ.18)GO TO 10
      GO TO 11
  9   DO 16 M=9,16
 16   D(M)=C(M+8) $GOTO11
 10   DO 18 M=1,8
 18   D(M)=C(M)
 11   IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.C(M3).EQ.0.)GO TO 156
      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2) .AND.LFLAG.EQ.1
     1 .AND.M3.EQ.17)D(M3)=((1.-EXP( (I-RTIME)/
     1 (-KTIME/6.)))*(C(34)-C(33)))+C(33)
      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.3
     1 .AND. M3.EQ.17)D(M3)=((1.-EXP((I-RTIME)/(-KTIME/6.)))*(C(34)-C(33
     1)))+C(33)
      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.2
     1 .AND.M3.EQ.18)D(M3)=((1.-EXP((I-RTIME)/
     1 (-KTIME/6.)))*(C(36)-C(35)))+C(35)
      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.3.AND.
     1 M3.EQ.18)D(M3)=((1.-EXP((I-RTIME)/(-KTIME/6.)))*(C(36)-C(35)))+C(
     135)
```

```
117         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.1
118        1  .AND.M3.EQ.18)D(M3)=(((C(36)-C(35))/
119        1  (KTIME))*(I-RTIME))+C(35)
120         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.4.AND.M3.EQ.18)
121        1  D(M3)=(((C(36)-C(35))/(KTIME))*(I-RTIME))+C(35)
122         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.2
123        1  .AND.M3.EQ.17) D(M3)=(((C(34)-C(33))/
124        1  (KTIME))*(I-RTIME))+C(33)
125         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.M3.EQ.17.AND.LFLAG.EQ.4)
126        1  D(M3)=(((C(34)-C(33))/(KTIME))*(I-RTIME))+C(33)
127     156 CALL CMPTLT(D,X1,X2,Y3,T1,T2)
128         IF(D(1).NE.D(3).OR.D(5).NE.D(7).OR.D(2).NE.D(8)
129        1  .OR.D(4).NE.D(6).OR.D(9).NE.D(11).OR.D(13).NE.D(15)
130        1  .OR.D(10).NE.D(16).OR.D(12).NE.D(14))WRITE(7,67)
131      67 FORMAT(*CORNERS NOT INCREMENTING PROPERLY*)
132         AREAS(I)=(D(15)-D(9))*(D(12)-D(10))
133         AREAD(I)=(D(7)-D(1))*(D(4)-D(2))
134         SLIPU1(I)=D(17) $SLIPU3(I)=D(18)
135         C4X1(I)=C(MVAR1) $D4X1(I)=C(MVAR2) $T(I)=I $TEW(I)=T2
136     103 TNS(I)=T1
137         K7=IJKL-1
138         DO 106 I=1,K7
139         AREAS(I)=AREAS(IJKL) $AREAD(I)=AREAD(IJKL) $TEW(I)=TEW(IJKL)
140         SLIPU1(I)=SLIPU1(IJKL) $SLIPU3(I)=SLIPU3(IJKL)
141         C4X1(I)=C4X1(IJKL) $D4X1(I)=D4X1(IJKL) $T(I)=I
142     106 TNS(I)=TNS(IJKL)
143         K8=NEND+1
144         DO 107 I=K8,NUM
145         SLIPU1(I)=SLIPU1(NEND) $SLIPU3(I)=SLIPU3(NEND) $T(I)=I
146         C4X1(I)=C4X1(NEND) $D4X1(I)=D4X1(NEND) $AREAD(I)=AREAD(NEND)
147         AREAS(I)=AREAS(NEND) $TEW(I)=TEW(NEND)
148     107 TNS(I)=TNS(NEND)
149         IF(IFLAG.EQ. 2)GO TO 108
150         A=TEW(1)     $B=TNS(1)
151         DO 109 I=1,NUM
152         TEW(I)=TEW(I)-A
153     109 TNS(I)=TNS(I)-B
154         GO TO 110
155     108 DO 111 I=1,NUM
156         TEW(I)=TEW(NUM)-TEW(I)
157     111 TNS(I)=TNS(NUM)-TNS(I)
158     110 CT=COS(THETA*.01745)
159         ST=SIN(THETA *.01745)
160         DO 21 I=1,NUM
161         B=TEW(I)  $A=TNS(I)
162         TNS(I)=A*CT+B*ST
163
164          TEW(I)=-A*ST+B*CT
165         TAMP(I)=SQRT((TEW(I)**2)+(TNS(I)**2))
166         IF(TNS(I).EQ.0.)TNS(I)=1.E-20
167         TAZM(I)=(ATAN(TEW(I)/TNS(I)))*(180./3.1415926)
168         IF(TNS(I).LT.0.)TAZM(I)=TAZM(I)+180.
169         IF(TAZM(I).LT.0.)TAZM(I)=TAZM(I)+360.
170      21 IF(TAZM(1).GT.360.)TAZM(I)=TAZM(I)-360.
171         TEWMID=TEW(1) $TNSMID=TNS(1) $TAMPMD=TAMP(1) $TAZMMD=TAZM(1)
172         C4X1MD=C4X1(1) $D4X1MD=D4X1(1)
173         TEWMIN=TEWMAX=TEWMID $TNSMIN=TNSMAX=TNSMID
174         TAMPMN=TAMPMX=TAMPMD $TAZMMN=TAZMMX=TAZMMD
```

```
      C4X1MN=C4X1MX=C4X1MD $D4X1MN=D4X1MX=D4X1MD
      DO 160 I=1,NUM
      IF(C4X1(I).LT.C4X1MN)C4X1MN=C4X1(I)
      IF(C4X1(I).GT.C4X1MX)C4X1MX=C4X1(I)
      IF(D4X1(I).LT.D4X1MN)D4X1MN=D4X1(I)
       IF(D4X1(I).GT.D4X1MX)D4X1MX=D4X1(I)
      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)
  160 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 *0=RE-START, 1=CONTINUE*)
      CALL GETNUM(A) $IKFLAG=A(1)
      IF(IKFLAG.EQ. 0) GO TO 113
      LU=7  $LNLGX=1  $LNLGY=1 $NCLX=2  $NCLY=2
      WRITE(7,199)
  199 FORMAT(*WRITE PLOT TITLE, 80 CHARACTERS*)
      READ(7,200)(LTITL(JM),JM=1,8)
  200 FORMAT(8A10)
      KX(1)=10HTIME-ARBIT
      KX(2)=10HRARY UNITS
      KY(1)=10H EW TILT
      KY(2)=10H           $MAJX=5 $MAJY=10 $LTITL2(1)=1
  903 XLT=T(1)$XRT=T(NUM)  $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(A)  $XLT=A(1)   $XRT=A(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(A)  $YLO=A(1)   $YUP=A(2)
   74 IF(TEWMIN.EQ.TEWMAX)YLO=YUP-1.
      IF(TEWMIN.EQ.TEWMAX)YUP=YUP+1.
      WRITE(7,1)
    1 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,NUM)
```

```
      WRITE(7,19) $CALL GETNUM(A)$IKFLAG=A(1)
      IF(IKFLAG.EQ.0)GOTO113$IF(IKFLAG.EQ.1)GOTO903
  900 KY(1)=10H NS TILT     $XLT=T(1)$XRT=T(NUM)
      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(A)$XLT=A(1)$XRT=A(2)
  710 WRITE(7,73)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO740
      WRITE(7,75)$CALL GETNUM(A),YLO=A(1)$YUP=A(2)
  740 IF(YLO.EQ.YUP)AAA=YLO$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,NUM)
      WRITE(7,19)
   19 FORMAT(*0=RE-START, 1=NEW PLOT*)
      CALL GETNUM(A) $IKFLAG=A(1)
      IF(IKFLAG.EQ.0)GO TO 113 $IF(IKFLAG.EQ.1)GOTO900
  911 KY(1)=10HTILT AMPLI  $XLT=T(1) $XRT=T(NUM)
      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(A) $XLT=A(1)$XRT=A(2)
  741 WRITE(7,73)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO742
      WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(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) $READ(7,66)IJVAR
  912 FORMAT(*SKIP PLOT OF TILT AMPLITUDE?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAMP,T,1,NUM)
      WRITE(7,19)$CALL GETNUM(A)$IKFLAG=A(1)
      IF(IKFLAG.EQ.0)GOTO113$IF(IKFLAG.EQ.1)GOTO911
  913 KY(1)=10HTILT AZIMU  $XLT=T(1)$XRT=T(NUM)
      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(A)$XLT=A(1)$XRT=A(2)
  743 WRITE(7,73)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO744
      WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(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,NUM)
      WRITE(7,19) $CALL GETNUM(A) $IKFLAG=A(1)
      IF(IKFLAG.EQ.0)GO TO 113$IF(IKFLAG.EQ.1)GOTO913
  915 XLT=T(1)$XRT=T(NUM)
      KY(1)=10HSTRIKESLIP $KY(2)=10H COMPONENT
      YLO=C(33)  $YUP=C(34)
      IF(C(33).EQ.C(34))YLO=YUP-1.
      IF(C(33).EQ.C(34))YUP=YUP+1.
      WRITE(7,916) $READ(7,66)IJVAR
  916 FORMAT(*SKIP PLOT OF STRIKE SLIP COMPONENT?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(SLIPU1,T,1,NUM)
      KY(1)=10HDIP SLIP C $KY(2)=10HOMPONENT
```

```
      YLO=C(35) $YUP=C(36)
      IF(C(35).EQ.C(36))YLO=YUP-1.
      IF(C(35).EQ.C(36))YUP=YUP+1.
      WRITE(7,917) $READ(7,66)IJVAR
  917 FORMAT(*SKIP PLOT OF DIP SLIP COMPONENT?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(SLIPU3,T,1,NUM)
      WRITE(7,919) $CALL GETNUM(A) $IKFLAG=A(1)
  919 FORMAT(*0=RE-START*)
      IF(IKFLAG.EQ.0)GO TO 113
      KY(1)=10HCORNER 1
      IF(MVAR1.EQ.1)KY(2)=10H D1X1=D2X1
      IF(MVAR1.EQ.2)KY(2)=10H D1X3=D4X3
      IF(MVAR1.EQ.4)KY(2)=10H D2X3=D3X3
      IF(MVAR1.EQ.5)KY(2)=10H D3X1=D4X1
      IF(MVAR1.EQ.9)KY(2)=10H C1X1=C2X1
      IF(MVAR1.EQ.10)KY(2)=10H C1X3=C4X3
      IF(MVAR1.EQ.12)KY(2)=10H C2X3=C3X3
      IF(MVAR1.EQ.13)KY(2)=10H C3X1=C4X1
      YLO=C4X1MN $YUP=C4X1MX
      IF(C4X1MN.EQ.C4X1MX)YLO=YUP-1.
      IF(C4X1MN.EQ.C4X1MX)YUP=YUP+1.
      WRITE(7,918) $READ(7,66)IJVAR
  918 FORMAT(*SKIP PLOT OF CORNER 1?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(C4X1,T,1,NUM)
      KY(1)=10HCORNER 2
      IF(MVAR2.EQ.1)KY(2)=10H D1X1=D2X1
      IF(MVAR2.EQ.2)KY(2)=10H D1X3=D4X3
      IF(MVAR2.EQ.4)KY(2)=10H D2X3=D3X3
      IF(MVAR2.EQ.5)KY(2)=10H D3X1=D4X1
      IF(MVAR2.EQ.9)KY(2)=10H C1X1=C2X1
      IF(MVAR2.EQ.10)KY(2)=10H C1X3=C4X3
      IF(MVAR2.EQ.12)KY(2)=10H C2X3=C3X3
      IF(MVAR2.EQ.13)KY(2)=10H C3X1=C4X1
      YLO=D4X1MN $YUP=D4X1MX
      IF(D4X1MN.EQ.D4X1MX)YLO=YUP-1.
      IF(D4X1MN.EQ.D4X1MX)YUP=YUP+1.
      WRITE(7,920) $READ(7,66)IJVAR
  920 FORMAT(*SKIP PLOT OF CORNER 2?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(D4X1,T,1,NUM)
      KY(1)=10HS.S.  AREA
      YLO=(C(23)-C(17))*(C(20)-C(18))
      YUP=(C(31)-C(25))*(C(28)-C(26))
      ABC=YLO $IF(YLO.EQ.YUP)ABC=YLO-1. $IF(YLO.EQ.YUP)YUP=YUP+1.
      YLO=ABC $WRITE(7,921) $READ(7,66)IJVAR
  921 FORMAT(*SKIP PLOT OF STRIKE SLIP AREA?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(AREAS,T,1,NUM)
      KY(1)=10HD.S.  AREA
      YLO=(C(7)-C(1))*(C(4)-C(2))
      YUP=(C(15)-C(9))*(C(12)-C(10))
      ABC=YLO $IF(YLO.EQ.YUP)ABC=YLO-1. $IF(YLO.EQ.YUP)YUP=YUP+1.
      YLO=ABC $WRITE(7,922) $READ(7,66)IJVAR
  922 FORMAT(*SKIP PLOT OF DIP SLIP AREA?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(AREAD,T,1,NUM)
      WRITE(7,68)(C(I),I=1,36)
   68 FORMAT(9(4F12.2,/))
      WRITE(7,69)IFLAG,IKFLAG,LFLAG,IJKL,NUM,NENC,M1,M2,M3,KTIME,K7,K8,
     1 TAU,RTIME
   69 FORMAT(7I10,/,5I10,2F10.3)
```

```
      113 WRITE(7,115)
      115 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*)
          CALL GETNUM(A) $MSTART=A(1)
          IF(MSTART.EQ.2)GOTO116 $IF(MSTART.EQ.3)GOTO117
          IF(MSTART.EQ.4)GOTO118 $IF(MSTART.EQ.5)GOTO119
          IF(MSTART.EQ.6)GOTO120 $IF(MSTART.EQ.7)GOTO630
          GO TO 20
      116 WRITE (7,12)
          CALL GETNUM(A) $IFLAG=A(1)  $WRITE(7,15)
          CALL GETNUM(A) $LFLAG=A(1)
          WRITE(7,23)
      23  FORMAT(*U1IN,U1FN*)
          CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
          CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
          GO TO 100
      117 WRITE(7,12)
          CALL GETNUM(A) $IFLAG=A(1)
          WRITE(7,15)
          CALL GETNUM(A) $LFLAG=A(1)
          WRITE(7,25)
      25  FORMAT(*U3IN,U3FN*)
          CALL GETNUM(A) $C(35)=A(1) $C(36)=A(2)
          CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
          GO TO 122
      118 WRITE(7,12)
          CALL GETNUM(A) $IFLAG=A(1)
          WRITE(7,15)
          CALL GETNUM(A) $LFLAG=A(1)
          WRITE(7,23)
          CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
          CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
          GO TO 123
      119 WRITE(7,12)
          CALL GETNUM(A) $IFLAG=A(1)
          WRITE(7,15)
          CALL GETNUM(A) $LFLAG=A(1)
          WRITE(7,25)
          CALL GETNUM(A) $C(35)=A(1) $C(36)=A(2)
          CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
          GO TO 123
      120 WRITE(7,4)
          CALL GETNUM(A) $X1=A(1) $X2=A(2)
          CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
          GO TO 123
      630 STOP
          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 WRITE1
      WRITE(7,1)
    1 FORMAT(*SPECIFY INITIAL AND FINAL SLIP VALUES,*,/,
     1 *RIGHT-LATERAL STRIKE-SLIP IS <0.*,/,
     1 *DIP-SLIP >0. FOR 'X2>0. SIDE' DOWN*,/,
     1 *FORMAT=U1IN,U1FN,U3IN,U3FN*)
      RETURN
      END
      SUBROUTINE WRITE2(C)
      DIMENSION C(40),A(30)
      WRITE(7,3)
    3 FORMAT(*SPECIFY INITIAL AND FINAL COORDINATES*,/,
     1 *OF DIP-SLIP ZONE CORNERS AS INDICATED.*,/,
     1 *D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,D3X1FN*)
      CALL GETNUM(A) $C(1)=A(1) $C(2)=A(2) $C(4)=A(3) $C(5)=A(4)
      C(9)=A(5) $C(10)=A(6) $C(12)=A(7) $C(13)=A(8)
      C(3)=C(1)  $C(7)=C(5)  $C(6)=C(4)  $C(8)=C(2)
      C(11)=C(9)  $C(16)=C(10)  $C(14)=C(12)  $C(15)=C(13)
      RETURN
      END
      SUBROUTINE WRITE3(C)
      REAL A(30)
      DIMENSION C(40)
      WRITE(7,8)
    8 FORMAT(*SPECIFY INITIAL AND FINAL VALUES *,/,
     1 *OF STRIKE-SLIP ZONE COORDINATES AS INDICATED*,/,
```

```
      1 *C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN*)
        CALL GETNUM(A) $C(17)=A(1) $C(18)=A(2) $C(20)=A(3) $C(21)=A(4)
        C(25)=A(5) $C(26)=A(6) $C(28)=A(7) $C(29)=A(8)
        C(19)=C(17) $C(22)=C(20) $C(23)=C(21) $C(24)=C(18)
        C(27)=C(25) $C(30)=C(28) $C(31)=C(29) $C(32)=C(26)
        RETURN
        END
        SUBROUTINE WRITE4(TRIGGR,M1,M2,M3,KCHECK)
        REAL A(30)
        M1=100 $M2=100 $M3=100
        WRITE(7,150)
  150 FORMAT(*DO YOU WISH THE 'TRIGGER' OPTION?*,/,
      1 *'YES' OR 'NO '*)
        READ(7,151)TRIGGR
  151 FORMAT(A3)
        KCHECK=0
        IF(TRIGGR.EQ.3HNO )GO TO 160
        WRITE(7,157)
  157 FORMAT(*SPECIFY 'TRIGGER' PARAMETERS IN THE FORM*,/,
      1 *D(M1).GE.C(M2)*)
        CALL GETNUM(A) $M1=A(1) $M2=A(2)
        WRITE(7,158)
  158 FORMAT(*SPECIFY WHICH SLIP COMPONENT IS TO BE =0.*,/,
      1 *WHEN D(M1).LT.C(M2)*,/,
      1 *STRIKE-SLIP=S, DIP-SLIP=D*)
        READ(7,159)SLIP
  159 FORMAT(A1)
        IF(SLIP.EQ.1HS)M3=17
        IF(SLIP.EQ.1HD)M3=18
  160 CONTINUE
        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
```

LISTING OF XPND02

```
 1  DELETE(LGO,OUTPUT,XPND)
 2  XPND.
 3  CXIT.
 4  LIBCOPY(GRAPHIC,TXLGO/RR,TXLGC)
 5  LIBCOPY(JDRAT,NPLGO/FR,NPLGC)
 6  DELETE(LGO,OUTPUT,XPND)
 7  RUN76(S)
 8  LINK(F=LGO,F=TXLGO,F=NPLGO,B=XPND)
 9  XPND.
10  FIN.
11  EOR
12        PROGRAM XPND(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
13        COMMON/TVPCOL/TVPOL(8)
14        COMMON/TVTONE/ITONE(30)
15        COMMON/JPLOT/XLT,XRT,YLO,YUP,MAJX,MAJY,KX(2),KY(2),
16       1LTITL(8),LU,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
17        DIMENSION IFET(8)
18        DIMENSION TEW(500),TNS(500),TAMP(500),TAZM(500),T(500)
19        DIMENSION SLIPO1(500),SLIPO3(500),C4X1(500),D4X1(500)
20        DIMENSION D(20),C(40),A(30),AREAS(500),AREAD(500)
21        DATA KTIME,ETIME,KCHECK,M1,M2,M3,AAA/1.,1.,1.,1.,1.,1.,-1.234E-30/
22        CALL FET(5LTAPE7,IFET,8)
23        IFET(2)=IFET(2).OR.0000 0010 0000 0000 0000B
24        IFET(8)=IFET(8).OR.4000 0000 0000 0000 0000B
25        CALL FET(5LTAPE7,IFET,-8)
26     20 CONTINUE
27        MSTART=9 $WRITE(7,12)
28     12 FORMAT(*1-ZONE EXPANDS*,/,
29       1 *2=ZONE CONTRACTS*)
30        CALL GETNUM (A) $IFLAG=A(1) $WRITE(7,15)
31     15 FORMAT(*1=STRIKE-SLIP COMPONENT INCREMENTED EXPONENTIALLY*,/,
32       1 *2=DIP-SLIP CONPONENT INCREMENTED EXPONENTIALLY*,/,
33       1 *3=BOTH SLIP COMPONENTS INCREMENTED EXPONENTIALLY*,/,
34       1 *4=NEITHER SLIP COMPONENT INCREMENTED EXPONENTIALLY*)
35        CALL GETNUM (A) $LFLAG=A(1)
36        CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK) $CALL WRITE1
37        CALL GETNUM (A) $C(33)=A(1) $C(34)=A(2) $C(35)=A(3) $C(36)=A(4)
38        DO 13 I=1,32
39     13 C(I)=0.
40    122 IF(C(35) .EQ. 0. .AND. C(36) .EQ. 0.) GO TO 100
41        CALL WRITE2(C)
42    100 IF(MSTART .EQ. 3) GO TO 123
43        IF(C(33) .EQ. 0. .AND. C(34) .EQ. 0.) GO TO 101
44        CALL WRITE3(C)
45    101 IF(MSTART .EQ. 2) GO TO 123
46        WRITE(7,4)
47      4 FORMAT(*COORDINATES OF STATION (X1,X2)*)
48        CALL GETNUM (A) $X1=A(1) $X2=A(2) $THETA=45.
49    123 WRITE(7,7) THETA
50      7 FORMAT(*THETA=ANGLE BETWEEN STRIKE OF FAULT*,/,
51       1 *AND NORTH=*,F10.3,* DEGREES*)
52        X3=0.  $NUM=500  $IJKL=(.075*NUM)+1.
53        WRITE(7,6)
54      6 FORMAT(*SPECIFY 2 CORNERS OF DISLOCATION SURFACE FOR DISPLAY*,/,
55       1 *1=D1X1=D2X1          2=D1X3=D4X3*,/,
56       1 *4=D2X3=D3X3          5=D3X1=D4X1*,/,
57       1 *9=C1X1=C2X1         10=C1X3=C4X3*,/,
58       1 *12=C2X3=C3X3        13=C3X1=C4X1*)
```

```
      CALL GETNUM(A) $MVAR1=A(1) $MVAR2=A(2)
      IF(C(35).EQ.0..AND.C(36).EQ.0.)GO TO 151
      IF(C(33).EQ.0..AND.C(34).EQ.0.)GO TO 152
      GO TO 153
  151 DO 154 I=1,16
  154 C(I)=0.
      GO TO 153
  152 DO 155 I=16,32
  155 C(I)=0.
  153 CONTINUE
      NEND=NUM-IJKL  $TAU=-(NEND-IJKL)/6.
      DO 103 I=IJKL,NEND
      DO 104 K=1,16
      IF(K.LE.8)L1=K   $IF(K.LE.8)L2=K+8
      IF(K.GT.8.AND.K.LE.16)L1=K+8
      IF(K.GT.8.AND.K.LE.16)L2=K+16
  104 D(K)=(((C(L2)-C(L1))/(NEND-IJKL))*(I-IJKL))+C(L1)
      IF(LFLAG.EQ.1.OR.LFLAG.EQ.3)D(17)=((1.-EXP((I-IJKL)/TAU))*(C(34)-C
     1(33)))+C(33)
      IF(LFLAG.EQ.2.OR.LFLAG.EQ.3)D(18)=((1.-EXP((I-IJKL)/TAU))*(C(36)-C
     1(35)))+C(35)
      IF(LFLAG.EQ.1)D(18)=(((C(36)-C(35))/(NEND-IJKL))*(I-IJKL))+C(35)
      IF(LFLAG.EQ.2)D(17)=(((C(34)-C(33))/(NEND-IJKL))*(I-IJKL))+C(33)
      IF(LFLAG.EQ.4)D(17)=(((C(34)-C(33))/(NEND-IJKL))*(I-IJKL))+C(33)
      IF(LFLAG.EQ.4)D(18)=(((C(36)-C(35))/(NEND-IJKL))*(I-IJKL))+C(35)
      IF(TRIGGR.NE.3HYES)GO TO 156
      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2))GO TO 114
      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2))KCHECK=KCHECK+1
      IF(KCHECK.EQ.1)KTIME=NEND-I
      IF(KCHECK.EQ.1)RTIME=I $IF(M3.EQ.18)GOTO 180
      DO 181 K=9,16
      L1=K+8  $L2=K+16
  181 D(K)=(((C(L2)-C(L1))/(KTIME))*(I-RTIME))+C(L1)
      GO TO 114
  180 DO 182 K=1,8
      L1=K   $L2=K+8
  182 D(K)=(((C(L2)-C(L1))/(KTIME))*(I-RTIME))+C(L1)
  114 IF(TRIGGR.EQ. 3HYES.AND.D(M1).GE.C(M2))D(M3)=0.
      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.M3.EQ.17)GO TO 9
      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.M3.EQ.18)GO TO 10
      GO TO 11
    9 DO 16 M=9,16
   16 D(M)=C(M+8) $GOTO11
   10 DO 18 M=1,8
   18 D(M)=C(M)
   11 IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.C(M3).EQ.0.)GO TO 156
      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2) .AND.LFLAG.EQ.1
     1 .AND.M3.EQ.17)D(M3)=((1.-EXP((I-RTIME)/
     1 (-KTIME/6.)))*(C(34)-C(33)))+C(33)
      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.LFLAG.EQ.3
     1 .AND. M3.EQ.17)D(M3)=((1.-EXP((I-RTIME)/(-KTIME/6.)))*(C(34)-C(33
     1)))+C(33)
      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.LFLAG.EQ.2
     1 .AND.M3.EQ.18)D(M3)=((1.-EXP((I-RTIME)/
     1 (-KTIME/6.)))*(C(36)-C(35)))+C(35)
      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.LFLAG.EQ.3.AND.
     1 M3.EQ.18)D(M3)=((1.-EXP((I-RTIME)/(-KTIME/6.)))*(C(36)-C(35)))+C(
     135)
```

```
      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.LFLAG.EQ.1
     1 .AND.M3.EQ.18)D(M3)=(((C(36)-C(35))/
     1 (KTIML))*(I-RTIME))+C(35)
      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.LFLAG.EQ.4.AND.M3.EQ.18)
     1 D(M3)=(((C(36)-C(35))/(KTIME))*(I-RTIME))+C(35)
      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.LFLAG.EQ.2
     1 .AND.M3.EQ.17) D(M3)=(((C(34)-C(33))/
     1 (KTIME))*(I-RTIME))+C(33)
      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.M3.EQ.17.AND.LFLAG.EQ.4)
     1 D(M3)=(((C(34)-C(33))/(KTIME))*(I-RTIME))+C(33)
  156 CALL CMPTLT(D,X1,X2,X3,T1,T2)
      IF(D(1).NE.D(3).OR.D(5).NE.D(7).OR.D(2).NE.D(8)
     1 .OR.D(4).NE.D(6).OR.D(9).NE.D(11).OR.D(13).NE.D(15)
     1 .OR.D(10).NE.D(16).OR.D(12).NE.D(14))WRITE(7,67)
   67 FORMAT(*CORNERS NOT INCREMENTING PROPERLY*)
      AREAS(I)=(D(15)-D(9))*(D(12)-D(10))
      AREAD(I)=(D(7)-D(1))*(D(4)-D(2))
      SLIPU1(I)=D(17) $SLIPU3(I)=D(18)
      C4X1(I)=C(MVAR1) $D4X1(I)=C(MVAR2) $T(I)=I $TEW(I)=T2
  103 TNS(I)=T1
      K7=IJKL-1
      DO 106 I=1,K7
      AREAS(I)=AREAS(IJKL) $AREAD(I)=AREAD(IJKL) $TEW(I)=TEW(IJKL)
      SLIPU1(I)=SLIPU1(IJKL) $SLIPU3(I)=SLIPU3(IJKL)
      C4X1(I)=C4X1(IJKL) $D4X1(I)=D4X1(IJKL) $T(I)=I
  106 TNS(I)=TNS(IJKL)
      K8=NEND+1
      DO 107 I=K8,NUM
      SLIPU1(I)=SLIPU1(NEND) $SLIPU3(I)=SLIPU3(NEND) $T(I)=I
      C4X1(I)=C4X1(NEND) $D4X1(I)=D4X1(NEND) $AREAD(I)=AREAD(NEND)
      AREAS(I)=AREAS(NEND) $TEW(I)=TEW(NEND)
  107 TNS(I)=TNS(NEND)
      IF(IFLAG.EQ. 2)GO TO 108
      A=TEW(1)    $B=TNS(1)
      DO 109 I=1,NUM
      TEW(I)=TEW(I)-A
  109 TNS(I)=TNS(I)-B
      GO TO 110
  108 DO 111 I=1,NUM
      TEW(I)=TEW(NUM)-TEW(I)
  111 TNS(I)=TNS(NUM)-TNS(I)
  110 CT=COS(THETA*.01745)
      ST=SIN(THETA *.01745)
      DO 21 I=1,NUM
      B=TEW(I)  $A=TNS(I)
      TNS(I)=A*CT+B*ST
      TEW(I)=-A*ST+B*CT
      TAMP(I)=SQRT((TEW(I)**2)+(TNS(I)**2))
      IF(TNS(I).EQ.0.)TNS(I)=1.E-20
      TAZM(I)=(ATAN(TEW(I)/TNS(I)))*(180./3.1415926)
      IF(TNS(I).LT.0.)TAZM(I)=TAZM(I)+180.
      IF(TAZM(I).LT.0.)TAZM(I)=TAZM(I)+360.
   21 IF(TAZM(I).GT.360.)TAZM(I)=TAZM(I)-360.
      TEWMID=TEW(1) $TNSMID=TNS(1) $TAMPMD=TAMP(1) $TAZMMD=TAZM(1)
      C4X1MD=C4X1(1) $D4X1MD=D4X1(1)
      TEWMIN=TEWMAX=TEWMID $TNSMIN=TNSMAX=TNSMID
      TAMPMN=TAMPMX=TAMPMD $TAZMMN=TAZMMX=TAZMMD
      C4X1MN=C4X1MX=C4X1MD $D4X1MN=D4X1MX=D4X1MD
```

```
      DO 160 I=1,NUM
      IF(C4X1(I).LT.C4X1MN)C4X1MN=C4X1(I)
      IF(C4X1(I).GT.C4X1MX)C4X1MX=C4X1(I)
      IF(D4X1(I).LT.D4X1MN)D4X1MN=D4X1(I)
      IF(D4X1(I).GT.D4X1MX)D4X1MX=D4X1(I)
      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)
  160 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 *0=RE-START, 1=CONTINUE*)
      CALL GETNUM(A) $IKFLAG=A(1)
      IF(IKFLAG.EQ. 0) GO TO 113
      LU=7  $LNLGX=1  $LNLGY=1 $NCLX=2  $NCLY=2
      WRITE(7,199)
  199 FORMAT(*WRITE PLOT TITLE, 80 CHARACTERS*)
      READ(7,200)(LTITL(JM),JM=1,8)
  200 FORMAT(8A10)
      KX(1)=10HTIME-ARBIT
      KX(2)=10HRARY UNITS
      KY(1)=10H EW TILT
      KY(2)=10H            $MAJX=5 $MAJY=10 $LTITL2(1)=1
  903 XLT=T(1)$XPT=T(NUM)  $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(A)  $XLT=A(1)   $XRT=A(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(A)  $YLO=A(1)   $YUP=A(2)
   74 IF(TEWMIN.EQ.TEWMAX)YLO=YUP-1.
      IF(TEWMIN.EQ.TEWMAX)YUP=YUP+1.
      WRITE(7,1)
    1 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,NUM)
      WRITE(7,19) $CALL GETNUM(A)$IKFLAG=A(1)
```

```
233         IF(IKFLAG.EQ.0)GOTO113$IF(IKFLAG.EQ.1)GOTO903
234     900 KY(1)=10H NS TILT      $XLT=T(1)$XRT=T(NUM)
235         KY(2)=10H
236         YLO=TNSMIN $YUP=TNSMAX $WRITE(7,70) $READ(7,66)CHARAC
237         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO710
238         WRITE(7,72)$CALL GETNUM(A)$XLT=A(1)$XRT=A(2)
239     710 WRITE(7,73)$READ(7,66)CHARAC
240         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO740
241         WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
242     740 IF(YLO.EQ.YUP)AAA=YLO$IF(YLO.EQ.AAA)YUP=YUP+1.
243         IF(YLO.EQ.AAA)YLO=YLO-1.
244         WRITE(7,910) $READ(7,66)IJVAR
245     910 FORMAT(*SKIP PLOT OF NS TILT?*)
246         IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TNS,T,1,NUM)
247         WRITE(7,19)
248     19  FORMAT(*0=RE-START, 1=NEW PLOT*)
249         CALL GETNUM(A) $IKFLAG=A(1)
250         IF(IKFLAG.EQ.0)GO TO 113 $IF(IKFLAG.EQ.1)GOTO900
251     911 KY(1)=10HTILT AMPLI  $XLT=T(1) $XRT=T(NUM)
252         KY(2)=10HTUDE
253         YLO=TAMPMN $YUP=TAMPMX $WRITE(7,70)$READ(7,66)CHARAC
254         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO741
255         WRITE(7,72)$CALL GETNUM(A) $XLT=A(1)$XRT=A(2)
256     741 WRITE(7,73)$READ(7,66)CHARAC
257         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO742
258         WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
259     742 IF(YLO.EQ.YUP)AAA=YLO$IF(YLO.EQ.AAA)YUP=YUP+1.
260         IF(YLO.EQ.AAA)YLO=YLO-1.
261         WRITE(7,912) $READ(7,66)IJVAR
262     912 FORMAT(*SKIP PLOT OF TILT AMPLITUDE?*)
263         IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAMP,T,1,NUM)
264         WRITE(7,19)$CALL GETNUM(A)$IKFLAG=A(1)
265         IF(IKFLAG.EQ.0)GOTO113$IF(IKFLAG.EQ.1)GOTO911
266     913 KY(1)=10HTILT AZIMU  $XLT=T(1)$XRT=T(NUM)
267         KY(2)=10HTH          $YLO=TAZMMN  $YUP=TAZMMX
268         WRITE(7,70) $READ(7,66)CHARAC
269         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO743
270         WRITE(7,72)$CALL GETNUM(A)$XLT=A(1)$XRT=A(2)
271     743 WRITE(7,73)$READ(7,66)CHARAC
272         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO744
273         WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
274     744 IF(YLO.EQ.YUP)AAA=YLO$IF(YLO.EQ.AAA)YLO=YUP+1.
275         IF(YLO.EQ.AAA)YLO=YLO-1.
276         WRITE(7,914) $READ(7,66)IJVAR
277     914 FORMAT(*SKIP PLOT OF TILT AZIMUTH?*)
278         IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAZM,T,1,NUM)
279         WRITE(7,19) $CALL GETNUM(A) $IKFLAG=A(1)
280         IF(IKFLAG.EQ.0)GO TO 113$IF(IKFLAG.EQ.1)GOTO913
281     915 XLT=T(1)$XRT=T(NUM)
282         KY(1)=10HSTRIKE SLIP $KY(2)=10H COMPONENT
283         YLO=C(33)  $YUP=C(34)
284         IF(C(33).EQ.C(34))YLO=YUP-1.
285         IF(C(33).EQ.C(34))YUP=YUP+1.
286         WRITE(7,916) $READ(7,66)IJVAR
287     916 FORMAT(*SKIP PLOT OF STRIKE SLIP COMPONENT?*)
288         IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(SLIPU1,T,1,NUM)
289         KY(1)=10HDIP SLIP C $KY(2)=10HOMPONENT
290         YLO=C(35) $YUP=C(36)
```

```
      IF(C(35).EQ.C(36))YLO=YUP-1.
      IF(C(35).EQ.C(36))YUP=YUP+1.
      WRITE(7,917) $READ(7,66)IJVAR
  917 FORMAT(*SKIP PLOT OF DIP SLIP COMPONENT?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(SLIPU3,T,1,NUM)
      WRITE(7,919) $CALL GETNUM(A) $IKFLAG=A(1)
  919 FORMAT(*0=RE-START*)
      IF(IKFLAG.EQ.0)GO TO 113
      KY(1)=10HCORNER 1
      IF(MVAR1.EQ.1)KY(2)=10H D1X1=D2X1
      IF(MVAR1.EQ.2)KY(2)=10H D1X3=D4X3
      IF(MVAR1.EQ.4)KY(2)=10H D2X3=D3X3
      IF(MVAR1.EQ.5)KY(2)=10H D3X1=D4X1
      IF(MVAR1.EQ.9)KY(2)=10H C1X1=C2X1
      IF(MVAR1.EQ.10)KY(2)=10H C1X3=C4X3
      IF(MVAR1.EQ.12)KY(2)=10H C2X3=C3X3
      IF(MVAR1.EQ.13)KY(2)=10H C3X1=C4X1
      YLO=C4X1MN $YUP=C4X1MX
      IF(C4X1MN.EQ.C4X1MX)YLO=YUP-1.
      IF(C4X1MN.EQ.C4X1MX)YUP=YUP+1.
      WRITE(7,918) $READ(7,66)IJVAR
  918 FORMAT(*SKIP PLOT OF CORNER 1?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(C4X1,T,1,NUM)
      KY(1)=10HCORNER 2
      IF(MVAR2.EQ.1)KY(2)=10H D1X1=D2X1
      IF(MVAR2.EQ.2)KY(2)=10H D1X3=D4X3
      IF(MVAR2.EQ.4)KY(2)=10H D2X3=D3X3
      IF(MVAR2.EQ.5)KY(2)=10H D3X1=D4X1
      IF(MVAR2.EQ.9)KY(2)=10H C1X1=C2X1
      IF(MVAR2.EQ.10)KY(2)=10H C1X3=C4X3
      IF(MVAR2.EQ.12)KY(2)=10H C2X3=C3X3
      IF(MVAR2.EQ.13)KY(2)=10H C3X1=C4X1
      YLO=D4X1MN $YUP=D4X1MX
      IF(D4X1MN.EQ.D4X1MX)YLO=YUP-1.
      IF(D4X1MN.EQ.D4X1MX)YUP=YUP+1.
      WRITE(7,920) $READ(7,66)IJVAR
  920 FORMAT(*SKIP PLOT OF CORNER 2?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(D4X1,T,1,NUM)
      KY(1)=10HS.S.  AREA
      YLO=(C(23)-C(17))*(C(20)-C(18))
      YUP=(C(31)-C(25))*(C(28)-C(26))
      ABC=YLO $IF(YLO.EQ.YUP)ABC=YLO-1. $IF(YLO.EQ.YUP)YUP=YUP+1.
      YLO=ABC $WRITE(7,921) $READ(7,66)IJVAR
  921 FORMAT(*SKIP PLOT OF STRIKE SLIP AREA?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(AREAS,T,1,NUM)
      KY(1)=10HD.S.  AREA
      YLO=(C(7)-C(1))*(C(4)-C(2))
      YUP=(C(15)-C(9))*(C(12)-C(10))
      ABC=YLO $IF(YLO.EQ.YUP)ABC=YLO-1. $IF(YLO.EQ.YUP)YUP=YUP+1.
      YLO=ABC $WRITE(7,922) $READ(7,66)IJVAR
  922 FORMAT(*SKIP PLOT OF DIP SLIP AREA?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(AREAD,T,1,NUM)
      WRITE(7,68)(C(I),I=1,36)
   68 FORMAT(9(4F12.2,/))
      WRITE(7,69)IFLAG,IKFLAG,LFLAG,IJKL,NUM,NENC,M1,M2,M3,KTIME,K7,K8,
     1 TAU,PTIME
   69 FORMAT(7I10,/,5I10,2F10.3)
  113 WRITE(7,115)
```

```
349     115 FORMAT(*1=RE-START WITH ALL NEW VALUES*,/,
350        1 *2=RE-START WITH NEW STRIKE-SLIP VALUE AND ZONE COORDINATES*,/,
351        1 *3=RE-START WITH NEW DIP-SLIP VALUE AND ZONE COORDINATES*,/,
352        1 *4=RE-START WITH NEW STRIKE-SLIP VALUE ONLY*,/,
353        1 *5=RE-START WITH NEW DIP-SLIP VALUE ONLY*,/,
354        1 *6=RE-START WITH NEW TILTMETER COORDINATES ONLY*,/,
355        1 *7=STOP*)
356         CALL GETNUM(A) $MSTART=A(1)
357         IF(MSTART.EQ.2)GOTO116 $IF(MSTART.EQ.3)GOTO117
358         IF(MSTART.EQ.4)GOTO118 $IF(MSTART.EQ.5)GOTO119
359         IF(MSTART.EQ.6)GOTO120 $IF(MSTART.EQ.7)GOTO630
360         GO TO 20
361     116 WRITE (7,12)
362         CALL GETNUM(A) $IFLAG=A(1)  $WRITE(7,15)
363         CALL GETNUM(A) $LFLAG=A(1)
364         WRITE(7,23)
365     23  FORMAT(*U1IN,U1FN*)
366         CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
367         CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
368         GO TO 100
369     117 WRITE(7,12)
370         CALL GETNUM(A) $IFLAG=A(1)
371         WRITE(7,15)
372         CALL GETNUM(A) $LFLAG=A(1)
373         WRITE(7,25)
374     25  FORMAT(*U3IN,U3FN*)
375         CALL GETNUM(A) $C(35)=A(1) $C(36)=A(2)
376         CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
377         GO TO 122
378     118 WRITE(7,12)
379         CALL GETNUM(A) $IFLAG=A(1)
380         WRITE(7,15)
381         CALL GETNUM(A) $LFLAG=A(1)
382         WRITE(7,23)
383         CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
384         CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
385         GO TO 123
386     119 WRITE(7,12)
387         CALL GETNUM(A) $IFLAG=A(1)
388         WRITE(7,15)
389         CALL GETNUM(A) $LFLAG=A(1)
390         WRITE(7,25)
391         CALL GETNUM(A) $C(35)=A(1) $C(36)=A(2)
392         CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
393         GO TO 123
394     120 WRITE(7,4)
395         CALL GETNUM(A) $X1=A(1) $X2=A(2)
396         CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
397         GO TO 123
398     630 STOP
399         END
400         SUBROUTINE CMPTLT(D,X1,X2,X3,T1,T2)
401         DIMENSION D(20)
402         A1=A2=A3=A4=B1=B2=B3=B4=0.
403         DA1=DA2=DA3=DA4=DB1=DB2=DB3=DB4=0.
404         U1=D(17)
405         U3=D(18)
406         IF(U1.EQ.0.)GO TO 1
```

```
      CALL TILT(D(17),X1,X2,X3,C(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(C(17),X1,X2,X3,C(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)/F)*((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 WRITE1
      WRITE(7,1)
    1 FORMAT(*SPECIFY INITIAL AND FINAL SLIP VALUES,*,/,
     1 *RIGHT-LATERAL STRIKE-SLIP IS <0.*,/,
     1 *DIP-SLIP >0. FOR 'X2>0. SIDE' DOWN*,/,
     1 *FORMAT=U1IN,U1FN,U3IN,U3FN*)
      RETURN
      END
      SUBROUTINE WRITE2(C)
      DIMENSION C(40),A(30)
      WRITE(7,3)
    3 FORMAT(*SPECIFY INITIAL AND FINAL COORDINATES*,/,
     1 *OF DIP-SLIP ZONE CORNERS AS INDICATED.*,/,
     1 *D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,D3X1FN*)
      CALL GETNUM(A) $C(1)=A(1) $C(2)=A(2) $C(4)=A(3) $C(5)=A(4)
      C(9)=A(5) $C(10)=A(6) $C(12)=A(7) $C(13)=A(8)
      C(3)=C(1)  $C(7)=C(5)  $C(6)=C(4)  $C(8)=C(2)
      C(11)=C(9)  $C(16)=C(10)  $C(14)=C(12)  $C(15)=C(13)
      RETURN
      END
      SUBROUTINE WRITE3(C)
      REAL A(30)
      DIMENSION C(40)
      WRITE(7,8)
    8 FORMAT(*SPECIFY INITIAL AND FINAL VALUES *,/,
     1 *OF STRIKE-SLIP ZONE COORDINATES AS INDICATED*,/,
     1 *C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN*)
```

```
      CALL GETNUM(A) $C(17)=A(1) $C(18)=A(2) $C(20)=A(3) $C(21)=A(4)
      C(25)=A(5) $C(26)=A(6) $C(28)=A(7) $C(29)=A(8)
      C(19)=C(17) $C(22)=C(20) $C(23)=C(21) $C(24)=C(18)
      C(27)=C(25) $C(30)=C(28) $C(31)=C(29) $C(32)=C(26)
      RETURN
      END
      SUBROUTINE WRITE4(TRIGGR,M1,M2,M3,KCHECK)
      REAL A(30)
      M1=100 $M2=100 $M3=100
      WRITE(7,150)
  150 FORMAT('DO YOU WISH THE 'TRIGGER' OPTION?',/,
     1 *'YES' OR 'NO '*)
      READ(7,151)TRIGGR
  151 FORMAT(A3)
      KCHECK=0
      IF(TRIGGR.EQ.3HNO )GO TO 160
      WRITE(7,157)
  157 FORMAT(*SPECIFY 'TRIGGER' PARAMETERS IN THE FORM*,/,
     1 *D(M1).LT.C(M2)*)
      CALL GETNUM(A) $M1=A(1) $M2=A(2)
      WRITE(7,158)
  158 FORMAT(*SPECIFY WHICH SLIP COMPONENT IS TO BE =0.*,/,
     1 *WHEN D(M1).GE.C(M2)*,/,
     1 *STRIKE-SLIP=S, DIP-SLIP=D*)
      READ(7,159)SLIP
  159 FORMAT(A1)
      IF(SLIP.EQ.1HS)M3=17
      IF(SLIP.EQ.1HD)M3=18
  160 CONTINUE
      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.36)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(80A1)
      END
```

MAXVGT

This program computes quasi-static tilts at the surface of a Maxwell or Voigt viscoelastic semi-infinite medium for a vertical rectangular dislocation loop and strike-slip displacement using the equations in Rosenman and Singh (1973). Input to the program consists of the initial and final position of the slip zone and magnitude of slip, station location, and a series of flags and constants. The tilt component amplitudes, total tilt amplitude, tilt azimuth, and the tilt amplitude decay curve for a static zone comprise the output.

Note: Currently, the dip-slip contribution to total tilt is set equal to zero. All material in the program relating to dip-slip should be ignored.

Access and Use:

This program is intended for use on the LBL 6600 B or C computer and the Tektronix (4010-1) terminal. It is stored on library 'MCHUGH' and may be accessed, after logging on to the B or C machine with approximatley 120 K of core, using the command

LOAD, MAXVGT, MCHUGH

The program is ready for immediate use; the next command should be ^RUN.

Input:

The geometry and notation used in this program is shown in figure 1. It is assumed that the positive X direction is northwest, positive Y is northeast, and that the angle between X and north (theta) is 45°. (To change theta, it is necessary to alter line 49 in the main program). The

computer will ask for the information it requires as follows:

a) 1 = Zone Expands

2 = Zone Contracts

Entering a 1 cause the tilts to be computed for a zone expanding from its initial position to its final position (as entered in part e below); a 2 causes the opposite to occur (i.e., the zone appears to 'collapse').

b) 1 = strike-slip component incremented exponentially
2 = dip-slip compnent incremented exponentially
3 = both-slip components incremeted exponentially
4 = neither slip component incremented exponentially

The dip-slip contribution to the total tilt is currently set equal to zero, therefore, all options relating to dip-slip displacement are inoperative. If a 1 or 3 is entered, the strike-slip magnitude is incremeted in a $(1-e^{-t/\tau_s})$ fashion where the time constant for strike-slip displacement (τ_s) is set internally such that the slip has reached 99.998% of its final value at the end of the computations.

c) The computer next requires that the parameters in the 'TRIGGER' option be specified. This and the following steps are described in detail in the documentation for program XPND/XPNDØ1.

d) Specify slip values.

e) Specify zone coordinates.

f) Specify station coordinates.

g) enter Maxwell and Voigt time constants.

Specify the time constants used in the tilt equations for the Maxwell and Voigt solids (Rosenman and Singh, 1973).

Note: for an elastic material, the Maxwell time constant. τ_2, approaches

∞ and the Voigt time constant, τ_1, approaches zero. Because of limitations on the exponent that may be used in the exponential function, it is suggested that 'TIME' in line 380 be set equal to some number greater than 6 but not more than 10, and that 'TIME' in line 398 be set equal to zero.

h) the angle theta is written; no action required.

Output:

The output and responses required are nearly identical to those described in the Documentation for programs XPND/XPNDØ1 and will only be summarized here. The output for the Maxwell model occurs first, followed by the output for the Voigt model. The format of the solutions for the Voigt model is identical to the Maxwell model and will not be discussed separately. All plots are labelled automatically so that no confusion between the two sets of solutions can occur.

a) The minimum and maximum values of the two tilt components, the total tilt amplitude, and tilt azimuth are listed with a description of their units.

b) Enter the appropriate number to restart or continue the program.

c) Enter plot title.

d) and e) Set scales

f) Option to skip plot

g) After plot is displayed, enter a blank.

h) Restart or continue program.

i, j, k) Set scales and display next component.

Once the two tilt components have been displayed, the computer will ask for instructions regarding the display of the tilt as a function of time for a static zone. The zone index (corresponding to a particular static zone, or instant of time) must be between IJKL and NEND (those two variables are equivalent to a time $t_{initial}$ and t_{final}; that is, the zone starts

growing at time = IJKL and ends at time = NEND). Because of the visco-elastic response of the medium, the tilt amplitude for a static zone will decay with time. The components THETA1 and THETA2 are respectively the X and Y components of the tilt (i.e., $\frac{\partial W}{\partial X}$ and $\frac{\partial W}{\partial Y}$ in Rosenman and Singh, 1973) for a particular source-station geometry. To skip displaying THETA1 and THETA2, enter a 999; otherwise the minimum and maximum values of the tilt will be written. The graphs will be displayed if an N or blank is entered after the SKIP? response. The tilt amplitude and azimuth follow the THETA1 and THETA2 displays. Once the azimuth has been plotted, the same sequence for the output is followed for the Voigt model calculations.

After the final Voigt plot, the program may be restarted with all new values, only new slip values, or new instrument coordinates (as in programs XPND/XPNDØ1). If any option between 1 and 6 is selected, the computer will request the information it needs. If a 7 is entered, the program will stop.

References

Rosenman, M., and S.J. Singh - Quasi-static strains and tilts due to faulting in a viscoelastic half-space - Bull. Seismol. Soc. Amer., 63, 1737-1752, 1973.

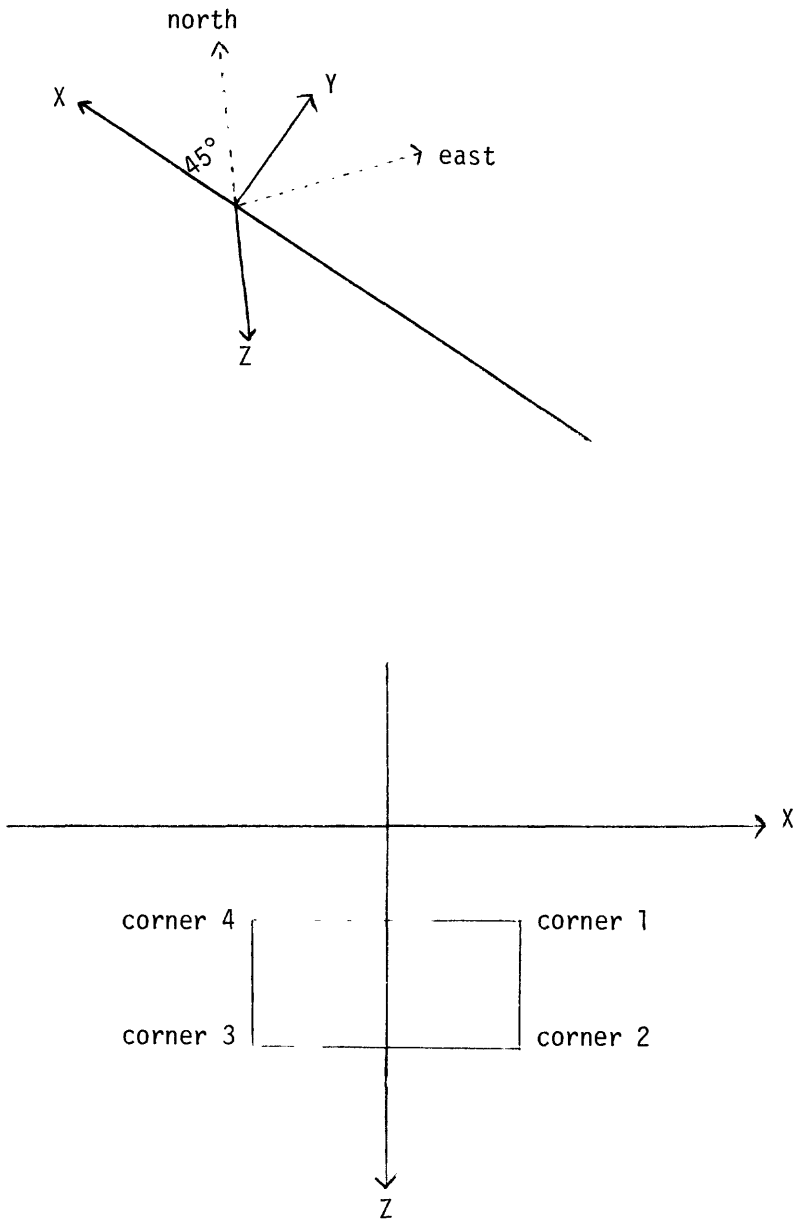


Figure 1

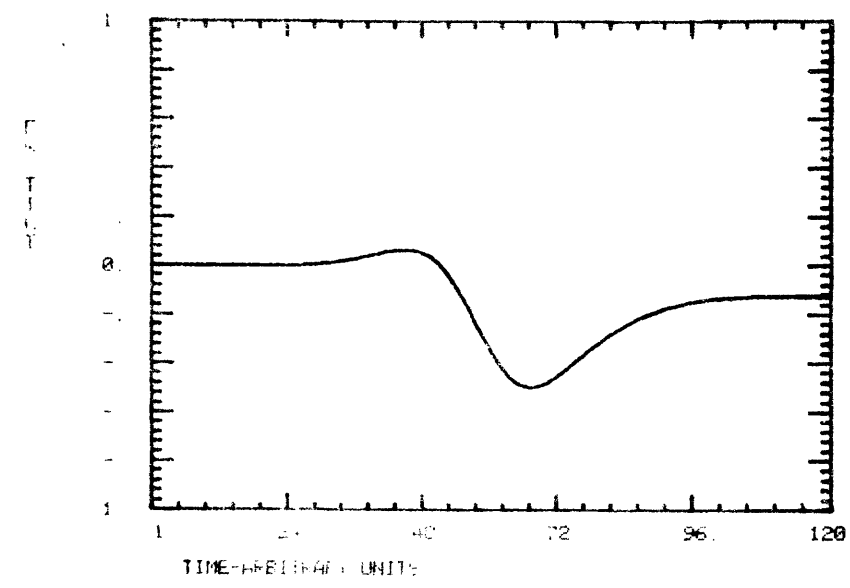
geometry and notation used in program MAXVGT

EXAMPLE FROM MAXVGT

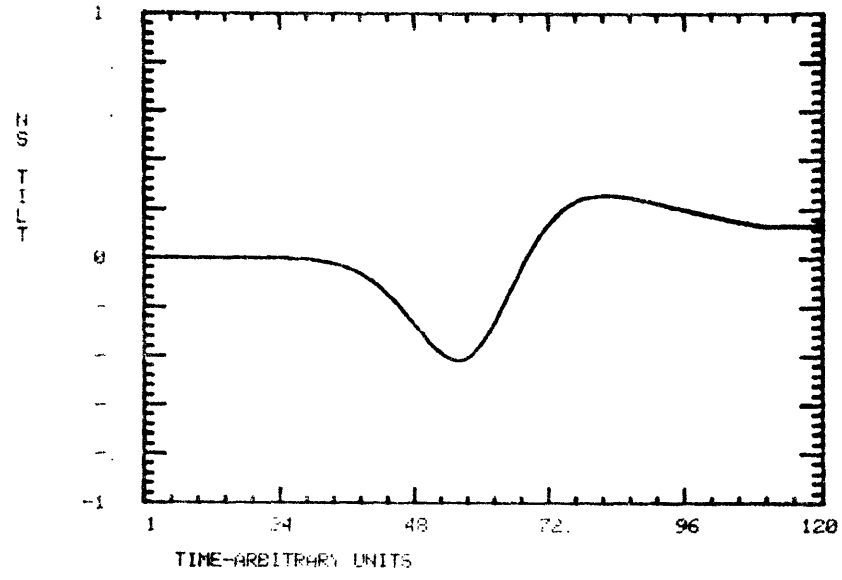
```
1=ZONE EXPANDS
2=ZONE CONTRACTS
1!
1=STRIKE-SLIP COMPONENT INCREMENTED EXPONENTIALLY
2=DIP SLIP COMPONENT INCREMENTED EXPONENTIALLY
3=BOTH SLIP COMPONENTS INCREMENTED EXPONENTIALLY
4=NEITHER SLIP COMPONENT INCREMENTED EXPONENTIALLY
1
DO YOU WISH THE TRIGGER OPTION
'YES' OR 'NO'
NO!
SPECIFY INITIAL AND FINAL SLIP VALUES
RIGHT-LATERAL STRIKE-SLIP IS +M
DIP-SLIP >0  FOR 'X2>0' SIDE DOWN
FORMAT=U1IN,U1FN,U2IN U2FN
0 -5 0 0!
SPECIFY INITIAL AND FINAL VALUES
OF STRIKE-SLIP ZONE COORDINATES AS INDICATED
C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN
0 0 0 0 2 0 1 0!
COORDINATES OF STATION (X1,X2)
1 .37!
ENTER MAXWELL AND VOIGT TIME CONSTANTS
.1  .1
```



```
!
OFFSET=ANGLE BETWEEN STRIKE OF FAULT
AND NORTH=    45.000 DEGREES
MIN/MAX VALUES OF EW COMPONENT  -4.982E-01   6.435E-02
MIN/MAX VALUES OF NS COMPONENT  -4.146E-01   2.589E-01
MIN/MAX VALUES OF AMPLITUDE      0           5.016E-01
MIN/MAX VALUES OF AZIMUTH        0.        322.899
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=ME-START, 1=CONTINUE
1!
WRITE PLOT TITLE, 80 CHARACTERS
LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE!
SET HORIZONTAL SCALE? Y OR N<=BLANK)
 !
SET VERTICAL SCALE? Y OR N<=BLANK)
Y!
MIN/MAX Y VALUES
-1  1!
SKIP PLOT OF  EW TILT  --- MAXWELL MODEL       ?
```

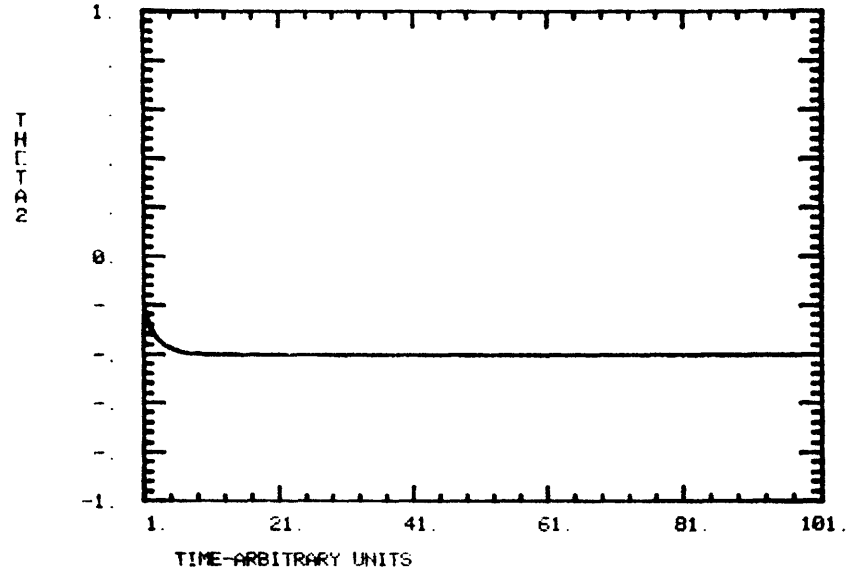

1
0.
1
1
40
72
96.
120.

LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
MAXWELL MODEL



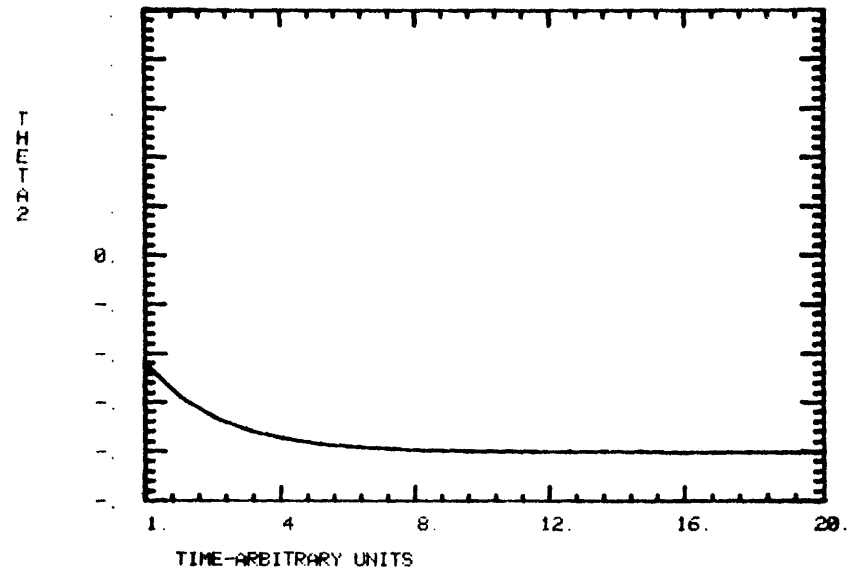
0=RE [illegible]RT, 1=MAIN PLOT, 2=[illegible]TURN
2
ENTER ZONE INDEX CAUTION--VALUE MUST BE
BETWEEN 10 AND 110 2=CONTINUE
SS
MIN,MAX VALUES OF THETA2 [illegible]= 4.011E-01 -2.189E-01
MIN,MAX VALUES OF THETA1 [illegible]= -1.973E-01 -9.019E-02
SET HORIZONTAL SCALE? Y OR N =PLANK)
Y
SET VERTICAL SCALE? Y OR N =BLANK)
Y
MIN,MAX Y VALUES
 1 1
 4 UP PLOT OF THETA2 ---- MAXWELL MODEL

LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
MAXWELL MODEL



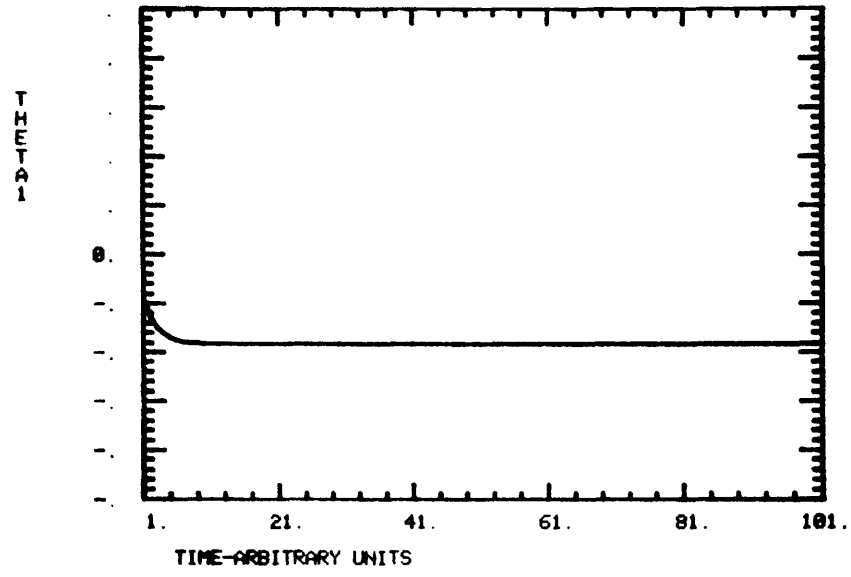
1 :NEW PLOT, 2=RETURN
:1
SET HORIZONTAL SCALE? Y OR N(=BLANK)
:Y
MIN,MAX X VALUES
1 201
SET VERTICAL SCALE? Y OR N(=BLANK)
:Y
MIN,MAX Y VALUES
-.5 .51
SKIP PLOT OF THETA2 --- MAXWELL MODEL ?

LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
MAXWELL MODEL



1 =NEW PLOT, 2=RETURN
?
1 =T HORIZONTAL SCALE? Y OR N=BLANK)
?
1 =T VERTICAL SCALE? Y OR N(=BLANK)
?
MIN MAX Y VALUES
-.5
1 =T PLOT OF THETA1 ---- MAXWELL MODEL
?

LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
MAXWELL MODEL

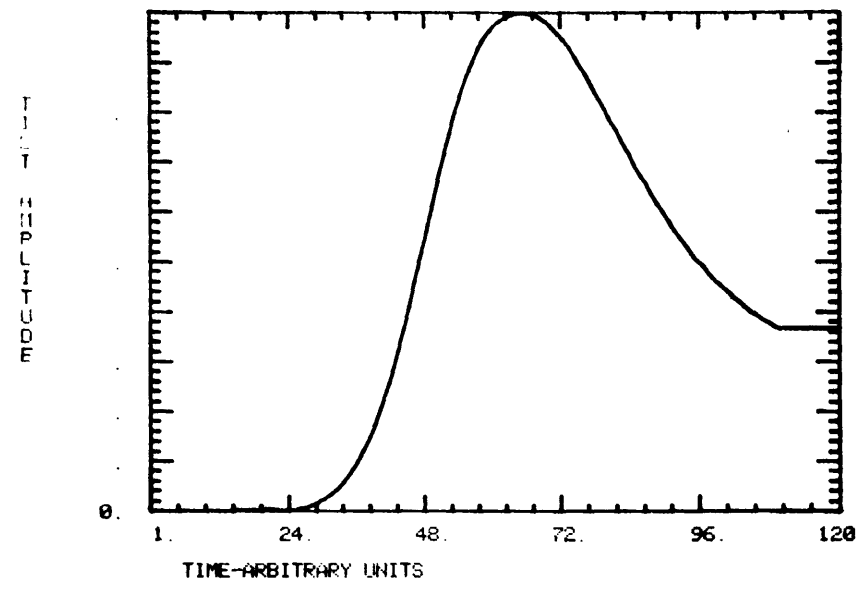


0=ME-START, 1=NEW PLOT, 2=RETURN
2
ENTER ZONE INDEX, CAUTION--VALUE MUST BE
BETWEEN 10 AND 110, 999=CONTINUE
999
SET HORIZONTAL SCALE? Y OR N(=BLANK)

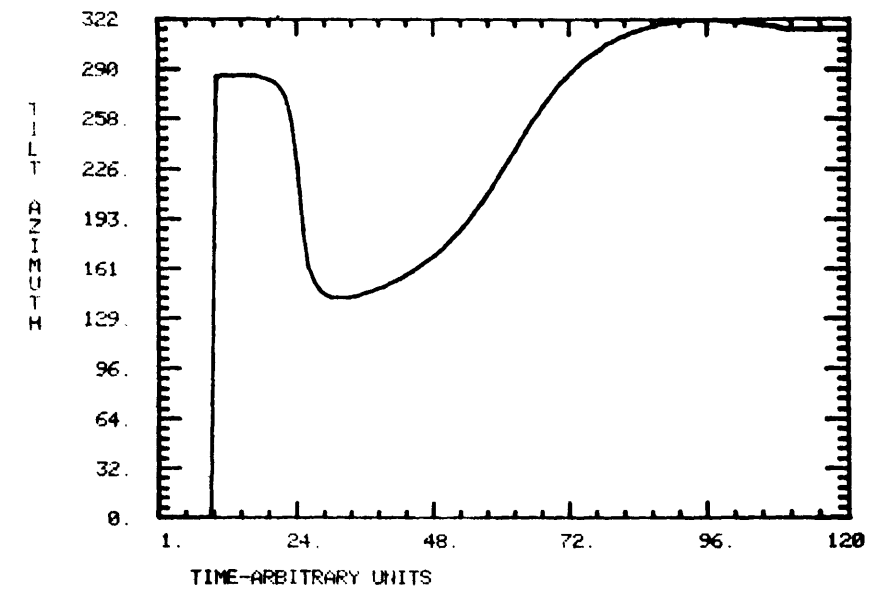
SET VERTICAL SCALE? Y OR N(=BLANK)

SKIP PLOT OF TILT AMPLITUDE ---MAXWELL MODEL ?

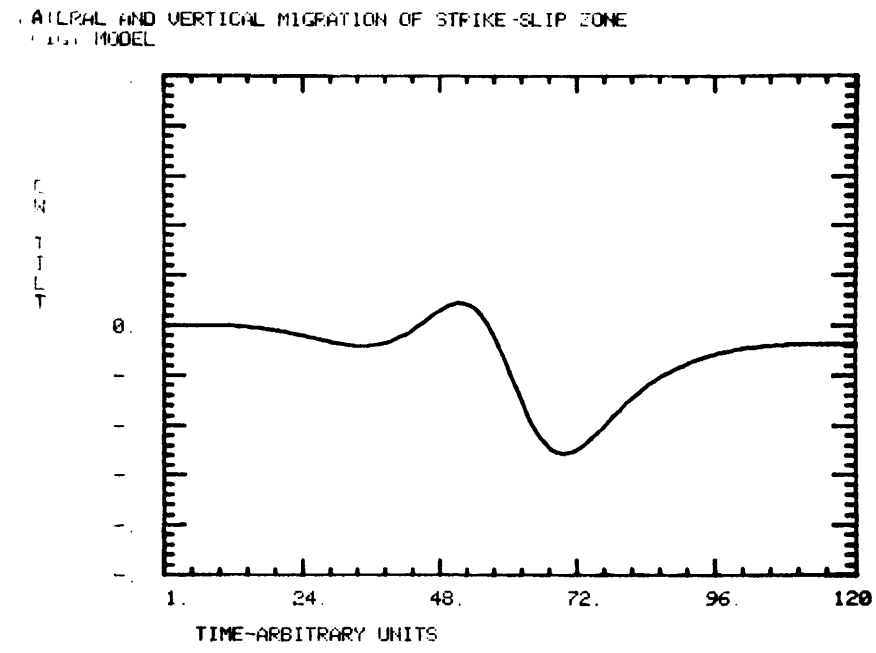
LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
MAXWELL MODEL



LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
MAXWELL MODEL

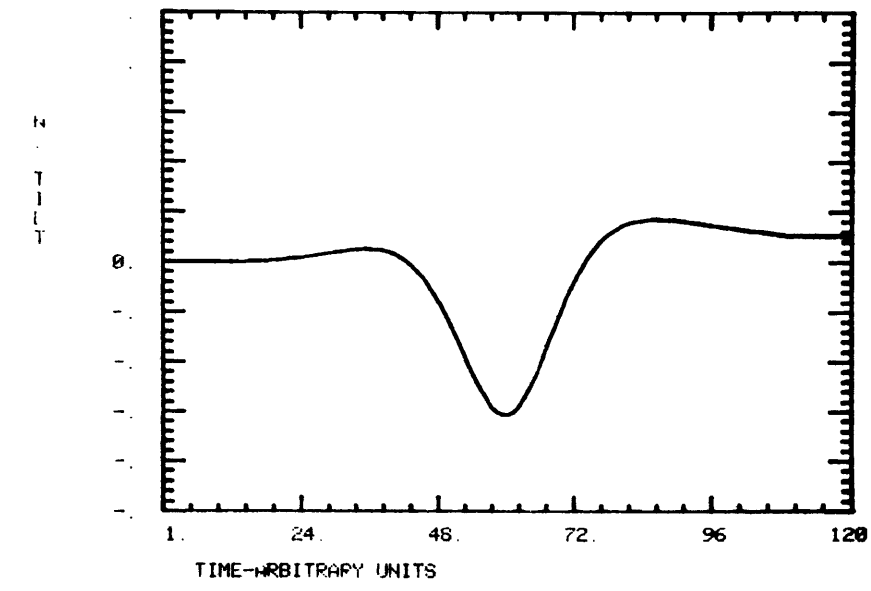


```
0=RE START, 1=NEW PLOT, 2=RETURN
2
MIN/MAX VALUES OF EW COMPONENT   -2.555E-01   4.500E-02
MIN/MAX VALUES OF NS COMPONENT    3.058E 01   8.534E-02
MIN/MAX VALUES OF AMPLITUDE       0           3.245E-01
MIN/MAX VALUES OF AZIMUTH         0         325.434
ENTER TILT AMPLITUDE MAX IN MICRORADIANS
AZIMUTH IN DEGREES
IF FOLLOWING ARE PLOTS OF THE EW AND NS
COMPONENTS OF TILT, AND THE TILT AMPLITUDE
AND AZIMUTH (MEASURED CLOCKWISE FROM NORTH).
RE START, 1=CONTINUE
1
ENTER PLOT TITLE, 80 CHARACTERS
LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE!
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  EW TILT  --- VOIGT MODEL        ?
```

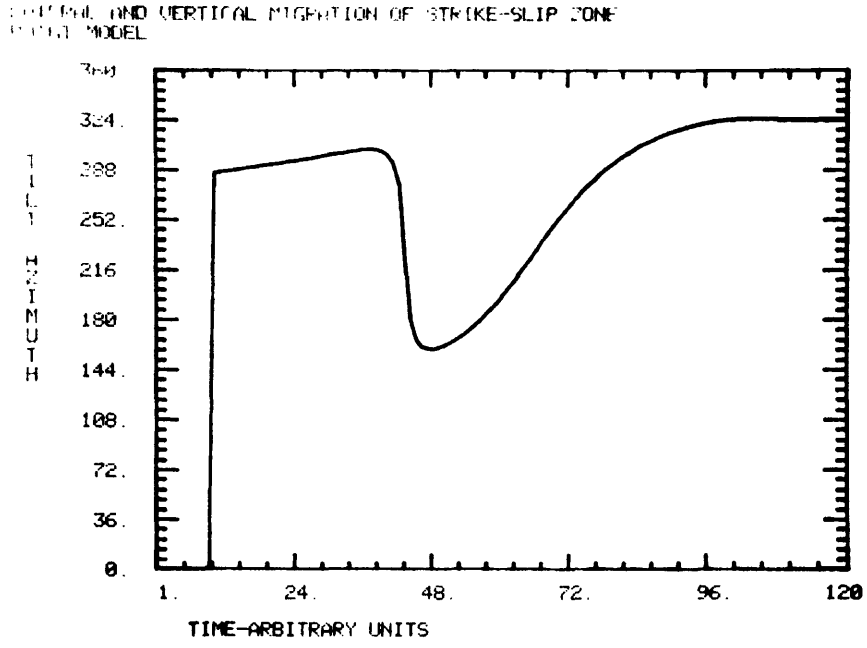



```
1=NEW PLOT, 2=RETURN
2
SET HORIZONTAL SCALE? Y OR N(=BLANK)

SET VERTICAL SCALE? Y OR N(=BLANK)
Y
MIN/MAX Y VALUES
-.5  .5
SKIP PLOT OF  NS TILT  --- VOIGT MODEL          ?
```


LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
VOIGT MODEL



```
0=PC START, 1=NEW PLOT, 2=RETURN
2
CRITICAL ZONE INDEX, CAUTION =VALUE MUST BE
BETWEEN   10 AND  110, 999=CONTINUE
999
SET HORIZONTAL SCALE? Y OR N(=BLANK)
N
SET VERTICAL SCALE? Y OR N(=BLANK)
N
SKIP PLOT OF TILT AMPLITUDE      ---VOIGT MODEL      ?
Y
1=NEW PLOT, 2=RETURN
2
SET HORIZONTAL SCALE? Y OR N(=BLANK)
N
SET VERTICAL SCALE? Y OR N(=BLANK)
Y
MIN/MAX Y VALUES
0 360
SKIP PLOT OF TILT AZIMUTH        ---VOIGT MODEL      ?
```

```
0=RE-START, 1=NEW PLOT, 2=RETURN
2
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
?
OK - ^EDIT
```

LISTING OF MAXVGT

```
24 MAY 76    10.53.15        MCHUGH .MAXVGT

  1  DELETE(LGO,OUTPUT,MAXVGT)
  2  MAXVGT.
  3  CXIT.
  4  LIBCOPY(GRAPHIC,TXLGO/RR,TXLGO)
  5  LIBCOPY(JDRAT,NPLGO/RR,NPLGO)
  6  DELETE(LGO,OUTPUT,MAXVGT)
  7  RUN76(S)
  8  LINK(F=LGO,F=TXLGO,F=NPLGO,P=MAXVGT)
  9  MAXVGT.
 10  FIN.
 11  EOR
 12        PROGRAM MAXVGT(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
 13        COMMON/TVPCOL/TVPUL(8)
 14        COMMON/TVTLNE/ITUNE(35)
 15        COMMON/JPLOT/XLT,XRT,YLO,YUP,MAJX,MAJY,KX(2),KY(2),
 16       1LTITL(8),LU,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
 17        DIMENSION IFET(8)
 18        DIMENSION TEW(120),TNS(120),TAMP(120),TAZM(120),T(120)
 19        DIMENSION B(20),C(40),A(30)
 20        DIMENSION THETA1(120,101),THETA2(120,101),TT(101)
 21        DATA KTIME,PTIME,KCHECK,M1,M2,M3/1.,1.,1.,1.,1.,1./
 22        CALL FET(5LTAPE7,IFET,8)
 23        IFET(2)=IFET(2).OR.0000 0010 0000 0000 0000B
 24        IFET(8)=IFET(8).OR.4000 0000 0000 0000 0000B
 25        CALL FET(5LTAPE7,IFET,-8)
 26     20 CONTINUE
 27        MSTART=3 $WRITE(7,12)
 28     12 FORMAT(*1=ZONE EXPANDS*,/,
 29       1 *2=ZONE CONTRACTS*)
 30        CALL GETNUM (A) $IFLAG=A(1) $WRITE(7,15)
 31     15 FORMAT(*1=STRIKE-SLIP COMPONENT INCREMENTED EXPONENTIALLY*,/,
 32       1 *2=DIP-SLIP COMPONENT INCREMENTED EXPONENTIALLY*,/,
 33       1 *3=BOTH SLIP COMPONENTS INCREMENTED EXPONENTIALLY*,/,
 34       1 *4=NEITHER SLIP COMPONENT INCREMENTED EXPONENTIALLY*)
 35        CALL GETNUM (A) ILFLAG=A(1)
 36        CALL WRITE1(TRIGGP,M1,M2,M3,KCHECK) $CALL WRITE1
 37        CALL GETNUM(A)$C(33)=-A(1)$C(34)=-A(2)$C(35)=A(3)$C(36)=A(4)
 38        DO 13 I=1,32
 39     13 C(I)=0.
 40    122 IF(C(35) .EQ. 0. .AND. C(36) .EQ. 0.) GO TO 100
 41        CALL WRITE2(C)
 42    100 IF(MSTART .EQ. 3) GO TO 123
 43        IF(C(33) .EQ. 0. .AND. C(34) .EQ. 0.) GO TO 101
 44        CALL WRITE3(C)
 45    101 IF(MSTART .EQ. 2) GO TO 123
 46        WRITE(7,4)
 47      4 FORMAT(*COORDINATES OF STATION (X1,X2)*)
 48        CALL GETNUM (A) $X1=A(1) $X2=A(2)
 49        WRITE(7,1234)           $THETA=45.
 50   1234 FORMAT(*ENTER MAXWELL AND VOIGT TIME CONSTANTS*)
 51        CALL GETNUM(A)$B(19)=A(1)$B(20)=A(2)
 52    123 WRITE(7,7)    THETA
 53      7 FORMAT(*THETA=ANGLE BETWEEN STRIKE OF FAULT*,/,
 54       1 *AND NORTH=*,F10.3,* DEGREES*)
 55        X3=0.  $NUM=120  IIJKL=(.075*NUM)+1.
 56        IF(C(35).EQ.0..AND.C(36).EQ.0.)GO TO 151
 57        IF(C(33).EQ.0..AND.C(34).EQ.0.)GO TO 152
 58        GO TO 153
```

```
 59     151 DO 154 I=1,16
 60     154 C(I)=0.
 61         GO TO 153
 62     152 DO 155 I=16,32
 63     155 C(I)=0.
 64     153 CONTINUE
 65         NEND=NUM-IJKL  $TAU=-(NEND-IJKL)/6.   $INDEX=NEND-IJKL+1
 66         COUNTR=1.
 67      22 DO 103 I=IJKL,NEND
 68         DO 104 K=1,16
 69         IF(K.LE.8)L1=K   $IF(K.LE.8)L2=K+8
 70         IF(K.GT.8.AND.K.LE.16)L1=K+8
 71         IF(K.GT.8.AND.K.LE.16)L2=K+16
 72     104 D(K)=(((C(L2)-C(L1))/(NEND-IJKL))*(I-IJKL))+C(L1)
 73         IF(LFLAG.EQ.1.OR.LFLAG.EQ.3)D(17)=((1.-EXP((I-IJKL)/TAU))*(C(34)-C
 74        1(33)))+C(33)
 75         IF(LFLAG.EQ.2.OR.LFLAG.EQ.3)D(18)=((1.-EXP((I-IJKL)/TAU))*(C(36)-C
 76        1(35)))+C(35)
 77         IF(LFLAG.EQ.1)D(18)=(((C(36)-C(35))/(NEND-IJKL))*(I-IJKL))+C(35)
 78         IF(LFLAG.EQ.2)D(17)=(((C(34)-C(33))/(NEND-IJKL))*(I-IJKL))+C(33)
 79         IF(LFLAG.EQ.4)D(17)=(((C(34)-C(33))/(NEND-IJKL))*(I-IJKL))+C(33)
 80         IF(LFLAG.EQ.4)D(18)=(((C(36)-C(35))/(NEND-IJKL))*(I-IJKL))+C(35)
 81         IF(TRIGGR.NE.3HYES)GO TO 156
 82         IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2))GO TO 114
 83         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2))KCHECK=KCHECK+1
 84         IF(KCHECK.EQ.1)KTIME=NEND-I
 85         IF(KCHECK.EQ.1)RTIME=I $IF(M3.EQ.18)GOTO 180
 86         DO 181 K=9,16
 87         L1=K+8  $L2=K+16
 88     181 D(K)=(((C(L2)-C(L1))/(KTIME))*(I-RTIME))+C(L1)
 89         GO TO 114
 90     180 DO 182 K=1,8
 91         L1=K   $L2=K+8
 92     182 D(K)=(((C(L2)-C(L1))/(KTIME))*(I-RTIME))+C(L1)
 93     114 IF(TRIGGR.EQ. 3HYES.AND.D(M1).LT.C(M2))D(M3)=0.
 94         IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.M3.EQ.17)GO TO 9
 95         IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.M3.EQ.18)GO TO 10
 96         GO TO 11
 97       9 DO 16 M=9,16
 98      16 D(M)=C(M+8) $GOTO11
 99      10 DO 18 M=1,8
100      18 D(M)=C(M)
101      11 IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.C(M3).EQ.0.)GO TO 156
102         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2) .AND.LFLAG.EQ.1
103        1 .AND.M3.EQ.17)D(M3)=((1.-EXP( (I-RTIME)/
104        1 (-KTIME/6.)))*(C(34)-C(33)))+C(33)
105         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.3
106        1 .AND. M3.EQ.17)D(M3)=((1.-EXP((I-RTIME)/(-KTIME/6.)))*(C(34)
107        1-C(33)))+C(33)
108         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.2
109        1 .AND.M3.EQ.18)D(M3)=((1.-EXP((I-RTIME)/
110        1 (-KTIME/6.)))*(C(36)-C(35)))+C(35)
111         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.3.AND.
112        1 M3.EQ.18)D(M3)=((1.-EXP((I-RTIME)/(-KTIME/6.)))*(C(36)-C(35)))+C(
113        135)
114         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.1
115        1 .AND.M3.EQ.18)D(M3)=(((C(36)-C(35))/
116        1 (KTIME))*(I-RTIME))+C(35)
```

```
117         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.4.AND.M3.EQ.18)
118        1 D(M3)=(((C(36)-C(35))/(KTIME))*(I-RTIME))+C(35)
119         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.2
120        1 .AND.M3.EQ.17) D(M3)=(((C(34)-C(33))/
121        1 (KTIME))*(I-RTIME))+C(33)
122         IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.M3.EQ.17.AND.LFLAG.EQ.4)
123        1 D(M3)=(((C(34)-C(33))/(KTIME))*(I-RTIME))+C(33)
124     156 CALL CMPTLT(D,X1,X2,X3,COUNTR,I,INDEX,THETA1,THETA2)
125         T(I)=I
126     103 CONTINUE
127         DO 2 J=IJKL,NEND
128         SUM11=SUM12=SUM21=SUM22=0.
129         DO 3 I=IJKL,J
130         A=THETA1(J-I+IJKL,I-IJKL+1)
131         B=THETA2(J-I+IJKL,I-IJKL+1)
132         SUM11=A+SUM11
133       3 SUM21=B+SUM21
134         LM=J-1
135         IF(LM.LT.IJKL)GOTO35
136         DO 5 I=IJKL,LM
137         A=THETA1(J-I+IJKL-1,I-IJKL+1)
138         B=THETA2(J-I+IJKL-1,I-IJKL+1)
139         SUM12=A+SUM12
140       5 SUM22=B+SUM22
141      35 TNS(J)=SUM11-SUM12
142       2 TEW(J)=SUM21-SUM22
143         K7=IJKL-1
144         DO 106 I=1,K7
145         TEW(I)=TEW(IJKL)
146         T(I)=I
147     106 TNS(I)=TNS(IJKL)
148         K8=NEND+1
149         DO 107 I=K8,NUM
150         T(I)=I
151         TEW(I)=TEW(NEND)
152     107 TNS(I)=TNS(NEND)
153         IF(IFLAG.EQ. 2)GO TO 108
154         A=TEW(1)     $B=TNS(1)
155         DO 109 I=1,NUM
156         TEW(I)=TEW(I)-A
157     109 TNS(I)=TNS(I)-B
158         GO TO 110
159     108 DO 111 I=1,NUM
160         TEW(I)=TEW(NUM)-TEW(I)
161     111 TNS(I)=TNS(NUM)-TNS(I)
162     110 CT=COS(THETA*.01745)
163         ST=SIN(THETA *.01745)
164         DO 21 I=1,NUM
165         B=TEW(I)  $A=TNS(I)
166         TNS(I)=A*CT+B*ST
167         TEW(I)=-A*ST+B*CT
168         TAMP(I)=SQRT((TEW(I)**2)+(TNS(I)**2))
169         IF(TNS(I).EQ.0.)TNS(I)=1.E-20
170         TAZM(I)=(ATAN(TEW(I)/TNS(I)))*(180./3.1415926)
171         IF(TNS(I).LT.0.)TAZM(I)=TAZM(1)+180.
172         IF(TAZM(I).LT.0.)TAZM(I)=TAZM(I)+360.
173      21 IF(TAZM(I).GT.360.)TAZM(I)=TAZM(I)-360.
174         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 160 I=1,NUM
      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)
  160 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 *0=RE-START, 1=CONTINUE*)
      CALL GETNUM(A) $IKFLAG=A(1)
      IF(IKFLAG.EQ. 0) GO TO 113
      LU=7  $LNLGX=1  $LNLGY=1 $NCLX=2  $NCLY=2
      WRITE(7,199)
  199 FORMAT(*WRITE PLOT TITLE, 80 CHARACTERS*)
      READ(7,200)(LTITL(JM),JM=1,8)
  200 FORMAT(8A10)
  202  IF(COUNTR.EQ.1)LTITL2(1)=10HMAXWELL MC
      IF(COUNTR.EQ.1)LTITL2(2)=10HDEL
      IF(COUNTR.EQ.2)LTITL2(1)=10HVOIGT MODE
      IF(COUNTR.EQ.2)LTITL2(2)=10HL
      DO 8 MM=3,8
    8 LTITL2(MM)=10H
      KX(1)=10HTIME-ARBIT
      KX(2)=10HRARY UNITS
      KY(1)=10H EW TILT
      KY(2)=10H            $MAJX=5 $MAJY=10  $XLT=T(1)
      XRT=T(NUM)  $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(A)  $XLT=A(1)   $XRT=A(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(A)  $YLO=A(1)   $YUP=A(2)
   74  AAA=YUP $IF(YLO.EQ.AAA)YUP=YUP+1.$IF(YLO.EQ.AAA)YLO=YLO-1.
      WRITE(7,1)KY(1),LTITL2(1),LTITL2(2)
    1 FORMAT(*SKIP PLOT OF *,A10,*--- *,2A10,*?*)
      READ(7,66)IJVAR
```

```
233      66 FORMAT(A1)
234         IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TEW,T,1,NUM)
235         WRITE(7,201)
236     201 FORMAT(*1=NEW PLOT, 2=RETURN*)
237         CALL GETNUM(A)$IRS=A(1)$IF(IRS.EQ.1)GOTO202
238     205 KY(1)=10H NS TILT      $KY(2)=10H
239         YLO=TNSMIN $YUP=TNSMAX $XLT=T(1)$XRT=T(NUM)
240         WRITE(7,70)$READ(7,66)CHARAC
241         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO203
242         WRITE(7,72)$CALL GETNUM(A)$XLT=A(1)$XRT=A(2)
243     203 WRITE(7,73)$READ(7,66)CHARAC
244         IF(CHARAC.EQ.1HN .OR.CHARAC.EQ.1H )GOTO204
245         WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
246     204 AAA=YUP $IF(YLO.EQ.AAA)YUP=YUP+1.$IF(YLO.EQ.AAA)YLO=YLO-1.
247         WRITE(7,1)KY(1),LTITL2(1),LTITL2(2) $READ(7,66)IJVAR
248         IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TNS,T,1,NUM)
249         WRITE(7,19)
250      19 FORMAT(*0=RE-START, 1=NEW PLOT, 2=RETURN*)
251         CALL GETNUM(A)$IF(A(1).EQ.0.)GOTO113$IF(A(1).EQ.1)GOTO205
252      32 KY(1)=10H THETA2      $KY(2)=10H
253     206 WRITE(7,30)IJKL,NEND
254      30 FORMAT(*ENTER ZONE INDEX, CAUTION--VALUE MUST BE *,/,
255        1     *BETWEEN *,I4,* AND *,I4,*, 999=CONTINUE*)
256         CALL GETNUM(A) $I=A(1) $IF(I.EQ.999)GOTO31  $IKRAK=I
257         IF(I.LT.IJKL.OR.I.GT.NEND)GOTO206
258         DO 24 K=1,INDEX
259         TT(K)=K$TNS(K)=THETA1(I,K)
260      24 TEW(K)=THETA2(I,K)
261         TNSMID=TNS(1) $TEWMID=TEW(1)
262         TEWMIN=TEWMAX=TEWMID$TNSMIN=TNSMAX=TNSMID
263         DO 33 I=1,INDEX
264         IF(TNS(I).LT.TNSMID)TNSMIN=TNS(I)
265         IF(TEW(I).LT.TEWMID)TEWMIN=TEW(I)
266         IF(TNS(I).GT.TNSMID)TNSMAX=TNS(I)
267      33 IF(TEW(I).GT.TEWMID)TEWMAX=TEW(I)
268         WRITE(7,34)IKRAK,TEWMIN,TEWMAX,IKRAK,TNSMIN,TNSMAX
269      34 FORMAT(*MIN/MAX VALUES OF THETA2(*,I3,*)=*,5X,E10.3,2X,E10.3,/,
270        1 *MIN/MAX VALUES OF THETA1(*,I3,*)=*,5X,E10.3,2X,E10.3)
271     216 XLT=TT(1) $XRT=TT(INDEX)$YLO=TEWMIN$YUP=TEWMAX
272         WRITE(7,70)$READ(7,66)CHARAC
273         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO207
274         WRITE(7,72) $CALL GETNUM(A)$XLT=A(1)$XRT=A(2)
275      207 WRITE(7,73)$READ(7,66)CHARAC
276         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO208
277         WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
278     208 AAA=YUP$IF(YLO.EQ.AAA)YUP=YUP+1.$IF(YLO.EQ.AAA)YLO=YLO-1.
279         WRITE(7,1)KY(1),LTITL2(1),LTITL2(2)  $READ(7,66)IJVAR
280         IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TEW,TT,1,INDEX)
281         WRITE(7,201)$CALL GETNUM(A)$IF(A(1).EQ.1)GOTO216
282      217 KY(1)=10HTHETA1       $XLT=TT(1)$XRT=TT(INDEX)
283         WRITE(7,70) $READ(7,66)CHARAC
284         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO209
285         WRITE(7,72)$CALL GETNUM(A) $XLT=A(1) $XRT=A(2)
286      209 WRITE(7,73)$READ(7,66)CHARAC
287         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO210
288         WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
289      210 AAA=YUP$IF(YLO.EQ.AAA)YUP=YUP+1.$IF(YLO.EQ.AAA)YLO=YLO-1.
290         WRITE(7,1)KY(1),LTITL2(1),LTITL2(2)  $READ(7,66)IJVAR
```

```
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TNS,TT,1,INDEX)
      WRITE(7,19)$CALL GETNUM(A)$IF(A(1).EQ.0.)GOTO113
      IF(A(1).EQ.1.)GOTO217
      GOTO32
   31 CONTINUE
  218 KY(1)=10HTILT AMPLI   $KY(2)=10HTUDE
      YLO=TAMPMN $YUP=TAMPMX $XLT=T(1) $XRT=T(NUM)
      WRITE(7,70) $READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO211
      WRITE(7,72)$CALL GETNUM(A)$XLT=A(1)$XRT=A(2)
  211 WRITE(7,73)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO212
      WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
  212 AAA=YUP$IF(YLO.EQ.AAA)YUP=YUP+1.$IF(YLO.EQ.AAA)YLO=YLO-1.
      WRITE(7,213)KY(1),KY(2),LTITL2(1),LTITL2(2)$READ(7,66)IJVAR
  213 FORMAT(*SKIP PLOT OF *,2A10,*---*,2A10,*?*)
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAMP,T,1,NUM)
      WRITE(7,201)$CALL GETNUM(A)$IF(A(1).EQ.1.)GOTO218
  220 KY(1)=10HTILT AZIMU   $KY(2)=10HTH          $YLO=TAZMMN
      YUP=TAZMMX $XLT=T(1) $XRT=T(NUM)
      WRITE(7,70)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO214
      WRITE(7,72)$CALL GETNUM(A)$XLT=A(1)$XRT=A(2)
  214 WRITE(7,73)$READ(7,66)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO215
      WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
  215 AAA=YUP$IF(YLO.EQ.AAA)YUP=YUP+1.$IF(YLO.EQ.AAA)YLO=YLO-1.
      WRITE(7,213)KY(1),KY(2),LTITL2(1),LTITL2(2)$READ(7,66)IJVAR
      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAZM,T,1,NUM)
      WRITE(7,19)$CALL GETNUM(A)
      IF(A(1).EQ.0.)GOTO113$IF(A(1).EQ.1.)GOTO220
  113 COUNTR=COUNTR+1
      IF(COUNTR.EQ.2)GOTO22
      WRITE(7,115)
  115 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*)
      CALL GETNUM(A) $MSTART=A(1)
      IF(MSTART.EQ.2)GOTO116 $IF(MSTART.EQ.3)GOTO117
      IF(MSTART.EQ.4)GOTO118 $IF(MSTART.EQ.5)GOTO119
      IF(MSTART.EQ.6)GOTO120 $IF(MSTART.EQ.7)GOTO630
      GO TO 20
  116 WRITE (7,12)
      CALL GETNUM(A) $IFLAG=A(1)  $WRITE(7,15)
      CALL GETNUM(A) $LFLAG=A(1)
      WRITE(7,23)
   23 FORMAT(*U1IN,U1FN*)
      CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
      CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
      GO TO 100
  117 WRITE(7,12)
      CALL GETNUM(A) $IFLAG=A(1)
      WRITE(7,15)
      CALL GETNUM(A) $LFLAG=A(1)
```

```
      WRITE(7,25)
   25 FORMAT(*U3IN,U3FN*)
      CALL GETNUM(A) $C(35)=A(1) $C(36)=A(2)
      CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
      GO TO 122
  118 WRITE(7,12)
      CALL GETNUM(A) $IFLAG=A(1)
      WRITE(7,15)
      CALL GETNUM(A) $LFLAG=A(1)
      WRITE(7,23)
      CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
      CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
      GO TO 123
  119 WRITE(7,12)
      CALL GETNUM(A) $IFLAG=A(1)
      WRITE(7,15)
      CALL GETNUM(A) $LFLAG=A(1)
      WRITE(7,25)
      CALL GETNUM(A) $C(35)=A(1) $C(36)=A(2)
      CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
      GO TO 123
  120 WRITE(7,4)
      CALL GETNUM(A) $X1=A(1) $X2=A(2)
      CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
      GO TO 123
  630 STOP
      END
      SUBROUTINE CMPTLT(D,X1,X2,X3,COUNTR,I,INDEX),THETA1,THETA2)
      DIMENSION D(20),THETA1(120,101),THETA2(120,101)
      TAUM=D(19) $TAUV=D(20)
      IF(COUNTR.EQ.2)GOTO3
      DO 4 K=1,INDEX
      TIME=((K-1.)/(INDEX/6.))/TAUM
      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.)GOTO1
      CALL MAXTLT(TIME,D(17),X1,X2,X3,D(11),D(12),A1,B1)
      CALL MAXTLT(TIME,D(17),X1,X2,X3,D(9),D(10),A2,B2)
      CALL MAXTLT(TIME,D(17),X1,X2,X3,D(13),D(14),A3,B3)
      CALL MAXTLT(TIME,D(17),X1,X2,X3,D(15),D(16),A4,B4)
    1 IF(U3.EQ.0.)GO TO 2
      CALL MAXDPS(TIME,D(18),X1,X2,X3,D(3),D(4),DA1,DB1)
      CALL MAXDPS(TIME,D(18),X1,X2,X3,D(1),D(2),DA2,DB2)
      CALL MAXDPS(TIME,D(18),X1,X2,X3,D(5),D(6),DA3,DB3)
      CALL MAXDPS(TIME,D(18),X1,X2,X3,D(7),D(8),DA4,DB4)
    2 THETA1(I,K)=A1-A2-A3+A4+DA1-DA2-DA3+DA4
    4 THETA2(I,K)=B1-B2-B3+B4+DB1-DB2-DB3+DB4
      GOTO5
    3 DO 6 K=1,INDEX
      TIME=((K-1)/(INDEX/6.))/TAUV
      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.)GOTO7
      CALL VGTTLT(TIME,D(17),X1,X2,X3,D(11),D(12),A1,B1)
      CALL VGTTLT(TIME,D(17),X1,X2,X3,D(9),D(10),A2,B2)
      CALL VGTTLT(TIME,D(17),X1,X2,X3,D(13),D(14),A3,B3)
```

```
      CALL VGTTLT(TIME,D(17),X1,X2,X3,D(15),D(16),A4,B4)
    7 IF(U3.EQ.0.)GOTO8
      CALL VGTDPS(TIME,D(18),X1,X2,X3,D(3),D(4),DA1,DB1)
      CALL VGTDPS(TIME,D(18),X1,X2,X3,D(1),D(2),DA2,DB2)
      CALL VGTDPS(TIME,D(18),X1,X2,X3,D(5),D(6),DA3,DB3)
      CALL VGTDPS(TIME,D(18),X1,X2,X3,D(7),D(8),DA4,DB4)
    8 THETA1(I,K)=A1-A2-A3+A4+DA1-DA2-DA3+DA4
    6 THETA2(I,K)=B1-B2-B3+B4+DB1-DB2-DB3+DB4
    5 RETURN
      END
      SUBROUTINE VGTTLT(TIME,U1,X1,X2,X3,P1,P3,T1,T2)
      PI=3.14159  $CHI=X2/SQRT(2.)
      R=SQRT(((X1-P1)**2)+(X2**2)+(P3**2))
      F1=(R**2)*((2.*(CHI**2))-(R*(R+P3)))
      F5=(2.*P3)*(((X2**2)*((2.*R)+P3))-((R**2)*(R+P3)))
      T1=(-U1/(4.*PI))*((((1.-5.*EXP(-6.*TIME))*(R**2))+((2.*P3)
     1 *((2.*R)+P3)))*(X2*(P1-X1)/((R**3)*((R+P3)**2))))
      T2=(U1/(4.*PI))*((((1.-5.*EXP(-6.*TIME))*F1)+F5)*(1./((R**3)*((
     1 R+P3)**2))))
      RETURN
      END
      SUBROUTINE MAXTLT(TIME,U1,X1,X2,X3,P1,P3,T1,T2)
      PI=3.14159  $CHI=X2/SQRT(2.)
      R=SQRT(((X1-P1)**2)+(X2**2)+(P3**2))
      F1=(R**2)*((2.*(CHI**2))-(R*(R+P3)))
      F5=(2.*P3)*(((X2**2)*((2.*R)+P3))-((R**2)*(R+P3)))
      T1=(-U1/(4.*PI))*((((2.-EXP((-5./6.)*TIME))*(R**2))+((2.*P3)*
     1 ((2.*R)+P3)))*((X2*(P1-X1))/((R**3)*((R+P3)**2))))
      T2=(U1/(4.*PI))*((((2.-EXP((-5./6.)*TIME))*F1)+F5)*(1./((R**3)
     1 *((R+P3)**2))))
      RETURN
      END
      SUBROUTINE WRITE1
      WRITE(7,1)
    1 FORMAT(*SPECIFY INITIAL AND FINAL SLIP VALUES,*,/,
     1 *RIGHT-LATERAL STRIKE-SLIP IS <0.*,/,
     1 *DIP-SLIP >0. FOR 'X2>0. SIDE' DOWN*,/,
     1 *FORMAT=U1IN,U1FN,U3IN,U3FN*)
      RETURN
      END
      SUBROUTINE WRITE2(C)
      DIMENSION C(40),A(30)
      WRITE(7,3)
    3 FORMAT(*SPECIFY INITIAL AND FINAL COORDINATES*,/,
     1 *OF DIP-SLIP ZONE CORNERS AS INDICATED.*,/,
     1 *D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,D3X1FN*)
      CALL GETNUM(A) $C(1)=A(1) $C(2)=A(2) $C(4)=A(3) $C(5)=A(4)
      C(9)=A(5) $C(10)=A(6) $C(12)=A(7) $C(13)=A(8)
      C(3)=C(1)  $C(7)=C(5)  $C(6)=C(4)  $C(8)=C(2)
      C(11)=C(9)  $C(16)=C(10)  $C(14)=C(12)  $C(15)=C(13)
      RETURN
      END
      SUBROUTINE WRITE3(C)
      REAL A(30)
      DIMENSION C(40)
      WRITE(7,8)
    8 FORMAT(*SPECIFY INITIAL AND FINAL VALUES *,/,
     1 *OF STRIKE-SLIP ZONE COORDINATES AS INDICATED*,/,
```

```
      1 *C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN*)
        CALL GETNUM(A) $C(17)=A(1) $C(18)=A(2) $C(20)=A(3) $C(21)=A(4)
        C(25)=A(5) $C(26)=A(6) $C(28)=A(7) $C(29)=A(8)
        C(19)=C(17) $C(22)=C(20) $C(23)=C(21) $C(24)=C(18)
        C(27)=C(25) $C(30)=C(28) $C(31)=C(29) $C(32)=C(26)
        RETURN
        END
        SUBROUTINE WRITE4(TRIGGR,M1,M2,M3,KCHECK)
        REAL A(30)
        M1=100 $M2=100 $M3=100
        WRITE(7,150)
  150   FORMAT(*DO YOU WISH THE 'TRIGGER' OPTION?*,/,
     1  *'YES' OR 'NO '*)
        READ(7,151)TRIGGR
  151   FORMAT(A3)
        KCHECK=0
        IF(TRIGGR.EQ.3HNO )GO TO 160
        WRITE(7,157)
  157   FORMAT(*SPECIFY 'TRIGGER' PARAMETERS IN THE FORM*,/,
     1  *D(M1).GE.C(M2)*)
        CALL GETNUM(A) $M1=A(1) $M2=A(2)
        WRITE(7,158)
  158   FORMAT(*SPECIFY WHICH SLIP COMPONENT IS TO BE =0.*,/,
     1  *WHEN D(M1).LT.C(M2)*,/,
     1  *STRIKE-SLIP=S, DIP-SLIP=D*)
        READ(7,159)SLIP
  159   FORMAT(A1)
        IF(SLIP.EQ.1HS)M3=17
        IF(SLIP.EQ.1HD)M3=18
  160   CONTINUE
        RETURN
        END
        SUBROUTINE GETNUM(P)
        DIMENSION P(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) $ P(J)=N/F*M $ GOTO 6
      9 FORMAT(80R1)
        END
        SUBROUTINE MAXDPS(TIME,R1,X1,X2,X3,R2,R3,T1,T2)
        ALPHA=TIME*R1*X1*X2*X3*R2*R3
        T1=ALPHA
        T1=T2=0.
        RETURN
        END
        SUBROUTINE VGTDPS(TIME,R1,X1,X2,X3,R2,R3,T1,T2)
        ALPHA=TIME*R1*X1*X2*X3*R2*R3
        T1=T2=0.
        RETURN
        END
```

ST3PHS, ST3PSZ

STNRML, SHRSTN

DBLNRM

C

DOCUMENTATION FOR PROGRAMS

ST3PHS/ST3PSZ/STNRML/SHRSTN/DBLNRM

These programs compute the shear stress, strain, and shear displacement produced by a screw dislocation embedded in an elastic three-phase material (ie. a vertical slab between two quarter spaces) using the equations and geometry in Chou (1966). The input consists of the dislocation position, the shear moduli of the three phases, and a series of flags that allow selection of various options. The output consists of the shear stress, strain, and shear displacement, plus a listing of the values assigned to the variables in the program.

ST3PHS and ST3PSZ compute and plot non-normalized quantities as a function of distance for a single and double dislocation respectively. STNRML and SHRSTN compute the normalized quantities and plot them as a function of distance and shear modulus contrast respectively. In the latter 2 programs, the quantities computed for the three-phase material are normalized by the same quantity computed for a homogeneous half-space. For a description of DBLNRM see the addendum.

Access and Use:

The programs are intended for use on the LBL 6600 B or C machines and the Tektronix (4010-1) terminal, and will automatically link to the necessary plotting routines for interactive graphics. They are stored on the library: MCHUGH. Once the user has logged on to the terminal (the programs require approximately 70 K of core space), the programs may be accessed using the command:

^LOAD, (subset), MCHUGH

where 'subset' is the program desired. The programs are intended for immediate use, so that the ^LOAD command may be followed immediately by ^RUN.

Input:

The geometry used in these programs is shown in Figure 1. The shear moduli are in cgs units, and are entered in the form wEv (eg. 3.0 E 11). The dislocation position, slab thickness, end points of profile, and position in the slab are in kilometers, and the Burger's vector is in millimeters. If the Burger's vector of the screw dislocation is positive, the stress and strain are positive and the displacement is left-lateral. The normalized quantities (ie. the value in the three-phase material divided by the same value in a homogeneous half-space) also require specification of the shear modulus of an elastic half-space. The input steps and computer responses are as follows:

I) ST3PHS, ST3PSZ, and STNRML --

a)

1) Shear moduli-in cgs units-MU1, MU2, MU3

Enter the three shear moduli for the geometry in Figure 1.

2) Shear modulus of homogeneous half-space. This response occurs in the STNRML program. The homogeneous half-space stresses, strains, and displacements will be used to normalize the three-phase quantities.

b) Position of dislocation in kms., (A, LAMBDA)

NOTE: A>0, LAMBDA>0.

Enter A (the x coordinate of the dislocation; note, the dislocation must be in the vertical slab, that is $D \leq A \leq 0.$) and λ (the y coordinate of the dislocation). Even though the medium occupies the region $y \leq 0.$ and points in the slab the region $X \leq 0.$, both A and λ are entered as positive quantities.

c)

1) Thickness of slab in kms. - D

Enter the slab thickness ($D \geq 0.$)

2) Width of slip plane

This step occurs only in ST3PSZ; enter the fault plane width (ie. distance between the 2 screw dislocations).

d) Burgers Vector, in mms - B

B<0 for right-lateral slip

Enter the Burger's vector.

e) 1 = amplitude vs. X, 2 = amplitude vs. y

Select the appropriate profile.

f) 1 = XZ stress, 2 - YZ stress

Select the shear stress desired. This will also determine which shear strain and displacement is to be computed because both the strain and displacement are computed from the stress.

g) Enter minimum and maximum X values in kms. OR:

Enter minimum and maximum Y values in kms.

Select the end points of the profile specified in part e. (NOTE: Treat the y values as though the medium occupied the region $y \geq 0.$)

h) Depth of profile in kms. OR

X coordinate of profile in kms.

Enter the position of the profile.

i) NOTE: Number of data points (numerical values)
Upper limit on series (numerical values)

No response required. The number of data points is the number of points along the profile for which a stress, strain, or displacement is computed. The 'upper limit on series' is the number of terms calculated in the infinite series specifying the shear stress. (To change these two values NUM and IUPPER respectively, lines 56, 58, and 59 in ST3PHS, ST3PSZ, and STNRML respectively must be altered).

j) If the depth selected in part h) puts the profile above the surface of the medium or below the dislocation, a warning is issued, the computer prints out the values of the variables, and the program returns to its starting position. If the profile specified is too close to the dislocation a warning is issued and the program must be restarted (this occurs if the profile is within a distance ε of the dislocation, where ε is the reciprocal of the grid point density).

II) SHRSTN

a) Shear moduli --in cgs units-- MU1, MU2LOW, MU2HIGH, MU3 = MU1

Enter the shear modulus of region 1 (μ_1), and the lower and upper bounds on the shear modulus in the vertical slab (μ_2^{low} and μ_2^{high}). It is assumed that $\mu_3 = \mu_1$, so it is not necessary to specify μ_3. If it is desired that $\mu_3 \neq \mu_1$, lines 31 and 85 in the main program must be altered. The shear modulus contrast, μ^*, is defined to be μ_2/μ_1.

The next responses from the computer correspond to parts a-2, and b through h as outlined in Section I. The appropriate quantities must be entered as indicated. Then a flag must be set as follows:

1 = Compute variation of global minimum of profile

2 = Compute variation of global maximum of profile

3 = Compute variation of stress and strain only at a point on the profile

4 = Compute variation of stress, strain and displacement at a point on the profile

If a 1 or 2 is entered, the computer will determine the overall (global) minimum or maximum amplitude of the normalized quantity along the profile specified and display that minimum or maximum as a function of modulus contrast, μ^* . Because a separate profile for each modulus contrast increment must be computed these options require much computer time. If a 3 or 4 is entered the normalized shear stress, strain and/or displacement at a particular point only will be computed as a function of μ^*. After specifying this flag, the computer will respond as in part i, Section I. If it is desired to change NUM and IUPPER, line 79 (in program SHRSTN) must be altered. The warnings in part j, section I also may occur in SHRSTN.

Output:

I) ST3PHS, ST3PSZ, and STNRML --

The output consists of a profile of xz or yz stress, strain, and displacement (these quantities are normalized by the homogeneous half-space values in the case of STNRML) versus x or y between the endpoints specified. The maximum and minimum values are printed out also. All plots will be scaled by the maximum and minimum values computed unless otherwise specified. The axes and units are set automatically.

a) MIN/MAX values of xz (or yz) stress--units are dynes/(cm2)

OR: MIN/MAX values of xz (or yz) microstrain

OR: MIN/MAX values of w(x) /or w(y)/ versus x (or y)

No response required; the values are printed automatically.

b) Plot title, 80 characters maximum

Enter up to 80 alphanumeric characters for the title of the plot.

c) Set vertical scale? Y or N (= blank)

Entering a Y (yes) will cause 'MIN/MAX Y values' to be printed, in which case the upper and lower limits of the vertical scale must be specified. Entering N (no) or a blank (space) causes the upper and lower limits of the vertical scale to be the maximum and minimum values computed.

d) Set horizontal scale? Y or N (= blank)

Entering a Y causes 'MIN/MAX X values' to be printed, then the endpoints of the horizontal axis must be specified. Otherwise the procedure is analogous to that in part c.

e) Skip?

Entering a Y (yes) causes the plot to be skipped; an N or blank will cause the graph to be displayed.

f) The plots occur in the following order, and in each case the computer will ask for the scaling desired:

1) Stress
2) Strain
3) Displacement

The results will be either the xz or yz quantities versus x or y depending upon which flags were set in the input section. NOTE: If the endpoints of the y profile lie above the free surface and/or below the dislocation, the endpoints are automatically reset to occur <u>at</u> the free surface and/or <u>at</u> the dislocation.

If an xz quantity versus y, or yz quantity versus x, is requested, the displacement will not be given. Instead the following is printed:

NOTE: Only w(x) vs. X or w(y) vs. Y is computed

g) After the plot is finished, entering a blank (space) causes the computer to respond:

0 = Restart, 1 = New Plot

Entering a zero will cause the program to move to Section h below; a 1 will cause the computer to ask for scaling information and the same plot with new scaling will be displayed. Any other number will cause the computer to move to the next graph until the displacement values have been plotted.

h) At this point the values of the various flags and variables used in the program will be printed out. A description of these quantities is given in the appendix.

i) 0 = Restart

1 = Stop

2 = Restart with same values, different profile

To restart with all new values, enter 0; to stop the program, enter 1.

To use the same input values but display a different profile, enter a 2.

If either a zero or a 2 is entered, the computer will request the information it needs as before.

NOTE: The profile within the vertical slab may not be symmetric if the grid point density is too small. In some of the examples that follow it will be found that the stress or strain at x = 0.0 does not appear to be the same as at x = -1.0 (the slab extends from x = -1.0 to x = 0.0 in these examples). This discrepancy is caused by too large a grid point interval. That is, the stress is computed at a point very slightly to the left of the x = -1.0 boundary, at x = -0.9, and, near the right-hand boundary at a point slightly to the left of x = 0.0. Consequently the stress or strain profile within the slab appears to be non-symmetric. This problem can be avoided by increasing the number of grid points (NUM, eg. line 56 in ST3PHS) or reducing the distance between the end points of the profile.

ADDENDUM:

DBLNRM

This program computes the normalized shear stress, strain, and displacement produced by a finite width slip zone (ie. two screw dislocations, with equal and antiparallel Burger's vectors, separated by some non-zero width). The input sequence and computations are similar to those in programs ST3PSZ and STNRML.

NOTE: A finite width slip zone will produce stresses and strains that approach zero outside the fault zone; this effect is particularly noticeable for widths less than, or on the order of 100 kms. The displacements are computed by summing the strains and adjusting the profile so that the displacement at a point above the dislocation is numerically zero*. Consequently, the normalized displacement profile for finite width slip zones suffers from: 1) numerical approximation errors in the summation process, and 2) numerical error as the strain approaches zero and at points near the dislocation. No completely satisfactory way has been found to avoid these difficulties and keep the grid point density finite. The normalized displacements should be viewed with suspicion until their accuracy has been checked against the results from the other programs.

*The zero point of the displacement profile is found by comparing the x or y position of each grid point to the dislocation position. When a point is found that is within one-one hundredth grid point interval of the dislocation, the displacement value at that point is subtracted from all other displacement values in that profile. It is also possible that no point on the profile is within one-one hundredth grid point interval of the dislocation, in which case the quantity subtracted from the displacement values is:

$$\left(D(\text{final}) - D(\text{initial})\right)/\ 2. + D\ (\text{initial})$$

This difference between these two definitions of the zero point is slight, but because of numerical approximation errors the differences are quite noticeable in the normalized displacement profiles for finite width slip zones.

REFERENCES

Chou, Y. T., Screw dislocations in and near lamellar inclusions, Phys. Stat. Sol., 17, 509-516, 1966.

APPENDIX

Description of Variables and Flags Occurring in Output

MU1, MU2, MU3 - shear moduli in regions 1, 2, and 3 of three-phase material (figure 1).

MUH - shear modulus of homogeneous half-space

M, K - modulus contrasts as defined in Chou (1966)

$$M = (\mu_3 - \mu_2)/(\mu_3 + \mu_2)$$

$$K = (\mu_1 - \mu_2)/(\mu_1 + \mu_2)$$

MU2LOW, MU2HIGH - lower and upper limits on shear modulus in slab (used in SHRSTN)

Iflag - 1 = xz, 2 = yz quantities

Kflag - 1 = x, 2 = y profile

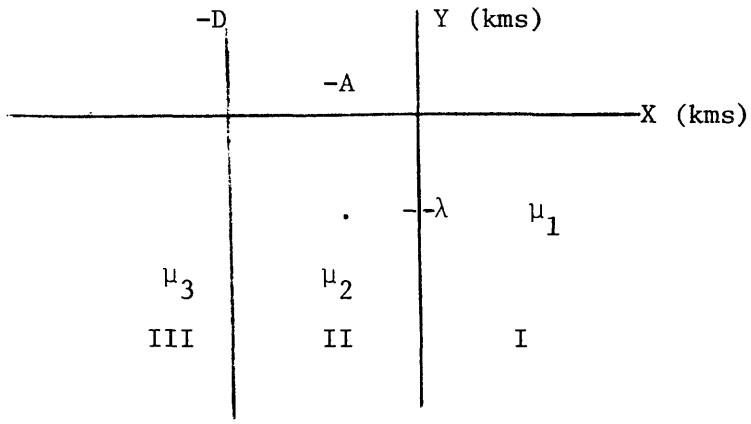
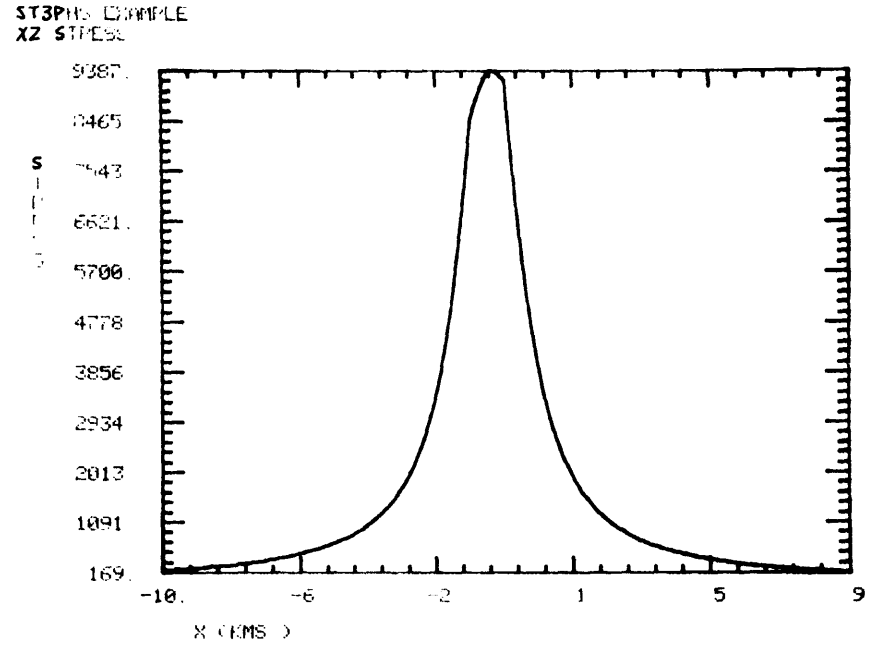


FIGURE 1

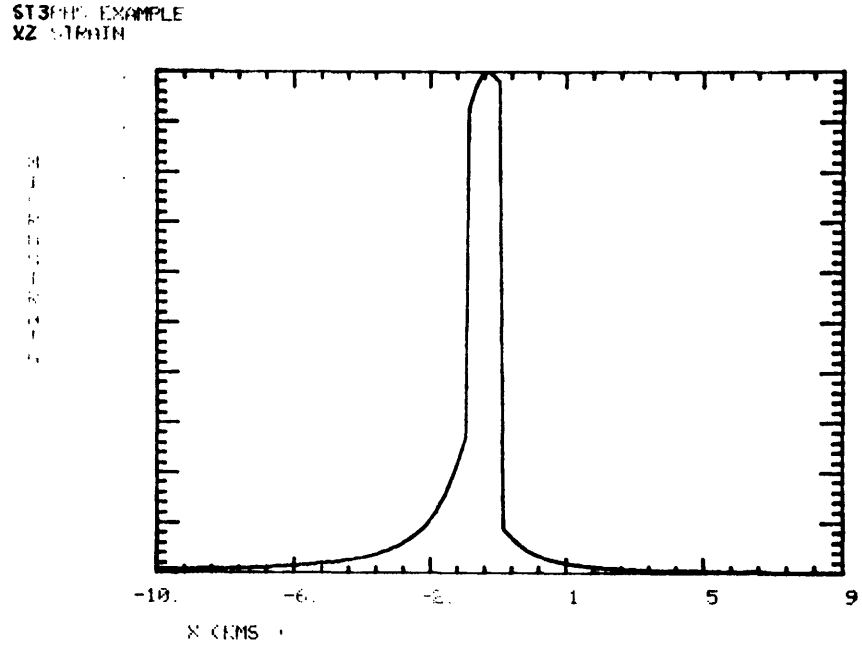
Geometry and Notation used in Programs ST3PHS, ST3PSZ, STNRML, SHRSTN

EXAMPLE FROM ST3PHS

```
^LOAD .[SAM,MCHUGH]
LOAD - COMPLETE - ENTERING EDIT
OK - EDIT
^RUN
SHEAR MODULI--IN [illegible] MU2
3E11 3E10 1E11
POSITION OF DISLOCATION--IN KMS - (A/LAMBDA)
NOTE---A>0 , LAMBDA>0
-5 1!
THICKNESS OF SLAB, IN KMS --C
1!
BURGERS VECTOR, IN MMS --B
B<0 FOR RIGHT LATERAL SLIP
1!
1=AMPLITUDE VS X, 2=AMPLITUDE VS Y
1!
1=XZ STRESS, 2=YZ STRESS
1!
ENTER MINIMUM AND MAXIMUM X VALUES, IN KMS.
-10 10!
DEPTH OF PROFILE, IN KMS.
0!
NOTE--NUMBER OF DATA POINTS =   200
      UPPER LIMIT ON SERIES =    20
MIN/MAX VALUES OF XZ STRESS--UNITS ARE DYNES/(CM2)
    1 6955E+02          9.3871E+03
PLOT TITLE, 80 CHARACTERS MAXIMUM
ST3PHS EXAMPLE!
SET VERTICAL SCALE? Y OR N(=BLANK)
 !
SET HORIZONTAL SCALE? Y OR N(=BLANK)
 !
SKIP?
```

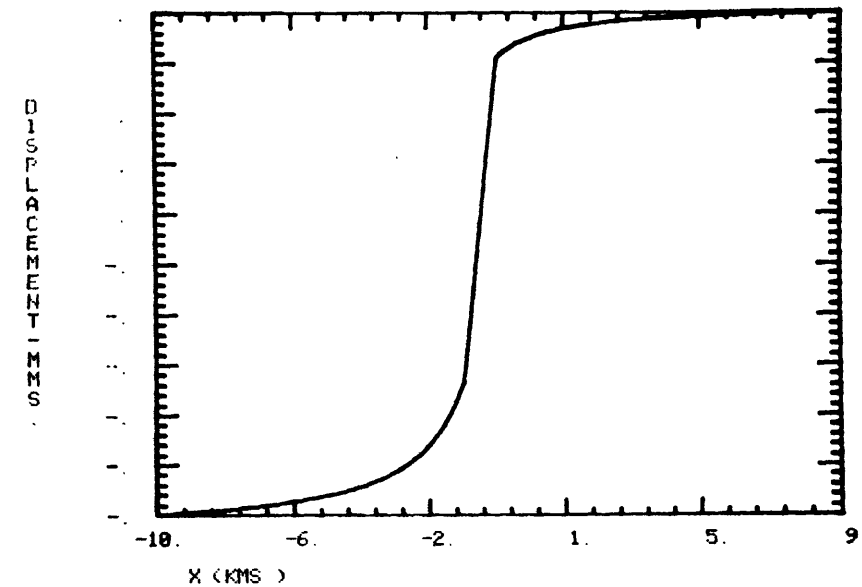



```
 0=ME-START, 1=NEW PLOT
?1
MIN-MAX VALUES OF /Z MICROSTRAIN
     5 6516E-04          3 1290E-01
SET VERTICAL SCALE? Y OR N(=BLANK)
 ?
SET HORIZONTAL SCALE? Y OR N(=BLANK)
 ?
SKIP?
```

```
!0=PRE-START, 1=NEW PLOT
2!
MIN-MAX VALUES OF W(X) VS. X
    -5.5602E-01          3.9560E-01
SET VERTICAL SCALE? Y OR N(=BLANK)
!
SET HORIZONTAL SCALE? Y (OR N(=BLANK)
!
SKIP?
```

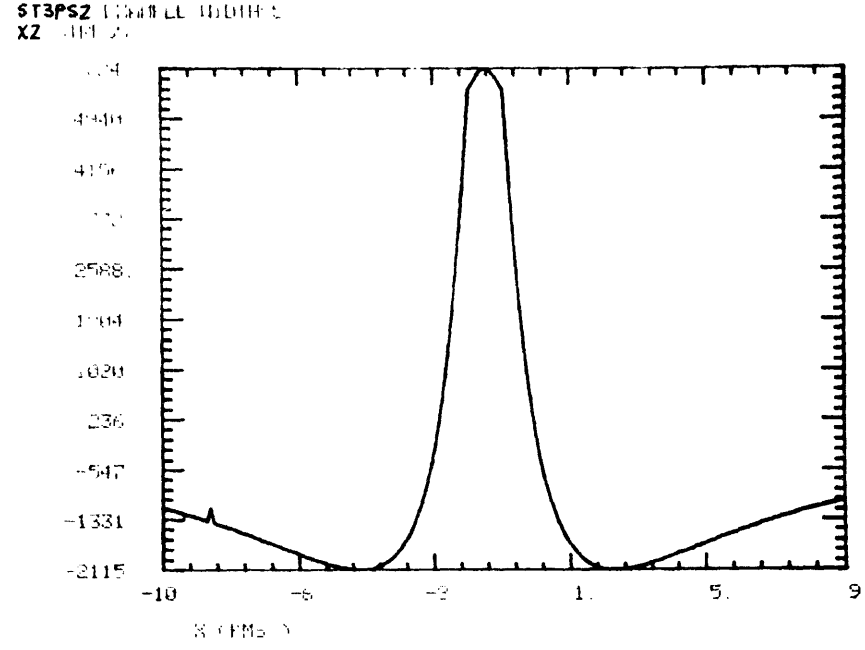

ST3PMS EXAMPLE
W(2) VS. X



```
 1
0=RE-START, 1=NEW PLOT
2!
MU1 =            3.0000E+11      MU2 =            3.0000E+10
MU3 =            1.0000E+11        M =            5.3846E-01
  K =            8.1818E-01
DISLOCATION POSITION = -         500, -     1.000 KMS.
SLAB THICKNESS       =         1.000 KMS.
BURGERS VECTOR       =        -1.000 MMS.
X: STRESS/STRAIN VS. X
0=RE-START
1=STOP
2=RE-START WITH SAME VALUES, DIFFERENT PROFILE
1!
OK - ^EDIT
```

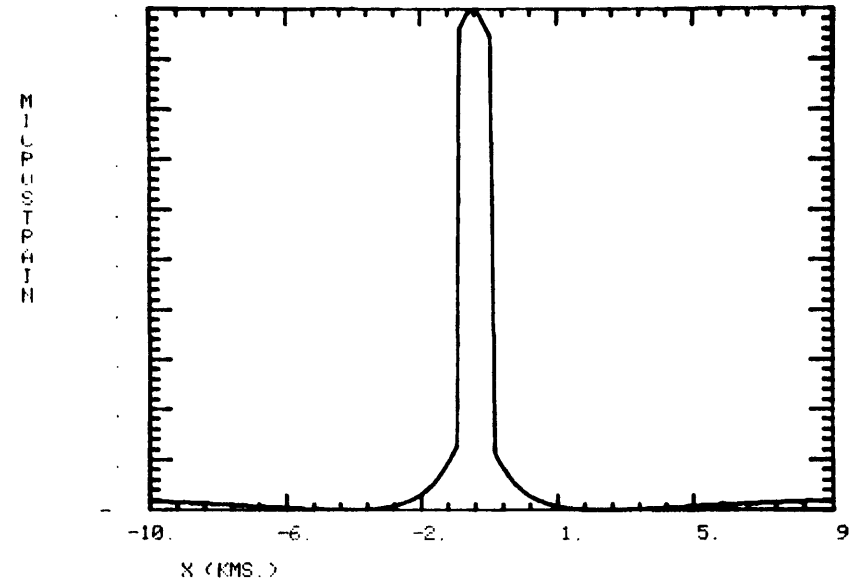
EXAMPLE FROM ST3PSZ

```
SHEAR MODULI -IN [illegible] DATA   "1"=MU, M/Z
3E11 3E10 2E11
POSITION OF DISLOCATION= -IN KM, [illegible] SAME AS
NOTE    0 0 = COMMON B
.5
THICKNESS OF SLAB  IN KM=  "
!
DEPTH OF SLIP PLANE= IN
5
BURGERS VECTOR, IN MM  =5
B<0 FOR RIGHT LATERAL  1.0
!
AMPLITUDE VS  X, Z=AMPLITUDE AS
!
1=X, 2=TIME, 3=YZ =TREC=
!
ENTER MINIMUM AND MAXIMUM  VALUES, IN KM,
 10 10
DEPTH OF PROFILE, IN KM=
!
NOTE --NUMBER OF DATA POINTS=     200
       UPPER LIMIT ON SERIES =     20
MIN MAX VALUES OF XZ STRESS--UNITS ARE DYNES/CM2
    -2 1156E+03          5 7247E+03
PLOT TITLE, 80 CHARACTERS MAXIMUM
ST3PSZ EXAMPLE WIDTH=5
SET VERTICAL SCALE? Y OR N =BLANK
 !
SET HORIZONTAL SCALE? Y OR N =BLANK
 !
SHIFT
```

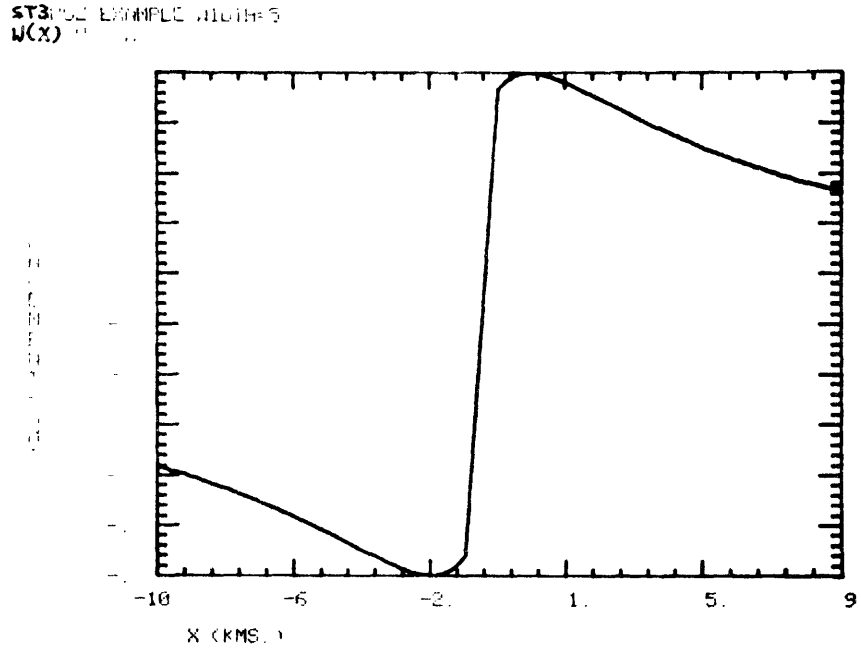



0= [illegible] 1=NEW PLOT
2
MIN MAX VALUES OF XZ MICROSTRAIN
[illegible]
SE [illegible]
SE [illegible]
SE [illegible]

ST3PSZ (SAMPLE/WIDTH=5
XZ MICROSTRAIN



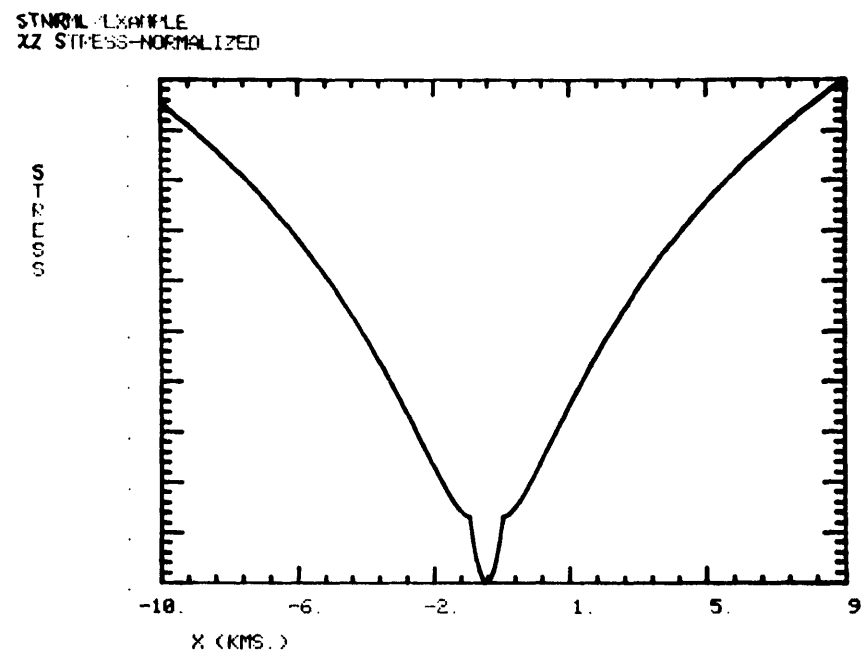
```
 !
0=RE-START, 1=NEW PLOT
2!
!!!! MAX VALUES OF W(X) VS. X
    -2.0400E-01              1.9825E-01
SET VERTICAL SCALE? Y OR N=BLANK)
 !
SET HORIZONTAL SCALE? Y OR N=BLANK)
 !
SKIP?
```

```
0=RE-START, 1=NEW PLOT
2
MU1 =              3.0000E+11      MU2 =              3.0000E+10
MU3 =              3.0000E+11        M =              0.1818E-01
  K                8.1818E-01
DISLOCATION POSITION = -          .500  -      1.000 KMS.
WIDTH OF SLIP ZONE   =          5.000
SLIP THICKNESS       =          1.000 (MS.
BURGERS VECTOR       =         -1.000 MM.
1: STRESS-STRAIN VS. X
0=RE-START
1=STOP
2=RE-START WITH SAME VALUES, DIFFERENT PROFILE
1
OK - ^EDIT
```

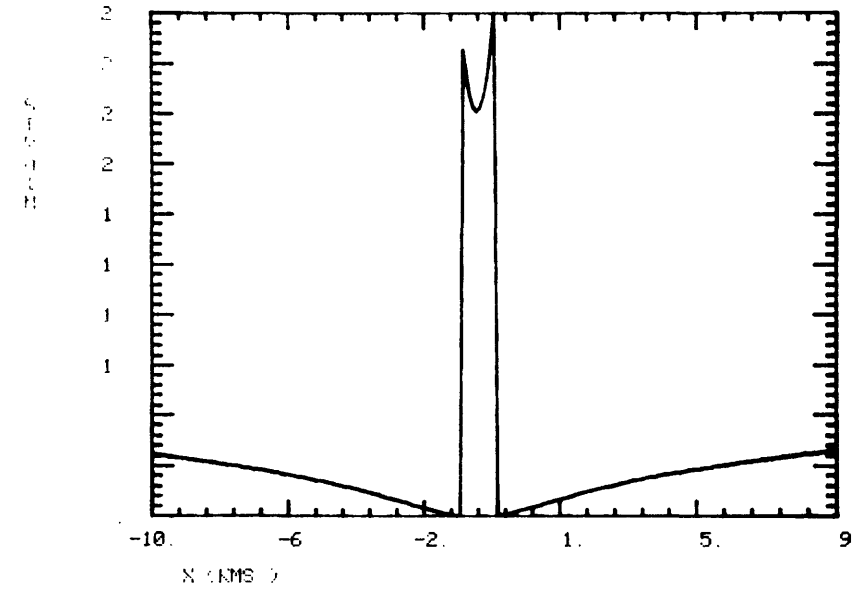
EXAMPLE FROM STNRML

```
SHEAR MODULI--IN CGS UNITS--MU1,MU2,MU3
3E11 3.1E10 3E11!
SHEAR MODULUS OF HOMOGENEOUS HALF-SPACE
3E11!
POSITION OF DISLOCATION--IN KMS , (=0,1 MEGA)
NOTE --H=0 , LAMBDA=0
.5 1!
THICKNESS OF SLAB, IN KMS --0
1!
BURGERS VECTOR, IN MMS --0
B<0 FOR RIGHT LATERAL SLIP
1!
1=AMPLITUDE VS. X, 2=AMPLITUDE VS. Y
1!
1=XZ STRESS, 2=YZ STRESS
1!
ENTER MINIMUM AND MAXIMUM X VALUES, IN KMS
-10 10!
DEPTH OF PROFILE, IN KMS
0!
NOTE--NUMBER OF DATA POINTS =   200
      UPPER LIMIT ON SERIES =    10
MIN,MAX VALUES OF XZ NORMALIZED STRESS
    2.3212E-01          6.0766E-01
PLOT TITLE, 80 CHARACTERS MAXIMUM
STNRML/EXAMPLE!
SET VERTICAL SCALE? Y OR NC=BLANK)
 !
SET HORIZONTAL SCALE? Y OR NC=BLANK)
 !
SKIP?
```

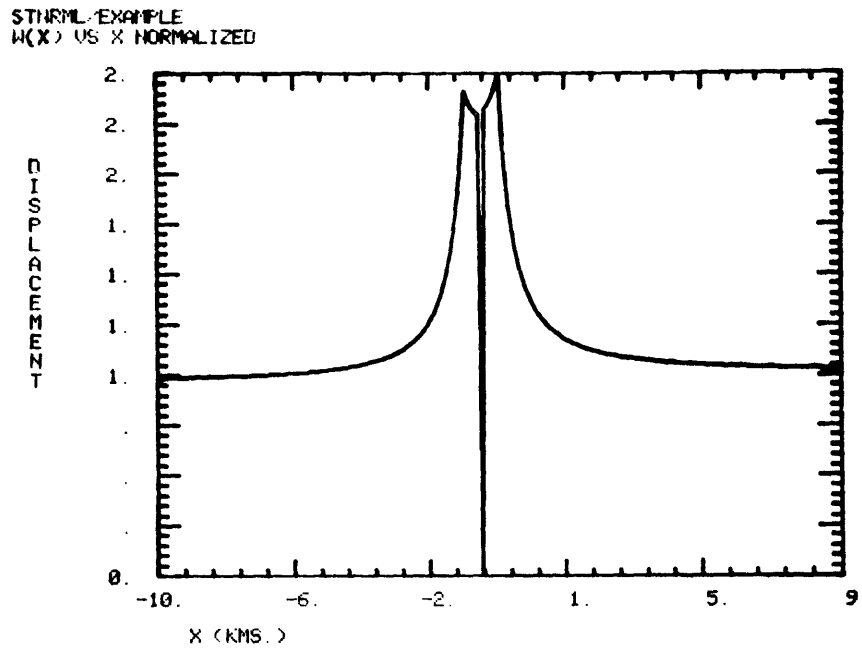



0=S =START, 1=NEW PLOT
2
THIS MAY VALUES OF XZ NORMALIZED STRAIN
2.81734E-01 [illegible]
S [illegible] VERTICAL SCALE= [illegible] OF MAXIMUM
[illegible]
[illegible] HORIZONTAL SCALE? [illegible] OR (Y=ENTER)
[illegible]

STRAIN-EXAMPLE
XZ STRAIN NORMALIZED



```
 |
0=RE-START, 1=NEW PLOT
3|
MIN MAX VALUES OF W(X) VS. X
     0                    2.5327E+01
SET VERTICAL SCALE? Y OR N(=BLANK)
 |
SET HORIZONTAL SCALE? Y OR N(=BLANK)
 |
SKIP?
```

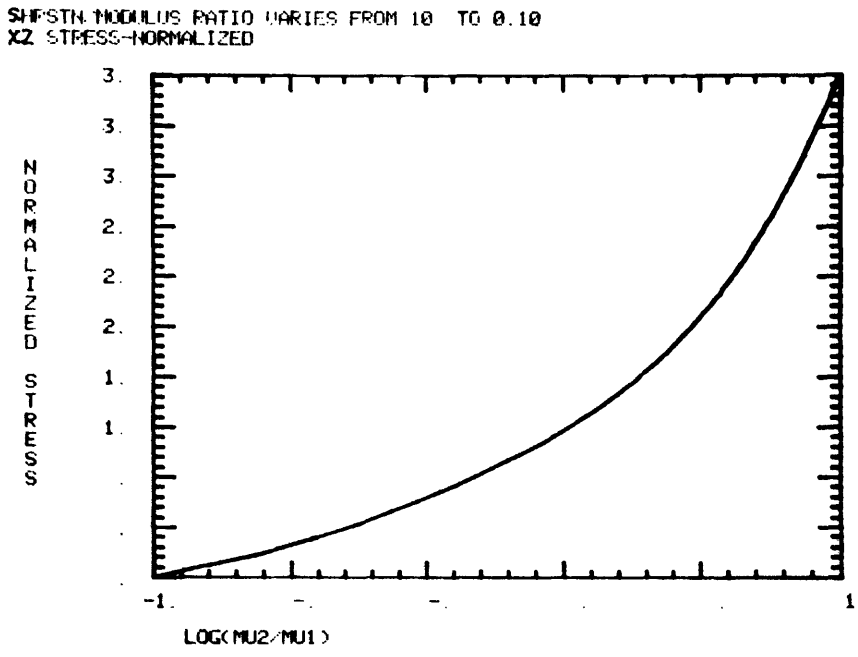
```
 1
0=RE-START, 1=NEW PLOT
?1
MU1 =              3.0000E+11      MU2 =              3.0000E+10
MU3 =              3.0000E+11      MUH =              3.0000E+11
  M =              8.1813E-01        K =              8.1813E-01
DISLOCATION POSITION = -              .500,  -      1.000 KMS.
 LAB THICKNESS                =        1.500 KMS.
BURGER's VECTOR               =       -1.000 MMS.
 C STRESS/STRAIN VS. X
0=RE-START
1=STOP
2=RE-START WITH SAME VALUES, DIFFERENT PROFILE
?1
OK  - ^EDIT
```

EXAMPLE FROM SHRSTN

```
SHEAR MODULI--IN CGS UNIT --MU1,MU LOW,MU2HIGH,MU3 MU1
3E11 3E14 3E12
SHEAR MODULUS OF HOMOGENEOUS HALF-SPACE
3E11
POSITION OF DISLOCATION--IN KMS ,  H,LAMBDA
NOTE --H > 0 , LAMBDA>0
.5 1
THICKNESS OF SLAB, IN KMS --D
1
BURGERS VECTOR, IN MMS.--B
B<0 FOR RIGHT LATERAL SLIP
1
1=AMPLITUDE VS  X, 2=AMPLITUDE VS  Y
1
1=XZ STRESS, 2=YZ STRESS
1
1=COMPUTE VARIATION OF GLOBAL MINIMUM OF PROFILE
2=COMPUTE VARIATION OF GLOBAL MAXIMUM OF PROFILE
3=COMPUTE VARIATION OF STRESS AND STRAIN ONLY AT A
  POINT ON THE PROFILE
4=COMPUTE VARIATION OF STRESS, STRAIN, AND DISPLACEMENT
  AT A POINT ON THE PROFILE
3
ENTER X COORDINATE OF POINT
-.5
DEPTH OF PROFILE, IN KMS.
0
```

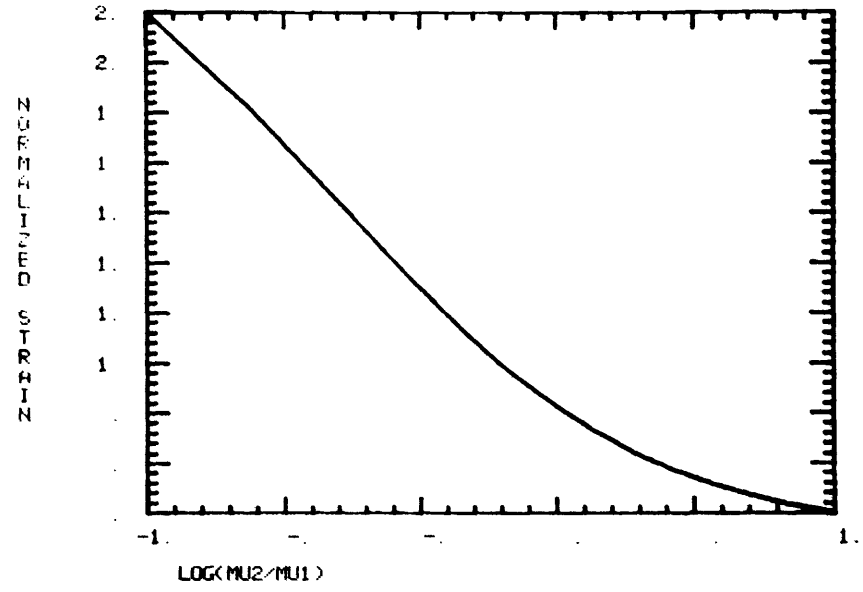


```
NOTE -NUMBER OF DATA POINTS =   10
     UPPER LIMIT ON SERIES =   20
MIN MAX VALUES OF NORMALIZED STRESS
VER.OD SHEAR MODULUS CONTRAST
          2.3214E-01                3.7198E+00
PLOT TITLE, 80 CHARACTERS MAXIMUM
SHRSTN/MODULUS RATIO VARIES FROM 10  TO 0.10
SCT VERTICAL SCALE? Y OR N(=BLANK)
Y
SET HORIZONTAL SCALE? Y OR N(=BLANK)
Y
-1 1 MAX X VALUES
-1 1
0 10
```



0= 1 NEW PLOT
2
SET
SET
VALUES

SHRSTN. MODULUS RATIO VARIES FROM 10 TO 0.10
XZ STRAIN-NORMALIZED

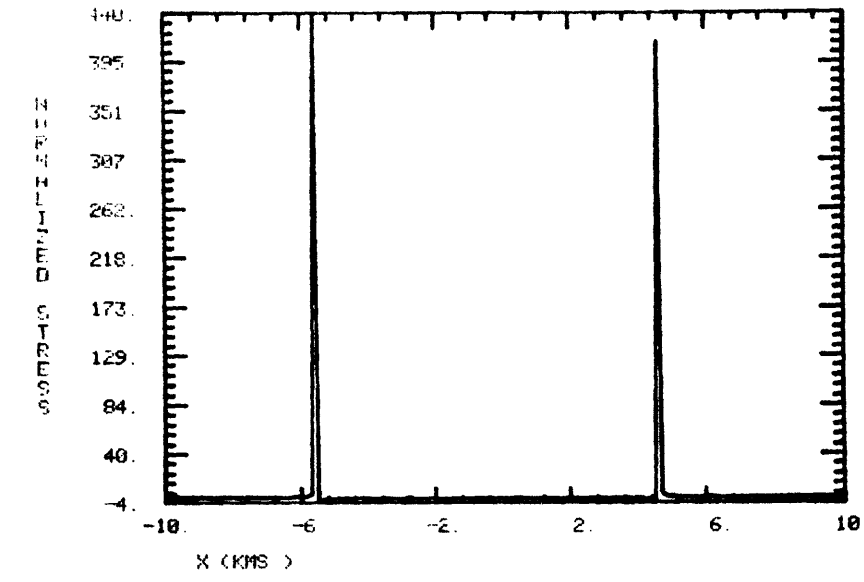



```
0=ALL START, 1=NEW PLOT
2
MU1       =          3.0000E+11     MU LOW  =          3.0000E+10
MU HIGH   =          3.0000E+12     MU2     =          3.0000E+11
MUH       =          3.0000E+11     M       =         -8.1653E-01
K         =         -8.1653E-01
3 STRESS/STRAIN VS  X
4 LOCATION POSITION =  -      500.  -     1.000 KMS.
5 SHR THICKNESS     =        1.000 KMS
6 PARAMETERS VECTOR =       -1.000 MMS.
7 ALL-START
8 SETUP
9 RE-START WITH SAME VALUES, DIFFERENT PROFILE
10
99 = >EDIT
```

EXAMPLE FROM DBLNRM

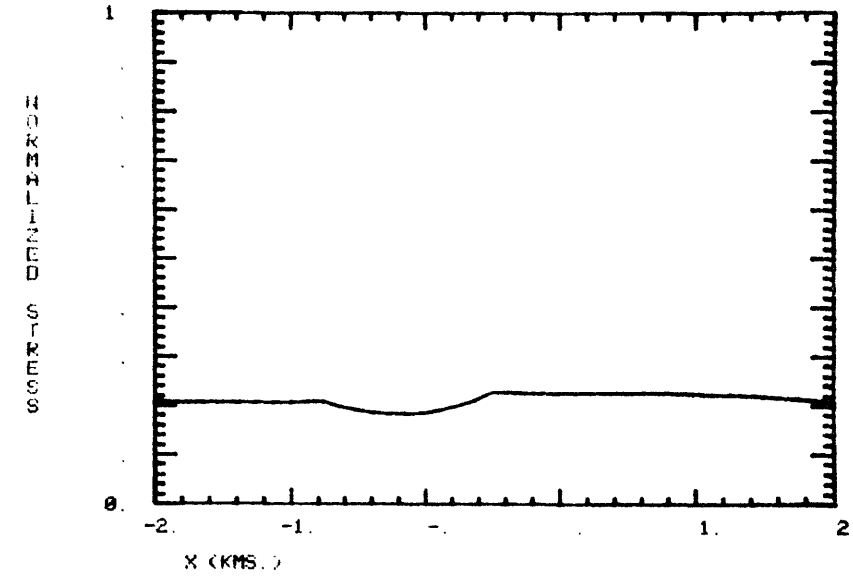
```
SHEAR MODULI--IN CGS UNITS--MU1,MU2,MU3
3E11  3E10  1E11!
SHEAR MODULUS OF HOMOGENEOUS HALF-SPACE
3E11!
POSITION OF DISLOCATION--IN (MU2 + A LAMBDA)
NOTE  A=0   LAMBDA=0
.5  !
THICKNESS OF SLAB, IN KMS --T>0
1!
WIDTH OF SLIP PLANE, IN KMS ---W>0
25!
NUMBERS VECTOR, IN MMS --B
(-1 FOR RIGHT LATERAL SLIP
1!
1=AMPLITUDE VS. X, 2=AMPLITUDE VS. Y
1!
1=XZ STRESS, 2=YZ STRESS
1!
ENTER MINIMUM AND MAXIMUM X VALUES, IN KMS.
-10  10!
DEPTH OF PROFILE, IN KMS
0!
NOTE--NUMBER OF DATA POINTS =   200
      UPPER LIMIT ON SERIES =    20
MIN/MAX VALUES OF NORMALIZED XZ STRESS
   -3.9629E+00          4.4036E+02
PLOT TITLE, 80 CHARACTERS MAXIMUM
DBLNRM/WIDTH =25!
SET VERTICAL SCALE? Y OR NK=BLANK)
!
SET HORIZONTAL SCALE? Y OR NK=BLANK)
Y!
MIN/MAX X VALUES
-10 10!
SKIP?
```


DBLNRM/WIDTH=25
NORMALIZED XZ STRESS



0=RE[illegible]PT, 1=NEW PLOT
1
PLOT TITLE, 80 CHARACTERS MAXIMUM
DBLHM .25=WIDTH
SET VERTICAL SCALE? 1 OR N=SECANT
Y
[illegible] Y VALUES
[illegible]
SET HORIZONTAL SCALE? [illegible]
Y
[illegible] X VALUES
[illegible]

DBLHM .25=WIDTH
NORMALIZED X2 STRESS

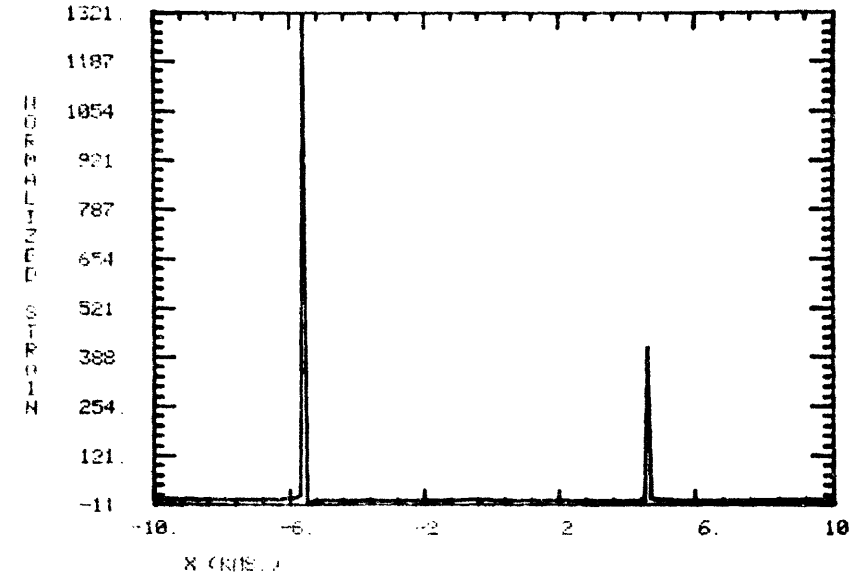


0=RE START, 1=NEW PLOT
2
MIN MAX VALUES OF NORMALIZED X2 STRAIN
-1.1889E+01 1.3211E+01
SET VERTICAL SCALE? Y OR N (BLANK)

SET HORIZONTAL SCALE? Y OR N (BLANK)

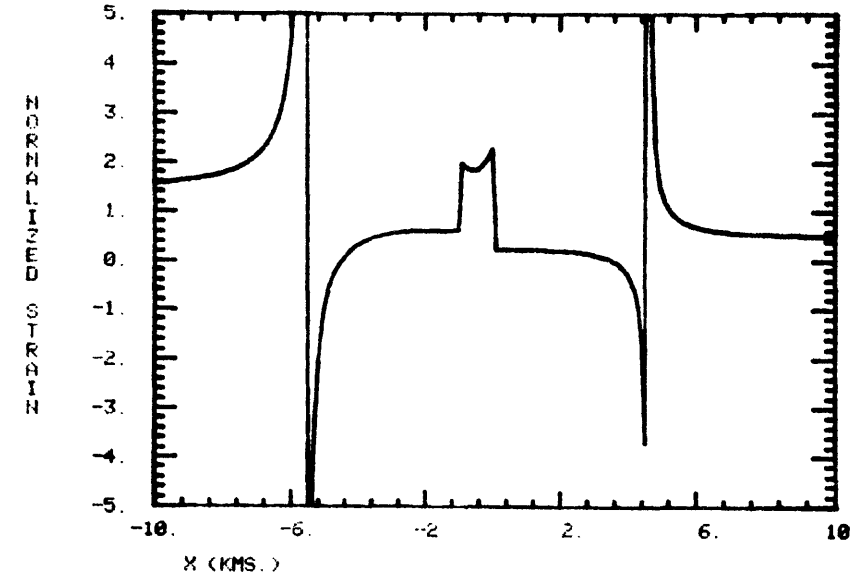
SKIP?

DOUBLE WIDTH =25
NORMALIZED X2 STRAIN

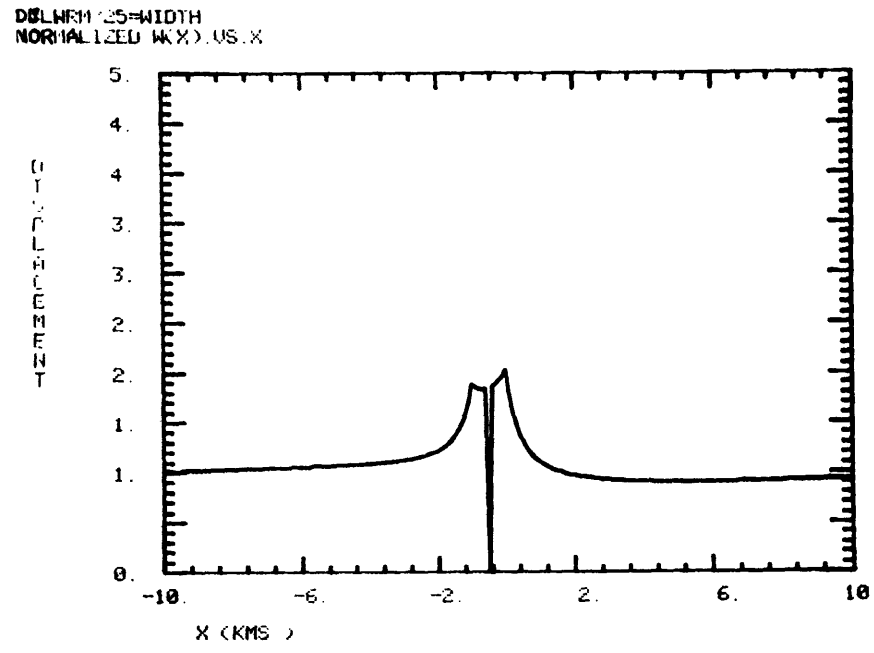


0=X - START, 1=NEW PLOT
1
SET VERTICAL SCALE? Y OR N (=BLANK)
Y
WITH MAX Y VALUES
-5 5
SET HORIZONTAL SCALE? Y OR N (=BLANK)
Y
WITH MAX X VALUES
-10 10
SKIP?

DOUBLE=1/25=WIDTH
NORMALIZED XZ STRAIN



```
0=RE-START, 1=NEW PLOT
2!
WARNING--CAUTION--WARNING
NORMALIZED DISPLACEMENTS MAY BE INCORRECT
CHECK DBLNRM DOCUMENTATION FOR EXPLANATION
MIN/MAX VALUES OF NORMALIZED W(X) ARE
     0.                 2.0913E+00
SET VERTICAL SCALE? Y OR N=BLANK)
Y!
MIN/MAX Y VALUES
0 5!
SET HORIZONTAL SCALE? Y OR N=BLANK)
Y!
MIN/MAX X VALUES
-10 10!
SKIP?
```

```
 1
0=RE-START, 1=NEW PLOT
2 1
MU1 =              3.0000E+11      MU2 =              3.0000E+10
MU3 =              1.0000E+11        M =              5.3846E-01
  K =              8.1818E-01
DISLOCATION POSITION = -          .500, -      1.000 KMS.
WIDTH OF SLIP ZONE   =         25.000
SLAB THICKNESS       =          1.000 KMS.
BURGERS VECTOR       =         -1.000 MMS.
X2 STRESS/STRAIN VS. X
0=RE-START
1=STOP
2=RE-START WITH SAME VALUES, DIFFERENT PROFILE
1 1
```

LISTING OF ST3PHS

```
  1  DELETE(LGO,OUTPUT,ST3PHS)
  2  ST3PHS.
  3  CXIT.
  4  LIBCOPY(GRAPHIC,TXLGO/RR,TXLGO)
  5  LIBCOPY(JORAT,NPLGO/RR,NPLGO)
  6  DELETE(LGO,OUTPUT,ST3PHS)
  7  RUN76(S)
  8  LINK(F=LGO,F=TXLGO,F=NPLGO,B=ST3PHS)
  9  ST3PHS.
 10  FIN.
 11  EOR
 12        PROGRAM ST3PHS(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
 13        COMMON/TVPOOL/TVPUL(8)
 14        COMMON/TVTUNE/ITUNE(30)
 15        COMMON/JPLOT/XLT,XRT,YLO,YUP,MAJX,MAJY,KX(2),KY(2),
 16       1LTITL(8),LU,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
 17        DIMENSION IFET(8)
 18        DIMENSION X(500),Y(500),SXZ(500),SYZ(500),AB(30)
 19        DIMENSION PLOTY(500),PLOTX(500)
 20        DATA YMIN,YMAX,XCUT,CUTMAX/-1.E+10,-1.E+10,-1.E+10,-1.E+10/
 21        CALL FET(SLTAPE7,IFET,8)
 22        IFET(2)=IFET(2).OR.0000 0010 0000 0000 0000B
 23        IFET(8)=IFET(8).OR.4000 0000 0000 0000 0000B
 24        CALL FET(SLTAPE7,IFET,-8)
 25    48  WRITE(7,15)
 26    15  FORMAT(*SHEAR MODULI--IN CGS UNITS--MU1,MU2,MU3*)
 27        CALL GETNUM(AB) $IAB2=AB(2) $IAB4=AB(4) $IAB6=AB(6)
 28        AMU1=AB(1)*(10.**IAB2) $AMU2=AB(3)*(10.**IAB4)
 29        AMU3=AB(5)*(10.**IAB6)
 30        WRITE(7,16)
 31    16  FORMAT(*POSITION OF DISLOCATION--IN KMS., (A,LAMBDA)*,/,
 32       1          *NOTE---A>0., LAMBDA>0.*)
 33        CALL GETNUM(AB) $A=AB(1) $ALAMBD=AB(2)
 34        WRITE(7,17)
 35    17  FORMAT(*THICKNESS OF SLAB, IN KMS.--D*)
 36        CALL GETNUM(AB) $D=AB(1)  $WRITE(7,18)
 37    18  FORMAT(*BURGERS VECTOR, IN MMS.--B*,/,
 38       1       *B<0 FOR RIGHT LATERAL SLIP*)
 39        CALL GETNUM(AB) $B=-AB(1)*1.E-06
 40    70  WRITE(7,19)
 41    19  FORMAT(*1=AMPLITUDE VS. X, 2=AMPLITUDE VS. Y*)
 42        CALL GETNUM(AB) $KFLAG=AB(1)$WRITE(7,20)
 43    20  FORMAT(*1=XZ STRESS, 2=YZ STRESS*)
 44        CALL GETNUM(AB)  $IFLAG=AB(1)
 45        IF(KFLAG.EQ.2)GOTO201
 46        WRITE(7,21) $CALL GETNUM(AB) $XMIN=AB(1)
 47        XMAX=AB(2) $ WRITE(7,22)  $CALL GETNUM(AB) $SET=-AB(1)
 48        GO TO 202
 49    201 WRITE(7,23) $CALL GETNUM(AB)
 50        YMAX=-AB(1) $YMIN=-AB(2) $WRITE(7,24)
 51        CALL GETNUM(AB) $SET=AB(1)
 52    21  FORMAT(*ENTER MINIMUM AND MAXIMUM X VALUES, IN KMS.*)
 53   22   FORMAT(*DEPTH OF PROFILE, IN KMS.*)
 54    23  FORMAT(*ENTER MINIMUM AND MAXIMUM Y VALUES, IN KMS.*)
 55     24 FORMAT(*X COORDINATE OF PROFILE, IN KMS.*)
 56    202 NUM=200 $IUPPER=20   $PI=3.14159
 57        IF(KFLAG.EQ.1)XINC=(XMAX-XMIN)/NUM
 58        IF(KFLAG.EQ.2)YINC=(YMAX-YMIN)/NUM
```

```
      WRITE(7,25) NUM,IUPPER
   25 FORMAT(*NOTE--NUMBER OF DATA POINTS = *,I5,/,
     1       *      UPPER LIMIT ON SERIES = *,I5)
      AM=(AMU3-AMU2)/(AMU3+AMU2)$IF(AM.EQ.0.)AM=1.E-20
      AK=(AMU1-AMU2)/(AMU1+AMU2) $IF(AK.EQ.0.)AK=1.E-20
      BOUND=-D $AL=-ALAMBD $AR=-A
      IF(KFLAG.EQ.1 .AND.AL.GT.SET)GOTO74
      GOTO75
   74 WRITE(7,76)
   76 FORMAT(*WARNING---PROFILE BELOW DISLOCATION LINE*,/,
     1       *SOLUTION INVALID, TRY AGAIN*)
      GOTO63
   75 IF(KFLAG.EQ.1)GOTO90
      RDIST=ABS(SET-(-A))
      IF(RDIST.LT.YINC)WRITE(7,91)
      IF(RDIST.LT.YINC)GOTO63
      GOTO92
   90 RDIST=ABS(SET-(-ALAMBD))
      IF(RDIST.LT.XINC)WRITE(7,91)
      IF(RDIST.LT.XINC)GOTO63
   91 FORMAT(*WARNING---PROFILE TOO CLOSE TO DISLOCATION CORE,*,/,
     1       *CHOOSE NEW PROFILE OR INCREASE GRID POINT DENSITY*)
   92 CONTINUE
      IF(KFLAG.EQ.1.AND.SET.GT.0.)WRITE(7,93)
      IF(KFLAG.EQ.1.AND.SET.GT.0.)GOTO63
   93 FORMAT(*WARNING---PROFILE ABOVE SURFACE OF HALF-SPACE,*,/,
     1       *SOLUTION INVALID, TRY AGAIN*)
      DO 200 J=1,NUM
      IF(KFLAG.EQ.1)X(J)=XMIN+(XINC*(J-1))
      IF(KFLAG.EQ.1) Y(J)=SET
      IF(KFLAG.EQ.2)X(J)=SET
      IF(KFLAG.EQ.2)Y(J)=YMIN+(YINC*(J-1))
      IF(X(J).EQ.AR.AND.Y(J).EQ.AL)GOTO32
      GO TO 33
   32 X(J)=X(J)+1.E-10 $Y(J)=Y(J)+1.E-10
   33 IF(X(J) .LT.BOUND)GOTO103
      IF(X(J).GE.BOUND.AND.X(J).LE.0.)GOTO102
  101 IF(IFLAG .EQ. 2)GOTO12
      SUM1=SUM2=0.
      DO 1 I=1,IUPPER
      NU=I-1
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
      TERM1=((AM*AK)**NU)*F1
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,I,D,ALAMBD,F2)
      TERM2=  (AM**I)*(AK**NU)  *F2
      SUM1=SUM1+TERM1     $SUM2=SUM2+TERM2
    1 CONTINUE
      SXZ(J)=(((-AMU1*B)/(2.*PI))*(1.-AK))*(SUM1+SUM2) $GOTO200
   12 SUM1=SUM2=SUM3=SUM4=0.
      DO 5 I=1,IUPPER
      NU=I-1
      CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
      CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMBD,F2)
      CALL FUNC1(X(J),-A,I,D,Y(J),ALAMBD,F3)
      CALL FUNC1(X(J),-A,I,D,Y(J),-ALAMBD,F4)
      TERM1=((AM*AK)**NU)*F1 $TERM2=((AM*AK)**NU)*F2
      TERM3=(AM**I)*(AK**NU)*F3 $TERM4=(AM**I)*(AK**NU)*F4
      SUM1=SUM1+TERM1 $SUM2=SUM2+TERM2 $SUM3=SUM3+TERM3
```

```
117         SUM4=SUM4+TERM4
118       5 CONTINUE
119         SYZ(J)=((AMU1*B)*(1.-AK)/(2.*PI))*(SUM1+SUM3-SUM2-SUM4)
120         GOTO200
121       102 IF(IFLAG.EQ.2)GOTO13
122         SUM1=0.
123         DO 7 NU=1,IUPPER
124         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
125         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMED,F2)
126         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,NU,D,ALAMED,F3)
127         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMED,F4)
128         TERM=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4) $SUM1=SUM1+TERM
129       7 CONTINUE
130         CALL FUNC3(Y(J),ALAMBD,X(J),A,F1)
131         CALL FUNC3(Y(J),ALAMBD,X(J),-A,F2)
132         TERM2=F1+(AK*F2)
133         SXZ(J)=((-AMU2*B)/(2.*PI))*(TERM2+SUM1) $GOTO200
134       13 SUM1=SUM2=0.
135         DO 3 NU=1,IUPPER
136         CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
137         CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMED,F2)
138         CALL FUNC1(X(J),-A,NU,D,Y(J),ALAMBD,F3)
139         CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMED,F4)
140         TERM1=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4)
141         SUM1=SUM1+TERM1
142         CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMBD,F5)
143         CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F6)
144         CALL FUNC1(X(J),-A,NU,D,Y(J),-ALAMED,F7)
145         CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMED,F8)
146         TERM2=((AM*AK)**NU)*(F5+(AK*F6)+((1./AK)*F7)+F8)
147         SUM2=SUM2+TERM2
148       3 CONTINUE
149         CALL FUNC3(X(J),A,Y(J),ALAMBD,F1)
150         CALL FUNC3(X(J),-A,Y(J),ALAMBD,F2)
151         CALL FUNC3(X(J),A,Y(J),-ALAMBD,F3)
152         CALL FUNC3(X(J),-A,Y(J),-ALAMBD,F4)
153         SYZ(J)=((AMU2*B)/(2.*PI))*((F1+(AK*F2)+SUM1)-(F3+(AK*F4)+SUM2))
154         GO TO 200
155       103 IF(IFLAG.EQ.2)GOTO14
156         SUM1=0.
157         DO 9 I=1,IUPPER
158         NU=I-1
159         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMBD,F1)
160         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMED,F2)
161         TERM=(F1+(AK*F2))*((AK*AM)**NU) $SUM1=SUM1+TERM
162       9 CONTINUE
163         SXZ(J)=((-AMU3*B)/(2.*PI))*(1.-AM)*SUM1 $GOTO200
164       14 SUM1=SUM2=0.
165         DO 11 I=1,IUPPER
166         NU=I-1
167         CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMBD,F1)
168         CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMED,F2)
169         TERM1=((AM*AK)**NU)*(F1+(AK*F2))
170         SUM1=SUM1+TERM1
171         CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMED,F1)
172         CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F2)
173         TERM2=((AM*AK)**NU)*(F1+(AK*F2)) $SUM2=SUM2+TERM2
174       11 CONTINUE
```

```
175         SYZ(J)=((AMU3*B)/(2.*PI))*(1.-AM)*(SUM1-SUM2)
176      200 CONTINUE
177         IF(IFLAG.EQ.1)STSMID=SXZ(1) $IF(IFLAG.EQ.2)STSMID=SYZ(1)
178         SUP=SDN=STSMID
179         DO 26 I=1,NUM
180         IF(IFLAG.EQ.1)GOTO27
181         IF(SYZ(I).GT.SUP)SUP=SYZ(I)
182         IF(SYZ(I).LT.SDN)SDN=SYZ(I)
183         GO TO 26
184       27 IF(SXZ(I).GT.SUP)SUP=SXZ(I)
185         IF(SXZ(I).LT. SDN)SDN=SXZ(I)
186       26 CONTINUE
187         IF(IFLAG.EQ.1)WRITE(7,28) $IF(IFLAG.EQ.2)WRITE(7,29)
188      28  FORMAT(*MIN/MAX VALUES OF XZ STRESS--UNITS ARE DYNES/(CM2)*)
189       29 FORMAT(*MIN/MAX VALUES OF YZ STRESS--UNITS ARE DYNES/(CM2)*)
190         WRITE(7,30)SDN,SUP
191       30 FORMAT(E15.4,5X,E15.4)
192         LUP=0
193         DO 31 I=1,NUM
194         IF(IFLAG.EQ.1)PLOTY(I)=SXZ(I)
195         IF(IFLAG.EQ.2)PLOTY(I)=SYZ(I)
196         IF(KFLAG.EQ.1)PLOTX(I)=X(I)
197       31 IF(KFLAG.EQ.2)PLOTX(I)=Y(I)
198         IF(KFLAG.EQ.2)GOTO83
199         GOTO85
200       83 DO 81 I=1,NUM
201         SXZ(I)=PLOTY(I)
202      81  SYZ(I)=PLOTX(I)
203         DO 84 I=1,NUM
204         L=NUM-I+1
205         PLOTY(L)=SXZ(I)
206       84 PLOTX(L)=SYZ(I)
207         DO 82 I=1,NUM
208         IF(KFLAG.EQ.2.AND.PLOTX(I).LT.AL)LUP=LUP+1
209       82 IF(LUP.EQ.1)XCUT=PLOTX(I-1)
210         LDN=0
211         DO 86 I=1,NUM
212         L=NUM-I+1
213         IF(KFLAG.EQ.2.AND.PLOTX(L).GT.0.)LDN=LDN+1
214       86 IF(LDN.EQ.1)CUTMAX=PLOTX(L-1)
215       85 CONTINUE
216      417 LU=7 $LNLGX=1 $LNLGY=1 $NCLX=2 $NCLY=2 $WRITE(7,34)
217       34 FORMAT(*PLOT TITLE, 80 CHARACTERS MAXIMUM*)
218         READ(7,35)(LTITL(JM),JM=1,8)
219       35 FORMAT(8A10)
220         IF(IFLAG.EQ.1)LTITL2(1)=10HXZ STRESS
221         IF(IFLAG.EQ.2)LTITL2(1)=10HYZ STRESS
222         DO 36 I=2,8
223      36  LTITL2(I)=10H
224         IF(KFLAG.EQ.1)KX(1)=10HX (KMS.)
225         IF(KFLAG.EQ.2)KX(1)=10HY (KMS.)
226         KX(2)=10H              $KY(1)=10HSTRESS       $KY(2)=10H
227         XRT=PLOTX(NUM) $XLT=PLOTX(1) $MAJX=5 $MAJY=10 $YLO=SDN $YUP=SUP
228         IF(KFLAG.EQ.2.AND.YMIN.LT.AL)XRT=XCUT
229         IF(KFLAG.EQ.2.AND.YMAX.GT.0.)XLT=CUTMAX
230         IF(KFLAG.EQ.2.AND.YMIN.GE.AL)XRT=PLOTX(NUM)
231         AA=YUP$IF(YLO.EQ.AA)YUP=YUP+1.$IF(YLO.EQ.AA)YLO=YLO-1.
232         WRITE(7,37)
```

```
233     37  FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
234         READ(7,38)CHARAC
235     38  FORMAT(A1)
236         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO39
237         WRITE(7,40)
238      40 FORMAT(*MIN/MAX Y VALUES*)
239         CALL GETNUM(AB) $YLO=AB(1) $YUP=AB(2)
240     39  WRITE(7,41)
241      41 FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
242         READ(7,38)CHARAC
243         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO42
244         WRITE(7,43)
245      43 FORMAT(*MIN/MAX X VALUES*)
246         CALL GETNUM(AB) $IF(KFLAG.EQ.1)XLT=AB(1)
247         IF(KFLAG.EQ.1)XRT=AB(2)
248         IF(KFLAG.EQ.2)XLT=-AB(2)
249         IF(KFLAG.EQ.2)XRT=-AB(1)
250      42 WRITE(7,44)
251      44 FORMAT(*SKIP?*)
252         READ(7,38)CHARAC
253         IF(CHARAC.EQ.1HN .OR. CHARAC.EQ.1H )CALL PLOTS(PLOTY,PLOTX,1,NUM)
254         WRITE(7,46)
255     46  FORMAT(*0=RE-START, 1=NEW PLOT*)
256         READ(7,71)LCHECK
257     71  FORMAT(I1)
258         IF(LCHECK.EQ.0)GOTO63$IF(LCHECK.EQ.1)GOTO417
259         DO 50 I=1,NUM
260         IF(X(I).LT.BOUND)AMU=AMU3
261         IF(X(I).GE.BOUND.AND.X(I).LE.0.)AMU=AMU2
262         IF(X(I).GT.0.)AMU=AMU1
263         IF(AMU.EQ.0.)AMU=1.E-10
264      50 PLOTY(I)=(PLOTY(I)/(AMU))*1.E+06
265         STSMID=PLOTY(1) $SUP=SDN=STSMID
266         DO 51 I=1,NUM
267         IF(PLOTY(I).GT.SUP)SUP=PLOTY(I)
268     51  IF(PLOTY(I).LT.SDN)SDN=PLOTY(I)
269         IF(IFLAG.EQ.1)WRITE(7,52) $IF(IFLAG.EQ.2)WRITE(7,53)
270     52  FORMAT(*MIN/MAX VALUES OF XZ MICROSTRAIN*)
271     53  FORMAT(*MIN/MAX VALUES OF YZ MICROSTRAIN*)
272         WRITE(7,30)SDN,SUP
273         IF(IFLAG.EQ.1)LTITL2(1)=10HXZ STRAIN
274         IF(IFLAG.EQ.2)LTITL2(1)=10HYZ STRAIN
275     418 KY(1)=10HMICROSTRAI          $KY(2)=10HN
276         YLO=SDN $YUP=SUP
277         AA=YUP $IF(YLO.EQ.AA)YUP=YUP+1. $IF(YLO.EQ.AA)YLO=YLO-1.
278         WRITE(7,37)  $READ(7,38)CHARAC
279         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO55
280         WRITE(7,40) $CALL GETNUM(AB) $YLO=AB(1) $YUP=AB(2)
281     55  WRITE(7,41)$READ(7,38)CHARAC
282         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO301
283         WRITE(7,43) $CALL GETNUM(AB)
284         IF(KFLAG.EQ.1)XLT=AB(1) $IF(KFLAG.EQ.1)XRT=AB(2)
285         IF(KFLAG.EQ.2)XLT=-AB(2) $IF(KFLAG.EQ.2)XRT=-AB(1)
286     301 WRITE(7,44) $READ(7,38)CHARAC
287         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )CALL PLOTS(PLOTY,PLOTX,1,NUM)
288      56 WRITE(7,46) $CALL GETNUM(AB)$LCHECK=AB(1)
289         IF(LCHECK.EQ.0)GOTO63$IF(LCHECK.EQ.1)GOTO418
290         IF(KFLAG.NE.IFLAG)WRITE(7,57)
```

```
291     57  FORMAT(*NOTE--ONLY W(X) VS. X OR W(Y) VS. Y IS COMPUTED*)
292         IF(KFLAG.NE.JFLAG)GOTO63
293         IF(KFLAG.EQ.1)DINC=XINC*1.E+06
294         IF(KFLAG.EQ.2)DINC=YINC*1.E+06 $SUM=0.
295         DO 58 I=1,NUM
296         TERM=2.*PLCTY(I)*DINC*1.E-06 $SUM=SUM+TERM
297      58 PLOTY(I)=SUM
298         MCHECK=0
299         DO 789 M=1,NUM
300         IF(KFLAG.EQ.1)GRD=ABS((XMAX-XMIN)/NUM)
301         IF(KFLAG.EQ.2)GRD=ABS((YMAX-YMIN)/NUM)
302         IF(KFLAG.EQ.1) GOTO790
303         GRD1=-ALAMED+(GRD/100.) $GRD2=-ALAMBD-(GRD/100.)
304         IF(Y(M).GT.GRD2.AND.Y(M).LT.GRD1)KMID=M
305         IF(Y(M).GT.GRD2.AND.Y(M).LT.GRD1)MCHECK=1
306         GOTO789
307     790 GRD1=-A+(GRD/100.)  $GRD2=-A-(GRD/100.)
308         IF(X(M).GT.GRD2.AND.X(M).LT.GRD1)KMID=M
309         IF(X(M).GT.GRD2.AND.X(M).LT.GRD1)MCHECK=1
310     789 CONTINUE
311         IF(MCHECK.EQ.0)STSMID=((PLCTY(NUM)-PLCTY(1))/2.)+PLOTY(1)
312         IF(MCHECK.EQ.1)STSMID=PLCTY(KMID)
313         DO 73 I=1,NUM
314      73 PLOTY(I)=PLOTY(I)-STSMID
315         STSMID=PLOTY(1) $SUP=SDN=STSMID
316         DO 59 I=1,NUM
317         IF(PLOTY(I).GT.SUP)SUP=PLCTY(I)
318     59  IF(PLCTY(I).LT.SDN)SDN=PLCTY(I)
319         IF(IFLAG.EQ.1) WRITE(7,60) $IF(IFLAG.EQ.2)WRITE(7,61)
320         IF(KFLAG.EQ.1.AND.SET.NE.0.)WRITE(7,87)
321     87  FORMAT(*CAUTION---THIS IS NOT THE TOTAL DISPLACEMENT*)
322      60 FORMAT(*MIN/MAX VALUES OF W(X) VS. X*)
323      61 FORMAT(*MIN/MAX VALUES OF W(Y) VS. Y*)
324         WRITE(7,30) SDN,SUP
325     419 IF(IFLAG.EQ.1)LTITL2(1)=10HW(X) VS. X
326         IF(IFLAG.EQ.2)LTITL2(1)=10HW(Y) VS. Y
327         KY(1)=10HDISPLACEME       $KY(2)=10HNT-MMS.
328         YLO=SDN  $YUP=SUP$AA=YUP$IF(YLO.EQ.AA)YUP=YUP+1.
329         IF(YLO.EQ.AA)YLO=YLO-1.
330         WRITE(7,37) $READ(7,38)CHARAC
331         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO62
332         WRITE(7,40) $CALL GETNUM(AE) $YLO=AB(1) $YUP=AB(2)
333      62 WRITE(7,41) $READ(7,38)CHARAC
334         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO303
335         WRITE(7,43) $CALL GETNUM(AE)
336         IF(KFLAG.EQ.1)XLT=AB(1) $IF(KFLAG.EQ.1)XRT=AB(2)
337         IF(KFLAG.EQ.2)XLT=-AB(2)  $IF(KFLAG.EQ.2)XRT=-AB(1)
338     303 WRITE(7,44) $READ(7,38)CHARAC
339         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H ) CALL PLOTS(PLCTY,PLOTX,1,NUM)
340         WRITE(7,46)$CALL GETNUM(AE)$LCHECK=AB(1)$IF(LCHECK.EQ.1)GOTO419
341     63  WRITE(7,45)AMU1,AMU2,AML3,AM,AK
342         BB=B*1.E+06
343         WRITE(7,47)A,ALAMBD,D,BB
344         IF(IFLAG.EQ.1)CD1=2HXZ$IF(KFLAG.EQ.1)CD2=1HX
345         IF(IFLAG.EQ.2)CD1=2HYZ$IF(KFLAG.EQ.2)CD2=1HY
346         WRITE(7,49)CD1,CD2
347      45 FORMAT(*MU1 = *,E20.4,5X,*MU2 = *,E20.4,/,
348        1       *MU3 = *,E20.4,5X,*  M = *,E20.4,/,
```

```
     1       *  K = *,E20.4)
47   FORMAT(*DISLOCATION POSITION = -*,F10.3,*, -*,F10.3,* KMS.*,/,
     1       *SLAB THICKNESS      =  *,F10.3,* KMS.*,/,
     1       *BURGERS VECTOR      =  *,F10.3,* MMS.*)
49   FORMAT(A2,* STRESS/STRAIN \S. *,A2)
     WRITE(7,64)
64   FORMAT(*0=RE-START*,/,*1=STOP*,/,
     1       *2=RE-START WITH SAME VALUES, DIFFERENT PROFILE*)
     CALL GETNUM(AB) $LJK=AB(1)
     IF(LJK.EQ.0)GOTO48 $IF(LJK.EQ.2)GO TO 70
     STOP $END
     SUBROUTINE FUNC1(P,A,NU,D,G,ALAMBD,F)
     R=(((P+A+(2.*NU*D))**2)+((G+ALAMBD)**2))
     IF(R.EQ.0.)R=1.E-10$F=(P+A+(2.*NU*D))/R
     RETURN $END
     SUBROUTINE FUNC2(P,Q,R,S,A,NU,D,ALAMBD,F)
     T=(((R+A+(2.*NU*D))**2)+((S+ALAMBD)**2))
     IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
     RETURN $END
     SUBROUTINE FUNC3(P,Q,R,S,F)
     T=(((R+S)**2)+((P+Q)**2))
     IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
     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  $GOTO5
   2 M=-1 $S=4 $GOTO5
   3 P=I $S=4 $GOTO5
   4 IF(P.NE.0)F=10.**(I-P-1) $R(J)=N/F*M $GOTO6
   9 FORMAT(80F1)
     END
```

LISTING OF ST3PSZ

```
04 MAY 76    09.12.23        MCHUGH .ST3PSZ

  1  DELETE(LGO,OUTPUT,ST3PSZ)
  2  ST3PSZ.
  3  CXIT.
  4  LIBCOPY(GRAPHIC,TXLGO/RR,TXLGO)
  5  LIBCOPY(JDRAT,NPLGO/RR,NPLGO)
  6  DELETE(LGO,OUTPUT,ST3PSZ)
  7  RUN76(S)
  8  LINK(F=LGO,F=TXLGO,F=NPLGO,B=ST3PSZ)
  9  ST3PSZ.
 10  FIN.
 11  EOR
 12        PROGRAM ST3PSZ(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
 13        COMMON/TVPCOL/TVPUL(8)
 14        COMMON/TVTUNE/ITUNE(30)
 15        COMMON/JPLOT/XLT,XRT,YLO,YLP,MAJX,MAJY,KX(2),KY(2),
 16       1LTITL(8),LO,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
 17        DIMENSION IFET(8)
 18        DIMENSION X(200),Y(200),SXZ(200),SYZ(200),AB(30)
 19        DIMENSION PLOTY(200),PLOTX(200)
 20        DIMENSION SXZSLP(200,2),SYZSLP(200,2)
 21        DATA YMIN,YMAX,XCUT,CUTMAX/-1.E+10,-1.E+10,-1.E+10,-1.E+10/
 22        CALL FET(5LTAPE7,IFET,8)
 23        IFET(2)=IFET(2).OR.0000 0010 0000 0000 0000B
 24        IFET(8)=IFET(8).OR.4000 0000 0000 0000 0000B
 25        CALL FET(5LTAPE7,IFET,-8)
 26    48  WRITE(7,15)
 27    15  FORMAT(*SHEAR MODULI--IN CGS UNITS--MU1,MU2,MU3*)
 28        CALL GETNUM(AB) $IAB2=AB(2) $IAB4=AB(4) $IAB6=AB(6)
 29        AMU1=AB(1)*(10.**IAB2) $AMU2=AB(3)*(10.**IAB4)
 30        AMU3=AB(5)*(10.**IAB6)
 31        WRITE(7,16)
 32    16  FORMAT(*POSITION OF DISLOCATION--IN KMS., (A,LAMBDA)*,/,
 33       1          *NOTE---A>0., LAMBDA>0.*)
 34        CALL GETNUM(AB) $A=AB(1) $ALAMBD=AB(2) $WRITE(7,17)
 35    17  FORMAT(*THICKNESS OF SLAB, IN KMS.--D*)
 36        CALL GETNUM(AB) $D=AB(1)  $WRITE(7,900)
 37    900 FORMAT(*WIDTH OF SLIP PLANE--KMS.*)
 38        CALL GETNUM(AB)$WIDTH=AB(1)$WRITE(7,18)
 39    18  FORMAT(*BURGERS VECTOR, IN MMS.--B*,/,
 40       1       *B<0 FOR RIGHT LATERAL SLIP*)
 41        CALL GETNUM(AB) $B=-AB(1)*1.E-06
 42    70  WRITE(7,19)
 43    19  FORMAT(*1=AMPLITUDE VS. X, 2=AMPLITUDE VS. Y*)
 44        CALL GETNUM(AB) $KFLAG=AB(1)$WRITE(7,20)
 45    20  FORMAT(*1=XZ STRESS, 2=YZ STRESS*)
 46        CALL GETNUM(AB)  $IFLAG=AB(1)
 47        IF(KFLAG.EQ.2)GOTO201
 48        WRITE(7,21) $CALL GETNUM(AB) $XMIN=AB(1)
 49        XMAX=AB(2) $ WRITE(7,22)  $CALL GETNUM(AB) $SET=-AB(1)
 50        GO TO 202
 51    201 WRITE(7,23) $CALL GETNUM(AB)
 52        YMAX=-AB(1) $YMIN=-AB(2) $WRITE(7,24)
 53        CALL GETNUM(AB) $SET=AB(1)
 54    21  FORMAT(*ENTER MINIMUM AND MAXIMUM X VALUES, IN KMS.*)
 55    22  FORMAT(*DEPTH OF PROFILE, IN KMS.*)
 56    23  FORMAT(*ENTER MINIMUM AND MAXIMUM Y VALUES, IN KMS.*)
 57    24  FORMAT(*X COORDINATE OF PROFILE, IN KMS.*)
 58    202 NUM=200 $IUPPER=20 $PI=3.14159
```

```
      IF(KFLAG.EQ.1)XINC=(XMAX-XMIN)/NUM
      IF(KFLAG.EQ.2)YINC=(YMAX-YMIN)/NUM
      WRITE(7,25) NUM,IUPPER
   25 FORMAT(*NOTE--NUMBER OF DATA POINTS = *,I5,/,
     1       *      UPPER LIMIT ON SERIES = *,I5)
      AM=(AMU3-AMU2)/(AMU3+AMU2)$IF(AM.EQ.0.)AM=1.E-20
      AK=(AMU1-AMU2)/(AMU1+AMU2) $IF(AK.EQ.0.)AK=1.E-20
      BOUND=-D $AL=-ALAMBD $AR=-A
      IF(KFLAG.EQ.1 .AND.AL.GT.SET)GOTO74
      GOTO75
   74 WRITE(7,76)
   76 FORMAT(*WARNING---PROFILE BELOW DISLOCATION LINE*,/,
     1       *SOLUTION INVALID, TRY AGAIN*)
      GOTO63
   75 IF(KFLAG.EQ.1)GOTO90
      RDIST=ABS(SET-(-A))
      IF(RDIST.LT.YINC)WRITE(7,91)
      IF(RDIST.LT.YINC)GOTO63
      GOTO92
   90 RDIST=ABS(SET-(-ALAMBD))
      IF(RDIST.LT.XINC)WRITE(7,91)
      IF(RDIST.LT.XINC)GOTO63
   91 FORMAT(*WARNING---PROFILE TOO CLOSE TO DISLOCATION CORE,*,/,
     1       *CHOOSE NEW PROFILE OR INCREASE GRID POINT DENSITY*)
   92 CONTINUE
      IF(KFLAG.EQ.1.AND.SET.GT.0.)WRITE(7,93)
      IF(KFLAG.EQ.1.AND.SET.GT.0.)GOTO63
   93 FORMAT(*WARNING---PROFILE ABOVE SURFACE OF HALF-SPACE,*,/,
     1       *SOLUTION INVALID, TRY AGAIN*)
      DO 910 MK=1,2
      IF(MK.EQ.2)ALAMBD=ALAMBD+WIDTH
      IF(MK.EQ.2)B=-B
      DO 200 J=1,NUM
      IF(KFLAG.EQ.1)X(J)=XMIN+(XINC*(J-1))
      IF(KFLAG.EQ.1) Y(J)=SET
      IF(KFLAG.EQ.2)X(J)=SET
      IF(KFLAG.EQ.2)Y(J)=YMIN+(YINC*(J-1))
      IF(X(J).EQ.AR.AND.Y(J).EQ.AL)GOTO32
      GO TO 33
   32 X(J)=X(J)+1.E-10 $Y(J)=Y(J)+1.E-10
   33 IF(X(J) .LT.BOUND)GOTO103
      IF(X(J).GE.BOUND.AND.X(J).LE.0.)GOTO102
  101 IF(IFLAG .EQ. 2)GOTO12
      SUM1=SUM2=0.
      DO 1 I=1,IUPPER
      NU=I-1
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
      TERM1=((AM*AK)**NU)*F1
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,I,D,ALAMBD,F2)
      TERM2=  (AM**I)*(AK**NU)  *F2
      SUM1=SUM1+TERM1     $SUM2=SUM2+TERM2
    1 CONTINUE
      SXZ(J)=(((-AMU1*B)/(2.*PI))*(1.-AK))*(SUM1+SUM2) $GOTO200
   12 SUM1=SUM2=SUM3=SUM4=0.
      DO 5 I=1,IUPPER
      NU=I-1
      CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
      CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMBD,F2)
```

```
117         CALL FUNC1(X(J),-A,I,D,Y(J),ALAMBD,F3)
118         CALL FUNC1(X(J),-A,I,D,Y(J),-ALAMBD,F4)
119         TERM1=((AM*AK)**NU)*F1 $TERM2=((AM*AK)**NU)*F2
120         TERM3=(AM**I)*(AK**NU)*F3 $TERM4=(AM**I)*(AK**NU)*F4
121         SUM1=SUM1+TERM1 $SUM2=SUM2+TERM2 $SUM3=SUM3+TERM3
122         SUM4=SUM4+TERM4
123       5 CONTINUE
124         SYZ(J)=((AMU1*B)*(1.-AK)/(2.*PI))*(SUM1+SUM3-SUM2-SUM4)
125         GOTO200
126     102 IF(IFLAG.EQ.2)GOTO13
127         SUM1=0.
128         DO 7 NU=1,IUPPER
129         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
130         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMBD,F2)
131         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,NU,D,ALAMBD,F3)
132         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMBD,F4)
133         TERM=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4) $SUM1=SUM1+TERM
134       7 CONTINUE
135         CALL FUNC3(Y(J),ALAMBD,X(J),A,F1)
136         CALL FUNC3(Y(J),ALAMBD,X(J),-A,F2)
137         TERM2=F1+(AK*F2)
138         SXZ(J)=((-AMU2*B)/(2.*PI))*(TERM2+SUM1) $GOTO200
139      13 SUM1=SUM2=0.
140         DO 3 NU=1,IUPPER
141         CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
142         CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMBD,F2)
143         CALL FUNC1(X(J),-A,NU,D,Y(J),ALAMBD,F3)
144         CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMBD,F4)
145         TERM1=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4)
146         SUM1=SUM1+TERM1
147         CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMBD,F5)
148         CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F6)
149         CALL FUNC1(X(J),-A,NU,D,Y(J),-ALAMBD,F7)
150         CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMBD,F8)
151         TERM2=((AM*AK)**NU)*(F5+(AK*F6)+((1./AK)*F7)+F8)
152         SUM2=SUM2+TERM2
153       3 CONTINUE
154         CALL FUNC3(X(J),A,Y(J),ALAMBD,F1)
155         CALL FUNC3(X(J),-A,Y(J),ALAMBD,F2)
156         CALL FUNC3(X(J),A,Y(J),-ALAMBD,F3)
157         CALL FUNC3(X(J),-A,Y(J),-ALAMBD,F4)
158         SYZ(J)=((AMU2*B)/(2.*PI))*((F1+(AK*F2)+SUM1)-(F3+(AK*F4)+SUM2))
159         GO TO 200
160     103 IF(IFLAG.EQ.2)GOTO14
161         SUM1=0.
162         DO 9 I=1,IUPPER
163         NU=I-1
164         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMBD,F1)
165         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMBD,F2)
166         TERM=(F1+(AK*F2))*((AK*AM)**NU) $SUM1=SUM1+TERM
167       9 CONTINUE
168         SXZ(J)=((-AMU3*B)/(2.*PI))*(1.-AM)*SUM1 $GOTO200
169      14 SUM1=SUM2=0.
170         DO 11 I=1,IUPPER
171         NU=I-1
172         CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMBD,F1)
173         CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMBD,F2)
174         TERM1=((AM*AK)**NU)*(F1+(AK*F2))
```

```
175         SUM1=SUM1+TERM1
176         CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMED,F1)
177         CALL FUNC1(X(J),-A,-NU,C,Y(J),-ALAMBD,F2)
178         TERM2=((AM*AK)**NU)*(F1+(AK*F2)) $SUM2=SUM2+TERM2
179      11 CONTINUE
180         SYZ(J)=((AMU3*B)/(2.*PI))*(1.-AM)*(SUM1-SUM2)
181     200 CONTINUE
182         DO 920 IMK=1,NUM
183         IF(IFLAG.EQ.1)SXZSLP(IMK,MK)=SXZ(IMK)
184     920 IF(IFLAG.EQ.2)SYZSLP(IMK,MK)=SYZ(IMK)
185     910 CONTINUE
186         DO 930 I=1,NUM
187         IF(IFLAG.EQ.1)SXZ(I)=SXZSLP(I,1)+SXZSLP(I,2)
188     930 IF(IFLAG.EQ.2)SYZ(I)=SYZSLP(I,1)+SYZSLP(I,2)
189         IF(IFLAG.EQ.1)STSMID=(SXZ(NUM)-SXZ(1))/2.
190         IF(IFLAG.EQ.2)STSMID=(SYZ(NUM)-SYZ(1))/2.
191         SUP=SDN=STSMID
192         DO 26 I=1,NUM
193         IF(IFLAG.EQ.1)GOTO27
194         IF(SYZ(I).GT.SUP)SUP=SYZ(I)
195         IF(SYZ(I).LT.SDN)SDN=SYZ(I)
196         GO TO 26
197      27 IF(SXZ(I).GT.SUP)SUP=SXZ(I)
198         IF(SXZ(I).LT. SDN)SDN=SXZ(I)
199      26 CONTINUE
200         IF(IFLAG.EQ.1)WRITE(7,28) $IF(IFLAG.EQ.2)WRITE(7,29)
201     28  FORMAT(*MIN/MAX VALUES OF XZ STRESS--UNITS ARE DYNES/(CM2)*)
202      29 FORMAT(*MIN/MAX VALUES OF YZ STRESS--UNITS ARE DYNES/(CM2)*)
203         WRITE(7,30)SDN,SUP
204      30 FORMAT(E15.4,5X,E15.4)
205         LUP=0
206         DO 31 I=1,NUM
207         IF(IFLAG.EQ.1)PLOTY(I)=SXZ(I)
208         IF(IFLAG.EQ.2)PLOTY(I)=SYZ(I)
209         IF(KFLAG.EQ.1)PLOTX(I)=X(I)
210      31 IF(KFLAG.EQ.2)PLOTX(I)=Y(I)
211         IF(KFLAG.EQ.2)GOTO83
212         GOTO85
213      83 DO 81 I=1,NUM
214         SXZ(I)=PLOTY(I)
215     81  SYZ(I)=PLOTX(I)
216         DO 84 I=1,NUM
217         L=NUM-I+1
218         PLOTY(L)=SXZ(I)
219      84 PLOTX(L)=SYZ(I)
220         DO 82 I=1,NUM
221         IF(KFLAG.EQ.2.AND.PLOTX(I).LT.AL)LUP=LUP+1
222      82 IF(LUP.EQ.1)XCUT=PLOTX(I-1)
223         LDN=0
224         DO 86 I=1,NUM
225         L=NUM-I+1
226         IF(KFLAG.EQ.2.AND.PLOTX(L).GT.0.)LDN=LDN+1
227      86 IF(LDN.EQ.1)CUTMAX=PLOTX(L-1)
228      85 CONTINUE
229     417 LU=7 $LNLGX=1 $LNLGY=1 $NCLX=2 $NCLY=2 $WRITE(7,34)
230      34 FORMAT(*PLOT TITLE, 80 CHARACTERS MAXIMUM*)
231         READ(7,35)(LTITL(JM),JM=1,8)
232      35 FORMAT(8A10)
```

```
      IF(IFLAG.EQ.1)LTITL2(1)=10HXZ STRESS
      IF(IFLAG.EQ.2)LTITL2(1)=10HYZ STRESS
      DO 36 I=2,8
   36 LTITL2(I)=10H
      IF(KFLAG.EQ.1)KX(1)=10HX (KMS.)
      IF(KFLAG.EQ.2)KX(1)=10HY (KMS.)
      KX(2)=10H           $KY(1)=10HSTRESS      $KY(2)=10H
      XRT=PLOTX(NUM) $XLT=PLOTX(1) $MAJX=5 $MAJY=10 $YLO=SDN $YUP=SUP
      IF(KFLAG.EQ.2.AND.YMIN.LT.AL)XRT=XCUT
      IF(KFLAG.EQ.2.AND.YMAX.GT.0.)XLT=CUTMAX
      IF(KFLAG.EQ.2.AND.YMIN.GE.AL)XRT=PLOTX(NUM)
      AA=YUP$IF(YLO.EQ.AA)YUP=YUP+1.$IF(YLO.EQ.AA)YLO=YLO-1.
      WRITE(7,37)
   37 FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
      READ(7,38)CHARAC
   38 FORMAT(A1)
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO39
      WRITE(7,40)
   40 FORMAT(*MIN/MAX Y VALUES*)
      CALL GETNUM(AB) $YLO=AB(1) $YUP=AB(2)
   39 WRITE(7,41)
   41 FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
      READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO42
      WRITE(7,43)
   43 FORMAT(*MIN/MAX X VALUES*)
      CALL GETNUM(AB) $IF(KFLAG.EQ.1)XLT=AB(1)
      IF(KFLAG.EQ.1)XRT=AB(2)
      IF(KFLAG.EQ.2)XLT=-AB(2)
      IF(KFLAG.EQ.2)XRT=-AB(1)
   42 WRITE(7,44)
   44 FORMAT(*SKIP?*)
      READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN .OR. CHARAC.EQ.1H )CALL PLOTS(PLOTY,PLOTX,1,NUM)
      WRITE(7,46)
   46 FORMAT(*0=RE-START, 1=NEW PLOT*)
      READ(7,71)LCHECK
   71 FORMAT(I1)
      IF(LCHECK.EQ.0)GOTO63$IF(LCHECK.EQ.1)GOTO417
      DO 50 I=1,NUM
      IF(X(I).LT.BOUND)AMU=AMU3
      IF(X(I).GE.BOUND.AND.X(I).LE.0.)AMU=AMU2
      IF(X(I).GT.0.)AMU=AMU1
      IF(AMU.EQ.0.)AMU=1.E-10
   50 PLOTY(I)=(PLOTY(I)/(AMU))*1.E+06
      STSMID=(PLOTY(NUM)-PLOTY(1))/2.
      SUP=SDN=STSMID
      DO 51 I=1,NUM
      IF(PLOTY(I).GT.SUP)SUP=PLOTY(I)
   51 IF(PLOTY(I).LT.SDN)SDN=PLOTY(I)
      IF(IFLAG.EQ.1)WRITE(7,52) $IF(IFLAG.EQ.2)WRITE(7,53)
   52 FORMAT(*MIN/MAX VALUES OF XZ MICROSTRAIN*)
   53 FORMAT(*MIN/MAX VALUES OF YZ MICROSTRAIN*)
      WRITE(7,30)SDN,SUP
      IF(IFLAG.EQ.1)LTITL2(1)=10HXZ STRAIN
      IF(IFLAG.EQ.2)LTITL2(1)=10HYZ STRAIN
  418 KY(1)=10HMICROSTRAI          $KY(2)=10HN
      YLO=SDN $YUP=SUP
```

```
      AA=YUP $IF(YLO.EQ.AA)YUP=YLP+1. $IF(YLC.EQ.AA)YLO=YLO-1.
      WRITE(7,37)  $READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC55
      WRITE(7,40) $CALL GETNUM(AE) $YLO=AB(1) $YUP=AE(2)
   55 WRITE(7,41) $READ(7,38) CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC301
      WRITE(7,43) $CALL GETNUM(AE)
      IF(KFLAG.EQ.1)XLT=AB(1) $IF(KFLAG.EQ.1)XRT=AB(2)
      IF(KFLAG.EQ.2)XLT=-AB(2) $IF(KFLAG.EQ.2)XRT=-AB(1)
  301 WRITE(7,44) $READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )CALL PLCTS(PLCTY,PLOTX,1,NUM)
   56 WRITE(7,46) $CALL GETNUM(AE)$LCHECK=AB(1)
      IF(LCHECK.EQ.0)GOTO63$IF(LCHECK.EQ.1)GCTO418
      IF(KFLAG.NE.IFLAG)WRITE(7,57)
   57 FORMAT(*NOTE--ONLY W(X) VS. X OR W(Y) VS. Y IS COMPUTED*)
      IF(KFLAG.NE.IFLAG)GOTO63
      IF(KFLAG.EQ.1)DINC=XINC*1.E+06
      IF(KFLAG.EQ.2)DINC=YINC*1.E+06
      SUM=0.
      DO 58 I=1,NUM
      TERM=2.*PLOTY(I)*DINC*1.E-06 $SUM=SUM+TERM
   58 PLCTY(I)=SUM
      MCHECK=C
      DO 789 M=1,NUM
      IF(KFLAG.EQ.1)GRD=ABS((XMA)-XMIN)/NUM)
      IF(KFLAG.EQ.2)GRD=ABS((YMA)-YMIN)/NUM)
      IF(KFLAG.EQ.1)GOTO790
      GRD1=-ALAMED+(GRD/100.) $GRD2=-ALAMBD-(GRD/100.)
      IF(Y(M).GT.GRD2.AND.Y(M).LT.GRD1)KMID=M
      IF(Y(M).GT.GRD2.AND.Y(M).LT.GRD1)MCHECK=1
      GOTO789
  790 GRD1=-A+(GRD/100.)  $GRD2=-A-(GRD/100.)
      IF(X(M).GT.GRD2.AND.X(M).LT.GRD1)KMID=M
      IF(X(M).GT.GRD2.AND.X(M).LT.GRD1)MCHECK=1
  789 CONTINUE
      IF(MCHECK.EQ.0)STSMID=((PLCTY(NUM)-PLCTY(1))/2.)+PLOTY(1)
      IF(MCHECK.EQ.1)STSMID=PLOTY(KMID)
      DO 73 I=1,NUM
   73 PLCTY(I)=PLOTY(I)-STSMID
      STSMID=(PLCTY(NUM)-PLCTY(1))/2.
      SUP=SDN=STSMID
      DO 59 I=1,NUM
      IF(PLOTY(I).GT.SUP)SUP=PLOTY(I)
   59 IF(PLOTY(I).LT.SDN)SDN=PLCTY(I)
      IF(IFLAG.EQ.1) WRITE(7,60) $IF(IFLAG.EQ.2)WRITE(7,61)
      IF(KFLAG.EQ.1.AND.SET.NE.C.)WRITE(7,87)
   87 FORMAT(*CAUTION---THIS IS NOT THE TOTAL DISPLACEMENT*)
   60 FORMAT(*MIN/MAX VALUES OF W(X) VS. X*)
   61 FORMAT(*MIN/MAX VALUES OF W(Y) VS. Y*)
      WRITE(7,30) SDN,SUP
  419 IF(IFLAG.EQ.1)LTITL2(1)=10HW(X) VS. X
      IF(IFLAG.EQ.2)LTITL2(1)=10HW(Y) VS. Y
      KY(1)=10HDISPLACEME      $KY(2)=10HNT-MMS.
      YLO=SDN  $YUP=SUP$AA=YUP$IF(YLO.EQ.AA)YUP=YUP+1.
      IF(YLO.EQ.AA)YLO=YLO-1.
      WRITE(7,37) $READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO62
      WRITE(7,40) $CALL GETNUM(AE) $YLO=AB(1) $YUP=AE(2)
```

```
    62  WRITE(7,41) $READ(7,38)CHARAC
        IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO303
        WRITE(7,43) $CALL GETNUM(AB)
        IF(KFLAG.EQ.1)XLT=AB(1)  $IF(KFLAG.EQ.1)XRT=AB(2)
        IF(KFLAG.EQ.2)XLT=-AB(2) $IF(KFLAG.EQ.2)XRT=-AB(1)
   303  WRITE(7,44) $READ(7,38)CHARAC
        IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H ) CALL PLOTS(PLOTY,PLOTX,1,NUM)
        WRITE(7,46)$CALL GETNUM(AB)$LCHECK=AB(1)$IF(LCHECK.EQ.1)GO TO419
    63  ALAMBD=ALAMBD-WIDTH $B=-B
        WRITE(7,45)AMU1,AMU2,AMU3,AM,AK
        BB=B*1.E+06
        WRITE(7,47)A,ALAMBD,WIDTH,D,BB
        IF(IFLAG.EQ.1)CD1=2HXZ$IF(KFLAG.EQ.1)CD2=1HX
        IF(IFLAG.EQ.2)CD1=2HYZ$IF(KFLAG.EQ.2)CD2=1HY
        WRITE(7,49)CD1,CD2
    45  FORMAT(*MU1 = *,E20.4,5X,*MU2 = *,E20.4,/,
       1       *MU3 = *,E20.4,5X,*  M = *,E20.4,/,
       1       *  K = *,E20.4)
    47  FORMAT(*DISLOCATION POSITION = -*,F10.3,*, -*,F10.3,* KMS.*,/,
       1       *WIDTH OF SLIP ZONE   =  *,F10.3,/,
       1       *SLAB THICKNESS       =  *,F10.3,* KMS.*,/,
       1       *BURGERS VECTOR       =  *,F10.3,* MMS.*)
    49  FORMAT(A2,* STRESS/STRAIN IS. *,A2)
        WRITE(7,64)
    64  FORMAT(*0=RE-START*,/,*1=STOP*,/,
       1       *2=RE-START WITH SAME VALUES, DIFFERENT PROFILE*)
        CALL GETNUM(AB) $LJK=AB(1)
        IF(LJK.EQ.0)GOTO48 $IF(LJK.EQ.2)GO TO 70
        STOP $END
        SUBROUTINE FUNC1(P,A,NU,C,G,ALAMBD,F)
        R=(((P+A+(2.*NU*D))**2)+((G+ALAMBD)**2))
        IF(R.EQ.0.)R=1.E-10$F=(P+A+(2.*NU*D))/R
        RETURN $END
        SUBROUTINE FUNC2(P,Q,R,S,A,NU,D,ALAMBD,F)
        T=(((R+A+(2.*NU*D))**2)+((S+ALAMBD)**2))
        IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
        RETURN $END
        SUBROUTINE FUNC3(P,Q,R,S,F)
        T=(((R+S)**2)+((P+Q)**2))
        IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
        RETURN $END
        SUBROUTINE GETNUM(R)
        DIMENSION F(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  $GOTO5
      2 M=-1 $S=4 $GOTO5
      3 P=I $S=4 $GOTO5
      4 IF(P.NE.0)F=10.**(I-P-1) $R(J)=N/F*M $GOTO6
      9 FORMAT(80R1)
        END
```

LISTING OF STNRML

```
04 MAY 76    09.12.28         MCHUGH .STNMRL

  1  DELETE(LGO,OUTPUT,STNRML)
  2  STNRML.
  3  CXIT.
  4  LIBCOPY(GRAPHIC,TXLGO/RR,TXLGO)
  5  LIBCOPY(JDRAT,NPLGO/RR,NPLGO)
  6  DELETE(LGO,OUTPUT,STNRML)
  7  RUN76(S)
  8  LINK(F=LGO,F=TXLGO,F=NPLGO,B=STNRML)
  9  STNRML.
 10  FIN.
 11  EOR
 12        PROGRAM STNRML(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
 13        COMMON/TVPCOL/TVPUL(8)
 14        COMMON/TVTUNE/ITUNE(30)
 15        COMMON/JPLOT/XLT,XRT,YLC,YLP,MAJX,MAJY,KX(2),KY(2),
 16       1LTITL(8),LU,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
 17        DIMENSION IFET(8)
 18        DIMENSION X(200),Y(200),AMATRX(200,3,2),AB(30)
 19        DIMENSION PLOTY(200),PLOTX(200)
 20        DATA YMIN,YMAX,XCUT,CUTMAX/-1.E+10,-1.E+10,-1.E+10,-1.E+10/
 21        CALL FET(5LTAPE7,IFET,8)
 22        IFET(2)=IFET(2).OR.0000 0010 0000 0000 0000B
 23        IFET(8)=IFET(8).OR.4000 0000 0000 0000 0000B
 24        CALL FET(5LTAPE7,IFET,-8)
 25    48  WRITE(7,15)
 26    15  FORMAT(*SHEAR MODULI--IN CGS UNITS--MU1,MU2,MU3*)
 27        CALL GETNUM(AB) $IAB2=AB(2) $IAB4=AB(4) $IAB6=AB(6)
 28        SETMU1=AB(1)*(10.**IAB2) $SETMU2=AB(3)*(10.**IAB4)
 29        SETMU3=AB(5)*(10.**IAB6)
 30        WRITE(7,95)
 31     95 FORMAT(*SHEAR MODULUS OF HOMOGENEOUS HALF-SPACE*)
 32        CALL GETNUM(AB) $AMUHOM=AB(1)*(10.**AB(2))
 33        WRITE(7,16)
 34    16  FORMAT(*POSITION OF DISLOCATION--IN KMS., (A,LAMBDA)*,/,
 35       1         *NOTE---A>0., LAMBDA>0.*)
 36        CALL GETNUM(AB) $A=AB(1) $ALAMBD=AB(2)
 37        WRITE(7,17)
 38    17  FORMAT(*THICKNESS OF SLAB, IN KMS.--D*)
 39        CALL GETNUM(AB) $D=AB(1)  $WRITE(7,18)
 40    18  FORMAT(*BURGERS VECTOR, IN MMS.--B*,/,
 41       1       *B<0 FOR RIGHT LATERAL SLIP*)
 42        CALL GETNUM(AB) $B=-AB(1)*1.E-06
 43    70  WRITE(7,19)
 44    19  FORMAT(*1=AMPLITUDE VS. X, 2=AMPLITUDE VS. Y*)
 45        CALL GETNUM(AB) $KFLAG=AB(1)$WRITE(7,20)
 46    20  FORMAT(*1=XZ STRESS, 2=YZ STRESS*)
 47        CALL GETNUM(AB)  $IFLAG=AB(1)
 48        IF(KFLAG.EQ.2)GOTO201
 49        WRITE(7,21) $CALL GETNUM(AB) $XMIN=AB(1)
 50        XMAX=AB(2) $ WRITE(7,22)  $CALL GETNUM(AB) $SET=-AB(1)
 51        GO TO 202
 52    201 WRITE(7,23) $CALL GETNUM(AB)
 53        YMAX=-AB(1) $YMIN=-AB(2) $WRITE(7,24)
 54        CALL GETNUM(AB) $SET=AB(1)
 55    21  FORMAT(*ENTER MINIMUM AND MAXIMUM X VALUES, IN KMS.*)
 56    22  FORMAT(*DEPTH OF PROFILE, IN KMS.*)
 57    23  FORMAT(*ENTER MINIMUM AND MAXIMUM Y VALUES, IN KMS.*)
 58    24  FORMAT(*X COORDINATE OF PROFILE, IN KMS.*)
```

```
    202 NUM=200 $IUPPER=10  $EPS=1.E-10 $PI=3.14159
        IF(KFLAG.EQ.1)XINC=(XMAX-XMIN)/NUM
        IF(KFLAG.EQ.2)YINC=(YMAX-YMIN)/NUM
        WRITE(7,25) NUM,IUPPER
     25 FORMAT(*NOTE--NUMBER OF DATA POINTS = *,I5,/,
       1       *      UPPER LIMIT ON SERIES = *,I5)
        AMU1=SETMU1$AMU2=SETMU2$AMU3=SETMU3
        AM=(AMU3-AMU2)/(AMU3+AMU2)$IF(AM.EQ.0.)AM=1.E-20
        AK=(AMU1-AMU2)/(AMU1+AMU2) $IF(AK.EQ.0.)AK=1.E-20
        BOUND=-D $AL=-ALAMBD $AR=-A
        IF(KFLAG.EQ.1 .AND.AL.GT.SET)GOTO74
        GOTO75
     74 WRITE(7,76)
     76 FORMAT(*WARNING---PROFILE BELOW DISLOCATION LINE*,/,
       1       *SOLUTION INVALID, TRY AGAIN*)
        GOTO63
     75 IF(KFLAG.EQ.1)GOTO90
        RDIST=ABS(SET-(-A))
        IF(RDIST.LT.YINC)WRITE(7,91)
        IF(RDIST.LT.YINC)GOTO63
        GOTO92
     90 RDIST=ABS(SET-(-ALAMBD))
        IF(RDIST.LT.XINC)WRITE(7,91)
        IF(RDIST.LT.XINC)GOTO63
     91 FORMAT(*WARNING---PROFILE TOO CLOSE TO DISLOCATION CORE,*,/,
       1       *CHOOSE NEW PROFILE OR INCREASE GRID POINT DENSITY*)
     92 CONTINUE
        IF(KFLAG.EQ.1.AND.SET.GT.0.)WRITE(7,93)
        IF(KFLAG.EQ.1.AND.SET.GT.0.)GOTO63
     93 FORMAT(*WARNING---PROFILE ABOVE SURFACE OF HALF-SPACE,*,/,
       1       *SOLUTION INVALID, TRY AGAIN*)
        DO 230 K=1,2
        IF(K.EQ.2)GOTO231
        AMU1=AMU2=AMU3=AMUHOM
        GOTO221
    231 AMU1=SETMU1 $AMU2=SETMU2 $AMU3=SETMU3
    221 AM=(AMU3-AMU2)/(AMU3+AMU2)$IF(AM.EQ.0.)AM=1.E-20
        AK=(AMU1-AMU2)/(AMU1+AMU2)$IF(AK.EQ.0.)AK=1.E-20
        DO 200 J=1,NUM
        IF(KFLAG.EQ.1)X(J)=XMIN+(XINC*(J-1))
        IF(KFLAG.EQ.1) Y(J)=SET
        IF(KFLAG.EQ.2)X(J)=SET
        IF(KFLAG.EQ.2)Y(J)=YMIN+(YINC*(J-1))
        IF(X(J).EQ.AR.AND.Y(J).EQ.AL)GOTO32
        GO TO 33
     32 X(J)=X(J)+1.E-10 $Y(J)=Y(J)+1.E-10
     33 IF(X(J) .LT.BOUND)GOTO103
        IF(X(J).GE.BOUND.AND.X(J).LE.0.)GOTO102
    101 IF(IFLAG .EQ. 2)GOTO12
        SUM1=SUM2=0.
        DO 1 I=1,IUPPER
        NU=I-1
        CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
        TERM1=((AM*AK)**NU)*F1
        CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,I,D,ALAMBD,F2)
        TERM2=  (AM**I)*(AK**NU)  *F2
        SUM1=SUM1+TERM1     $SUM2=SUM2+TERM2
    1   CONTINUE
```

```
      AMATRX(J,1,K)=(((-AMU1*B)/(2.*PI))*(1.-AK))*(SUM1+SUM2) $GOTO200
   12 SUM1=SUM2=SUM3=SUM4=0.
      DO 5 I=1,IUPPER
      NU=I-1
      CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
      CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMBD,F2)
      CALL FUNC1(X(J),-A,I,D,Y(J),ALAMBD,F3)
      CALL FUNC1(X(J),-A,I,D,Y(J),-ALAMBD,F4)
      TERM1=((AM*AK)**NU)*F1 $TERM2=((AM*AK)**NU)*F2
      TERM3=(AM**I)*(AK**NU)*F3 $TERM4=(AM**I)*(AK**NU)*F4
      SUM1=SUM1+TERM1 $SUM2=SUM2+TERM2 $SUM3=SUM3+TERM3
      SUM4=SUM4+TERM4
    5 CONTINUE
      AMATRX(J,1,K)=((AMU1*B)*(1.-AK)/(2.*PI))*(SUM1+SUM3-SUM2-SUM4)
      GOTO200
  102 IF(IFLAG.EQ.2)GOTO13
      SUM1=0.
      DO 7 NU=1,IUPPER
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMBD,F2)
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,NU,D,ALAMBD,F3)
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMBD,F4)
      TERM=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4) $SUM1=SUM1+TERM
    7 CONTINUE
      CALL FUNC3(Y(J),ALAMBD,X(J),A,F1)
      CALL FUNC3(Y(J),ALAMBD,X(J),-A,F2)
      TERM2=F1+(AK*F2)
      AMATRX(J,1,K)=((-AMU2*B)/(2.*PI))*(TERM2+SUM1) $GOTO200
   13 SUM1=SUM2=0.
      DO 3 NU=1,IUPPER
      CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
      CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMBD,F2)
      CALL FUNC1(X(J),-A,NU,D,Y(J),ALAMBD,F3)
      CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMBD,F4)
      TERM1=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4)
      SUM1=SUM1+TERM1
      CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMBD,F5)
      CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F6)
      CALL FUNC1(X(J),-A,NU,D,Y(J),-ALAMBD,F7)
      CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMBD,F8)
      TERM2=((AM*AK)**NU)*(F5+(AK*F6)+((1./AK)*F7)+F8)
      SUM2=SUM2+TERM2
    3 CONTINUE
      CALL FUNC3(X(J),A,Y(J),ALAMBD,F1)
      CALL FUNC3(X(J),-A,Y(J),ALAMBD,F2)
      CALL FUNC3(X(J),A,Y(J),-ALAMBD,F3)
      CALL FUNC3(X(J),-A,Y(J),-ALAMBD,F4)
      AMATRX(J,1,K)=((AMU2*B)/(2.*PI))*((F1+(AK*F2)+SUM1)-
     1 (F3+(AK*F4)+SUM2))
      GO TO 200
  103 IF(IFLAG.EQ.2)GOTO14
      SUM1=0.
      DO 9 I=1,IUPPER
      NU=I-1
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMBD,F1)
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMBD,F2)
      TERM=(F1+(AK*F2))*((AK*AM)**NU) $SUM1=SUM1+TERM
    9 CONTINUE
```

```
      AMATRX(J,1,K)=((-AMU3*B)/(2.*PI))*(1.-AM)*SUM1 $GOTO200
   14 SUM1=SUM2=0.
      DO 11 I=1,IUPPER
      NU=I-1
      CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMED,F1)
      CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMED,F2)
      TERM1=((AM*AK)**NU)*(F1+(AK*F2))
      SUM1=SUM1+TERM1
      CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMED,F1)
      CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F2)
      TERM2=((AM*AK)**NU)*(F1+(AK*F2)) $SUM2=SUM2+TERM2
   11 CONTINUE
      AMATRX(J,1,K)=((AMU3*B)/(2.*PI))*(1.-AM)*(SUM1-SUM2)
  200 CONTINUE
  230 CONTINUE
      DO 204 I=1,NUM
      IF(AMATRX(I,1,1).EQ.0.)AMATRX(I,1,1`=1.E-10
  204 PLOTY(I)=AMATRX(I,1,2)/AMATRX(I,1,1)
      L12=NUM/2 $STSMID=PLOTY(L12)
      SUP=SDN=STSMID
      DO 26 I=1,NUM
      IF(PLOTY(I).GT.SUP)SUP=PLOTY(I)
   26 IF(PLOTY(I).LT.SDN)SDN=PLOTY(I)
      IF(IFLAG.EQ.1)WRITE(7,28) $IF(IFLAG.EQ.2)WRITE(7,29)
   28 FORMAT(*MIN/MAX VALUES OF XZ NORMALIZED STRESS*)
   29 FORMAT(*MIN/MAX VALUES OF YZ NORMALIZED STRESS*)
      WRITE(7,30)SDN,SUP
   30 FORMAT(E15.4,5X,E15.4)
      LUP=0
      DO 31 I=1,NUM
      IF(KFLAG.EQ.1)PLOTX(I)=X(I)
   31 IF(KFLAG.EQ.2)PLOTX(I)=Y(I)
      IF(KFLAG.EQ.2)GOTO83
      GOTO85
   83 DO 81 I=1,NUM
      AMATRX(I,2,1)=PLOTY(I)
   81 AMATRX(I,2,2)=PLOTX(I)
      DO 84 I=1,NUM
      L=NUM-I+1
      PLOTY(L)=AMATRX(I,2,1)
   84 PLOTX(L)=AMATRX(I,2,2)
      DO 82 I=1,NUM
      IF(KFLAG.EQ.2.AND.PLOTX(I).LT.AL)LUP=LUP+1
   82 IF(LUP.EQ.1)XCUT=PLOTX(I-1)
      LDN=0
      DO 86 I=1,NUM
      L=NUM-I+1
      IF(KFLAG.EQ.2.AND.PLOTX(L).GT.0.)LDN=LDN+1
   86 IF(LDN.EQ.1)CUTMAX=PLOTX(L-1)
   85 CONTINUE
  417 LU=7 $LNLGX=1 $LNLGY=1 $NCLX=2 $NCLY=2 $WRITE(7,34)
   34 FORMAT(*PLOT TITLE, 80 CHARACTERS MAXIMUM*)
      READ(7,35)(LTITL(JM),JM=1,8)
   35 FORMAT(8A10)
      IF(IFLAG.EQ.1)LTITL2(1)=10HXZ STRESS-
      IF(IFLAG.EQ.1)LTITL2(2)=10HNORMALIZED
      IF(IFLAG.EQ.2)LTITL2(1)=10HYZ STRESS-
      IF(IFLAG.EQ.2)LTITL2(2)=10HNORMALIZED
```

```
233         DO 36 I=3,8
234      36 LTITL2(I)=10H
235         IF(KFLAG.EQ.1)KX(1)=10HX (KMS.)
236         IF(KFLAG.EQ.2)KX(1)=10HY (KMS.)
237         KX(2)=10H              $KY(1)=10HSTRESS       $KY(2)=10H
238         XRT=PLOTX(NUM) $XLT=PLOTX(1) $MAJX=5 $MAJY=10 $YLO=SDN $YUP=SUP
239         IF(KFLAG.EQ.2.AND.YMIN.LT.AL)XRT=XCUT
240         IF(KFLAG.EQ.2.AND.YMAX.GT.0.)XLT=CUTMAX
241         IF(KFLAG.EQ.2.AND.YMIN.GE.AL)XRT=PLOTX(NUM)
242         AA=YUP$IF(YLO.EQ.AA)YUP=YLP+1.$IF(YLO.EQ.AA)YLO=YLO-1.
243         WRITE(7,37)
244      37 FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
245         READ(7,38)CHARAC
246      38 FORMAT(A1)
247         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO39
248         WRITE(7,40)
249       40 FORMAT(*MIN/MAX Y VALUES*)
250         CALL GETNUM(AB) $YLO=AB(1) $YUP=AB(2)
251      39 WRITE(7,41)
252       41 FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
253         READ(7,38)CHARAC
254         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO42
255         WRITE(7,43)
256       43 FORMAT(*MIN/MAX X VALUES*)
257         CALL GETNUM(AB) $IF(KFLAG.EQ.1)XLT=AB(1)
258         IF(KFLAG.EQ.1)XRT=AB(2)
259         IF(KFLAG.EQ.2)XLT=-AB(2)
260         IF(KFLAG.EQ.2)XRT=-AB(1)
261       42 WRITE(7,44)
262       44 FORMAT(*SKIP?*)
263         READ(7,38)CHARAC
264         IF(CHARAC.EQ.1HN .OR. CHARAC.EQ.1H )CALL PLOTS(PLOTY,PLOTX,1,NUM)
265         WRITE(7,46)
266      46 FORMAT(*0=RE-START, 1=NEW PLOT*)
267         READ(7,71)LCHECK
268      71 FORMAT(I1)
269         IF(LCHECK.EQ.0)GOTO63 $IF(LCHECK.EQ.1)GOTO417
270         DO 210 K=1,2
271         IF(K.EQ.2)GOTO211
272         AMU1=AMU2=AMU3=AMUHOM
273         GOTO212
274      211 AMU1=SETMU1 $AMU2=SETMU2 $AMU3=SETMU3
275      212 DO 50 I=1,NUM
276         IF(X(I).LT.BOUND)AMU=AMU3
277         IF(X(I).GE.BOUND.AND.X(I).LE.0.)AMU=AMU2
278         IF(X(I).GT.0.)AMU=AMU1
279         IF(AMU.EQ.0.)AMU=1.E-10
280      50 AMATRX(I,2,K)=(AMATRX(I,1,K)/AMU)*1.E+06
281      210 CONTINUE
282         DO 213 I=1,NUM
283         IF(AMATRX(I,2,1).EQ.0.)AMATRX(I,2,1)=1.E-10
284      213 PLOTY(I)=AMATRX(I,2,2)/AMATRX(I,2,1)
285         L12=NUM/2 $STSMID=PLOTY(L12)
286         SUP=SDN=STSMID
287         DO 51 I=1,NUM
288         IF(PLOTY(I).GT.SUP)SUP=PLOTY(I)
289      51 IF(PLOTY(I).LT.SDN)SDN=PLOTY(I)
290         IF(IFLAG.EQ.1)WRITE(7,52) $IF(IFLAG.EQ.2)WRITE(7,53)
```

```
291     52  FORMAT(*MIN/MAX VALUES OF )Z NORMALIZED STRAIN*)
292     53  FORMAT(*MIN/MAX VALUES OF YZ NORMALIZED STRAIN*)
293         WRITE(7,30)SDN,SUP
294         IF(IFLAG.EQ.1)LTITL2(1)=10HXZ STRAIN
295         IF(IFLAG.EQ.2)LTITL2(1)=10HYZ STRAIN
296     418 KY(1)=10HSTRAIN           $KY(2)=10H
297         YLO=SDN $YUP=SUP
298         AA=YUP $IF(YLO.EQ.AA)YUP=YLP+1. $IF(YLO.EQ.AA)YLO=YLO-1.
299         WRITE(7,37)  $READ(7,38)CHARAC
300         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO55
301         WRITE(7,40) $CALL GETNUM(AE) $YLC=AB(1) $YLP=AE(2)
302      55 WRITE(7,41) $READ(7,38)CHARAC
303         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO301
304         WRITE(7,43) $CALL GETNUM(AE)
305         IF(KFLAG.EQ.1)XLT=AB(1)  $IF(KFLAG.EQ.1)XRT=AB(2)
306         IF(KFLAG.EQ.2)XLT=-AB(2)  $IF(KFLAG.EQ.2)XRT=-AB(1)
307     301 WRITE(7,44) $READ(7,38)CHARAC
308         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )CALL PLOTS(PLOTY,PLOTX,1,NUM)
309      56 WRITE(7,46) $CALL GETNUM(AE)$LCHECK=AE(1)
310         IF(LCHECK.EQ.0)GOTO63 $IF(LCHECK.EQ.1)GOTO418
311         IF(KFLAG.NE.IFLAG)WRITE(7,57)
312     57  FORMAT(*NOTE--ONLY W(X) VS. X OR W(Y) VS. Y IS COMPUTED*)
313         IF(KFLAG.NE.IFLAG)GOTO63
314         IF(KFLAG.EQ.1)DINC=XINC*1.E+06
315         IF(KFLAG.EQ.2)DINC=YINC*1.E+06
316         DO 58 K=1,2
317         SUM=0.
318         DO 214 I=1,NUM
319         TERM=2.*AMATRX(I,2,K)*DINC*1.E-06
320         SUM=SUM+TERM
321     214 AMATRX(I,3,K)=SUM
322         MCHECK=0
323         DO 789 M=1,NUM
324         IF(KFLAG.EQ.1)GRD=ABS((XMAX-XMIN)/NUM)
325         IF(KFLAG.EQ.2)GRD=ABS((YMAX-YMIN)/NUM)
326         IF(KFLAG.EQ.1)GOTO790
327         GRD1=-ALAMED+(GRD/100.)  $GRD2=-ALAMBC-(GRD/100.)
328         IF(Y(M).GT.GRD2.AND.Y(M).LT.GRD1)KMID=M
329         IF(Y(M).GT.GRD2.AND.Y(M).LT.GRD1)MCHECK=1
330         GOTO789
331     790 GRD1=-A+(GRD/100.) $GRD2=-A-(GRD/100.)
332         IF(X(M).GT.GRD2.AND.X(M).LT.GRD1)KMID=M
333         IF(X(M).GT.GRD2.AND.X(M).LT.GRD1)MCHECK=1
334     789 CONTINUE
335         IF(MCHECK.EQ.0)STSMID=((AMATRX(NUM,3,K)-AMATRX(1,3,K))/2.)
336        1  +AMATRX(1,3,K)
337         IF(MCHECK.EQ.1)STSMID=AMATRX(KMID,3,K)
338         DO 73 I=1,NUM
339      73 AMATRX(I,3,K)=AMATRX(I,3,K)-STSMID
340      58 CONTINUE
341         DO 215 I=1,NUM
342         IF(AMATRX(I,3,1).EQ.0.)AMATRX(I,3,1)=1.E-10
343     215 PLOTY(I)=AMATRX(I,3,2)/AMATRX(I,3,1)
344         L12=NUM/2 $STSMID=PLOTY(L12)
345         SUP=SDN=STSMID
346         DO 59 I=1,NUM
347         IF(PLOTY(I).GT.SUP)SUP=PLOTY(I)
348     59  IF(PLOTY(I).LT.SDN)SDN=PLOTY(I)
```

```
      IF(IFLAG.EQ.1) WRITE(7,60) $IF(IFLAG.EQ.2)WRITE(7,61)
      IF(KFLAG.EQ.1.AND.SET.NE.0.)WRITE(7,87)
   87 FORMAT(*CAUTION---THIS IS NOT THE TOTAL DISPLACEMENT*)
   60 FORMAT(*MIN/MAX VALUES OF W(X) VS. X*)
   61 FORMAT(*MIN/MAX VALUES OF W(Y) VS. Y*)
      WRITE(7,30) SDN,SUP
  419 IF(IFLAG.EQ.1)LTITL2(1)=10HW(X) VS X
      IF(IFLAG.EQ.2)LTITL2(1)=10HW(Y) VS Y
      KY(1)=10HDISPLACEME      $KY(2)=10HNT
      YLO=SDN  $YUP=SUP$AA=YUP$IF(YLO.EQ.AA)YUP=YUP+1.
      IF(YLO.EQ.AA)YLO=YLO-1.
      WRITE(7,37) $READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO62
      WRITE(7,40) $CALL GETNUM(AB) $YLO=AB(1) $YUP=AB(2)
   62 WRITE(7,41) $READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H ) GOTO303
      WRITE(7,43)  $CALL GETNUM(AB)
      IF(KFLAG.EQ.1)XLT=AB(1)  $IF(KFLAG.EQ.1)XRT=AB(2)
      IF(KFLAG.EQ.2)XLT=-AB(2)  $IF(KFLAG.EQ.2)XRT=-AB(1)
  303 WRITE(7,44) $READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H ) CALL PLOTS(PLOTY,PLOTX,1,NUM)
      WRITE(7,46)$CALL GETNUM(AB)$LCHECK=AB(1)$IF(LCHECK.EQ.1)GOTO419
   63 WRITE(7,45)SETMU1,SETMU2,SETMU3,AMUHOM,AM,AK
      BB=B*1.E+06
      WRITE(7,47)A,ALAMBD,D,BB
      IF(IFLAG.EQ.1)CD1=2HXZ$IF(KFLAG.EQ.1)CD2=1HX
      IF(IFLAG.EQ.2)CD1=2HYZ$IF(KFLAG.EQ.2)CD2=1HY
      WRITE(7,49)CD1,CD2
   45 FORMAT(*MU1 = *,E20.4,5X,*MU2 = *,E20.4,/,
     1       *MU3 = *,E20.4,5X,*MUH = *,E20.4,/,
     1       *  M = *,E20.4,5X,*  K = *,E20.4)
   47 FORMAT(*DISLOCATION POSITION = -*,F10.3,*, -*,F10.3,* KMS.*,/,
     1       *SLAB THICKNESS       =  *,F10.3,* KMS.*,/,
     1       *BURGERS VECTOR       =  *,F10.3,* MMS.*)
   49 FORMAT(A2,* STRESS/STRAIN \S. *,A2)
      WRITE(7,64)
   64 FORMAT(*0=RE-START*,/,*1=STOP*,/,
     1       *2=RE-START WITH SAME VALUES, DIFFERENT PROFILE*)
      CALL GETNUM(AB) $LJK=AB(1)
      IF(LJK.EQ.0)GOTO48 $IF(LJK.EQ.2)GO TO 70
      STOP $END
      SUBROUTINE FUNC1(P,A,NU,D,Q,ALAMBD,F)
      R=(((P+A+(2.*NU*D))**2)+((Q+ALAMBD)**2))
      IF(R.EQ.0.)P=1.E-10$F=(P+A+(2.*NU*D))/R
      RETURN $END
      SUBROUTINE FUNC2(P,Q,R,S,A,NU,D,ALAMBD,F)
      T=(((R+A+(2.*NU*D))**2)+((S+ALAMBD)**2))
      IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
      RETURN $END
      SUBROUTINE FUNC3(P,Q,R,S,F)
      T=(((R+S)**2)+((P+Q)**2))
      IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
      RETURN $END
      SUBROUTINE GETNUM(K)
      DIMENSION K(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  $GOTO5
    2 M=-1 $S=4 $GOTO5
    3 P=I $S=4 $GOTO5
    4 IF(P.NE.0)F=10.**(I-P-1) $R(J)=N/F*M $GOTO6
    9 FORMAT(80R1)
      END
```

LISTING OF SHRSTN

```
04 MAY 76    09.10.43          MCHUGH .SHRSTN

  1  DELETE(LGO,OUTPUT,SHRSTN)
  2  SHRSTN.
  3  EXIT.
  4  LIBCOPY(GRAPHIC,TXLGO/RR,TXLGO)
  5  LIBCOPY(JDRAT,NPLGO/FR,NPLGO)
  6  DELETE(LGO,OUTPUT,SHRSTN)
  7  RUN76(S)
  8  LINK(F=LGO,F=TXLGO,F=NPLGO,B=SHRSTN)
  9  SHRSTN.
 10  FIN.
 11  FOR
 12        PROGRAM SHRSTN(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
 13        COMMON/TVPCOL/TVPUL(8)
 14        COMMON/TVTUNE/ITUNE(30)
 15        COMMON/JPLOT/XLT,XRT,YLC,YUP,MAJX,MAJY,KX(2),KY(2),
 16       1LTITL(8),LU,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
 17        DIMENSION IFET(8)
 18        DIMENSION X(200),Y(200),AMATRX(200,3,2),AB(30)
 19        DIMENSION PLOTY(200),PLOTX(200)
 20        DIMENSION STRSMX(200),STRNMX(200),DISPMX(200)
 21        DATA YMIN,YMAX,XCUT,CUTMAX/-1.E+10,-1.E+10,-1.E+10,-1.E+10/
 22        DATA YINC/1./
 23        CALL FET(5LTAPE7,IFET,8)
 24        IFET(2)=IFET(2).OR.0000 0010 0000 0000 0000B
 25        IFET(8)=IFET(8).OR.4000 0000 0000 0000 0000B
 26        CALL FET(5LTAPE7,IFET,-8)
 27    48  WRITE(7,15)
 28    15  FORMAT(*SHEAR MODULI--IN CGS UNITS--MU1,MU2LOW,MU2HIGH,MU3=MU1*)
 29        CALL GETNUM(AB) $IAB2=AB(2) $IAB4=AB(4) $IAB6=AB(6)
 30        SETMU1=AB(1)*(10.**IAB2) $BGNMU2=AB(3)*(10.**IAB4)
 31        ENDMU2=AB(5)*(10.**IAB6)
 32        WRITE(7,95)
 33     95 FORMAT(*SHEAR MODULUS OF HOMOGENEOUS HALF-SPACE*)
 34        CALL GETNUM(AB) $AMUHOM=AB(1)*(10.**AB(2))
 35        WRITE(7,16)
 36    16  FORMAT(*POSITION OF DISLOCATION--IN KMS., (A,LAMBDA)*,/,
 37       1          *NOTE---A>0., LAMBDA>0.*)
 38        CALL GETNUM(AB) $A=AB(1) $ALAMBD=AB(2)
 39        WRITE(7,17)
 40    17  FORMAT(*THICKNESS OF SLAB, IN KMS.--D*)
 41        CALL GETNUM(AB) $D=AB(1)  $WRITE(7,18)
 42    18  FORMAT(*BURGERS VECTOR, IN MMS.--B*,/,
 43       1       *B<0 FOR RIGHT LATERAL SLIP*)
 44        CALL GETNUM(AB) $B=-AB(1)*1.E-06
 45    70  WRITE(7,19)
 46    19  FORMAT(*1=AMPLITUDE VS. X, 2=AMPLITUDE VS. Y*)
 47        CALL GETNUM(AB) $KFLAG=AB(1)$WRITE(7,20)
 48    20  FORMAT(*1=XZ STRESS, 2=YZ STRESS*)
 49        CALL GETNUM(AB)  $IFLAG=AB(1) $WRITE(7,250)
 50        CALL GETNUM(AB) $LNUM=AB(1)
 51        IF(LNUM.EQ.3.AND.KFLAG.EQ.1)WRITE(7,251)
 52        IF(LNUM.EQ.4.AND.KFLAG.EQ.1)WRITE(7,251)
 53    251 FORMAT(*ENTER X COORDINATE OF POINT*)
 54        IF(LNUM.EQ.3 .AND.KFLAG.EQ.2)WRITE(7,252)
 55        IF(LNUM.EQ.4.AND.KFLAG.EQ.2)WRITE(7,252)
 56    252 FORMAT(*ENTER Y COORDINATE OF POINT*)
 57        IF(LNUM.EQ.3.OR.LNUM.EQ.4)CALL GETNUM(AB)
 58        IF(LNUM.EQ.3.OR.LNUM.EQ.4)COORD=AB(1)
```

```
      IF(KFLAG.EQ.1)WRITE(7,22)
      IF(KFLAG.EQ.1)CALL GETNUM(AB)
      IF(KFLAG.EQ.1)SET=-AB(1)
      IF(LNUM.EQ.3)GOTO253 $IF(KFLAG.EQ.2)GOTO201
      WRITE(7,21) $CALL GETNUM(AB) $XMIN=AB(1)
      XMAX=AB(2) $GOTO202
  201 WRITE(7,23) $CALL GETNUM(AB)
      YMAX=-AB(1) $YMIN=-AB(2) $WRITE(7,24)
      CALL GETNUM(AB) $SET=AB(1)
   21 FORMAT(*ENTER MINIMUM AND MAXIMUM X VALUES, IN KMS.*)
   22 FORMAT(*DEPTH OF PROFILE, IN KMS.*)
   23 FORMAT(*ENTER MINIMUM AND MAXIMUM Y VALUES, IN KMS.*)
   24 FORMAT(*X COORDINATE OF PROFILE, IN KMS.*)
  250 FORMAT(*1=COMPUTE VARIATION OF GLOBAL MINIMUM OF PROFILE*,/,
     1       *2=COMPUTE VARIATION OF GLOBAL MAXIMUM OF PROFILE*,/,
     1       *3=COMPUTE VARIATION OF STRESS AND STRAIN ONLY AT A *,/,
     1       *  POINT ON THE PROFILE*,/,
     1       *4=COMPUTE VARIATION OF STRESS, STRAIN, AND DISPLACEMENT*
     1    ,/,*  AT A POINT ON THE PROFILE*)
  202 CONTINUE
  253 NUM=100 $IUPPER=20  $PI=3.14159
      IF(KFLAG.EQ.1.AND.LNUM.NE.3)XINC=(XMAX-XMIN)/NUM
      IF(KFLAG.EQ.2.AND.LNUM.NE.3)YINC=(YMAX-YMIN)/NUM
      WRITE(7,25)NUM,IUPPER
   25 FORMAT(*NOTE--NUMBER OF DATA POINTS = *,I5,/,
     1       *      UPPER LIMIT ON SERIES = *,I5)
      AMU1=SETMU1$AMU2=ENDMU2$AMU3=SETMU3=SETMU1
      AM=(AMU3-AMU2)/(AMU3+AMU2)$IF(AM.EQ.0.)AM=1.E-20
      AK=(AMU1-AMU2)/(AMU1+AMU2) $IF(AK.EQ.0.)AK=1.E-20
      BOUND=-D $AL=-ALAMBD $AR=-A
      IF(KFLAG.EQ.1 .AND.AL.GT.SET)GOTO74
      GOTO75
   74 WRITE(7,76)
   76 FORMAT(*WARNING---PROFILE BELOW DISLOCATION LINE*,/,
     1       *SOLUTION INVALID, TRY AGAIN*)
      GOTO63
   75 IF(LNUM.EQ.3)GOTO254
      IF(KFLAG.EQ.1)GOTO90
      RDIST=ABS(SET-(-A))
      IF(RDIST.LT.YINC)WRITE(7,91)
      IF(RDIST.LT.YINC)GOTO63
      GOTO92
   90 RDIST=ABS(SET-(-ALAMBD))
      IF(RDIST.LT.XINC)WRITE(7,91)
      IF(RDIST.LT.XINC)GOTO63
   91 FORMAT(*WARNING---PROFILE TOO CLOSE TO DISLOCATION CORE,*,/,
     1       *CHOOSE NEW PROFILE OR INCREASE GRID POINT DENSITY*)
   92 CONTINUE
  254 IF(KFLAG.EQ.1.AND.SET.GT.0.)WRITE(7,93)
      IF(KFLAG.EQ.1.AND.SET.GT.0.)GOTO63
   93 FORMAT(*WARNING---PROFILE ABOVE SURFACE OF HALF-SPACE,*,/,
     1       *SOLUTION INVALID, TRY AGAIN*)
      DO 400 LJR=1,NUM
      SETMU2=BGNMU2+(((ENDMU2-BGNMU2)/NUM)*(LJR-1))
      SETMU3=SETMU1
      IF(SETMU2.EQ.0.)SETMU2=1.E-10$PLOTX(LJR)=ALOG10(SETMU2/SETMU1)
      DO 230 K=1,2
      IF(K.EQ.2)GOTO231
```

```
117         AMU1=AMU2=AMU3=AMUHOM
118         GOTO221
119     231 AMU1=SETMU1 $AMU2=SETMU2 $AMU3=SETMU3
120     221 AM=(AMU3-AMU2)/(AMU3+AMU2)$IF(AM.EQ.0.)AM=1.E-20
121         AK=(AMU1-AMU2)/(AMU1+AMU2)$IF(AK.EQ.0.)AK=1.E-20
122         INUM=NUM $IF(LNUM.EQ.3)INUM=1
123         DO 200 J=1,INUM
124         IF(KFLAG.EQ.1.AND.LNUM.NE.3)X(J)=XMIN+(XINC*(J-1))
125         IF(KFLAG.EQ.1.AND.LNUM.NE.3)Y(J)=SET
126         IF(KFLAG.EQ.2.AND.LNUM.NE.3)X(J)=SET
127         IF(KFLAG.EQ.2.AND.LNUM.NE.3)Y(J)=YMIN+(YINC*(J-1))
128         IF(KFLAG.EQ.1.AND.LNUM.EQ.3)X(J)=COORD
129         IF(KFLAG.EQ.1.AND.LNUM.EQ.3)Y(J)=SET
130         IF(KFLAG.EQ.2.AND.LNUM.EQ.3)X(J)=SET
131         IF(KFLAG.EQ.2.AND.LNUM.EQ.3)Y(J)=COORD
132         IF(X(J).EQ.AP.AND.Y(J).EQ.AL)GOTO32
133         GO TO 33
134      32 X(J)=X(J)+1.E-10 $Y(J)=Y(J)+1.E-10
135      33 IF(X(J) .LT.BOUND)GOTO103
136         IF(X(J).GE.BOUND.AND.X(J).LE.0.)GOTO102
137     101 IF(IFLAG .EQ. 2)GOTO12
138         SUM1=SUM2=0.
139         DO 1 I=1,IUPPER
140         NU=I-1
141         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
142         TERM1=((AM*AK)**NU)*F1
143         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,I,D,ALAMBD,F2)
144         TERM2=  (AM**I)*(AK**NU)  *F2
145         SUM1=SUM1+TERM1     $SUM2=SUM2+TERM2
146     1   CONTINUE
147         AMATRX(J,1,K)=(((-AMU1*B)/(2.*PI))*(1.-AK))*(SUM1+SUM2) $GOTO200
148      12 SUM1=SUM2=SUM3=SUM4=0.
149         DO 5 I=1,IUPPER
150         NU=I-1
151         CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
152         CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMBD,F2)
153         CALL FUNC1(X(J),-A,I,D,Y(J),ALAMBD,F3)
154         CALL FUNC1(X(J),-A,I,D,Y(J),-ALAMBD,F4)
155         TERM1=((AM*AK)**NU)*F1 $TERM2=((AM*AK)**NU)*F2
156         TERM3=(AM**I)*(AK**NU)*F3 $TERM4=(AM**I)*(AK**NU)*F4
157         SUM1=SUM1+TERM1 $SUM2=SUM2+TERM2 $SUM3=SUM3+TERM3
158         SUM4=SUM4+TERM4
159      5  CONTINUE
160         AMATRX(J,1,K)=((AMU1*B)*(1.-AK)/(2.*PI))*(SUM1+SUM3-SUM2-SUM4)
161         GOTO200
162     102 IF(IFLAG.EQ.2)GOTO13
163         SUM1=0.
164         DO 7 NU=1,IUPPER
165         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
166         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMBD,F2)
167         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,NU,D,ALAMBD,F3)
168         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMBD,F4)
169         TERM=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4) $SUM1=SUM1+TERM
170       7 CONTINUE
171         CALL FUNC3(Y(J),ALAMBD,X(J),A,F1)
172         CALL FUNC3(Y(J),ALAMBD,X(J),-A,F2)
173         TERM2=F1+(AK*F2)
174         AMATRX(J,1,K)=((-AMU2*B)/(2.*PI))*(TERM2+SUM1) $GOTO200
```

```
      13 SUM1=SUM2=0.
         DO 3 NU=1,IUPPER
         CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
         CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMED,F2)
         CALL FUNC1(X(J),-A,NU,D,Y(J),ALAMED,F3)
         CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMED,F4)
         TERM1=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4)
         SUM1=SUM1+TERM1
         CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMED,F5)
         CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F6)
         CALL FUNC1(X(J),-A,NU,D,Y(J),-ALAMED,F7)
         CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMED,F8)
         TERM2=((AM*AK)**NU)*(F5+(AK*F6)+((1./AK)*F7)+F8)
         SUM2=SUM2+TERM2
       3 CONTINUE
         CALL FUNC3(X(J),A,Y(J),ALAMBD,F1)
         CALL FUNC3(X(J),-A,Y(J),ALAMBD,F2)
         CALL FUNC3(X(J),A,Y(J),-ALAMBD,F3)
         CALL FUNC3(X(J),-A,Y(J),-ALAMBD,F4)
         AMATRX(J,1,K)=((AMU2*B)/(2.*PI))*((F1+(AK*F2)+SUM1)-
        1 (F3+(AK*F4)+SUM2))
         GO TO 200
     103 IF(IFLAG.EQ.2)GOTO14
         SUM1=0.
         DO 9 I=1,IUPPER
         NU=I-1
         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMED,F1)
         CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMED,F2)
         TERM=(F1+(AK*F2))*((AK*AM)**NU) $SUM1=SUM1+TERM
       9 CONTINUE
         AMATRX(J,1,K)=((-AMU3*B)/(2.*PI))*(1.-AM)*SUM1 $GOTO200
      14 SUM1=SUM2=0.
         DO 11 I=1,IUPPER
         NU=I-1
         CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMED,F1)
         CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMED,F2)
         TERM1=((AM*AK)**NU)*(F1+(AK*F2))
         SUM1=SUM1+TERM1
         CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMED,F1)
         CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F2)
         TERM2=((AM*AK)**NU)*(F1+(AK*F2)) $SUM2=SUM2+TERM2
      11 CONTINUE
         AMATRX(J,1,K)=((AMU3*B)/(2.*PI))*(1.-AM)*(SUM1-SUM2)
     200 CONTINUE
     230 CONTINUE
         DO 204 I=1,INUM
         IF(AMATRX(I,1,1).EQ.0.)AMATRX(I,1,1)=1.E-10
     204 PLOTY(I)=AMATRX(I,1,2)/AMATRX(I,1,1)
         STSMID=PLOTY(1) $SUP=SDN=STSMID
         IF(LNUM.EQ.3)GOTO255 $IF(LNUM.EQ.4)GOTO257
         GOTO258
     257 DO 259 I=1,NUM
         IF(KFLAG.EQ.1)RD=ABS(X(I)-COORD)
         IF(KFLAG.EQ.1.AND.RD.LE.XINC)LI2=I
         IF(KFLAG.EQ.2)RD=ABS(Y(I)-COORD)
     259 IF(KFLAG.EQ.2.AND.RD.LE.YINC)LI2=I
         STRSMX(LJR)=PLOTY(LI2) $GOTO255
     258 DO 26 I=1,NUM
```

```
233          IF(PLOTY(I).GT.SUP)SUP=PLOTY(I)
234      26  IF(PLOTY(I).LT.SDN)SDN=PLOTY(I)
235          IF(LNUM.EQ.1)STRSMX(LJR)=SDN
236          IF(LNUM.EQ.2)STRSMX(LJR)=SUP
237      255 IF(LNUM.EQ.3)STRSMX(LJR)=SUP
238          DO 210 K=1,2
239          IF(K.EQ.2)GOTO211
240          AMU1=AMU2=AMU3=AMUHOM
241          GOTO212
242      211 AMU1=SETMU1 $AMU2=SETMU2 $AMU3=SETMU3
243      212 DO 50 I=1,INUM
244          IF(X(I).LT.BOUND)AMU=AMU3
245          IF(X(I).GE.BOUND.AND.X(I).LE.0.)AMU=AMU2
246          IF(X(I).GT.0.)AMU=AMU1
247          IF(AMU.EQ.0.)AMU=1.E-10
248      50  AMATRX(I,2,K)=(AMATRX(I,1,K)/AMU)*1.E+06
249      210 CONTINUE
250          DO 213 I=1,INUM
251          IF(AMATRX(I,2,1).EQ.0.)AMATRX(I,2,1)=1.E-10
252      213 PLOTY(I)=AMATRX(I,2,2)/AMATRX(I,2,1)
253          STSMID=PLOTY(1) $SUP=SDN=STSMID
254          IF(LNUM.EQ.3)GOTO256
255          IF(LNUM.EQ.4)STRNMX(LJR)=PLOTY(LI2)
256          IF(LNUM.EQ.4)GOTO256
257          DO 51 I=1,NUM
258          IF(PLOTY(I).GT.SUP)SUP=PLOTY(I)
259      51  IF(PLOTY(I).LT.SDN)SDN=PLOTY(I)
260          IF(LNUM.EQ.1)STRNMX(LJR)=SDN $IF(LNUM.EQ.2)STRNMX(LJR)=SUP
261      256 IF(LNUM.EQ.3)STRNMX(LJR)=SUP
262          IF(KFLAG.NE.IFLAG)GOTO400
263          IF(KFLAG.EQ.1.AND.LNUM.NE.3)DINC=XINC*1.E+06
264          IF(KFLAG.EQ.2.AND.LNUM.NE.3)DINC=YINC*1.E+06
265          IF(LNUM.EQ.3)GOTO400
266          DO 58 K=1,2
267          SUM=0.
268          DO 214 I=1,NUM
269          TERM=2.*AMATRX(I,2,K)*DINC*1.E-06
270          SUM=SUM+TERM
271      214 AMATRX(I,3,K)=SUM
272          MCHECK=0
273          DO 789 M=1,NUM
274          IF(KFLAG.EQ.1)GRD=ABS((XMAX-XMIN)/NUM)
275          IF(KFLAG.EQ.2)GRD=ABS((YMAX-YMIN)/NUM)
276          IF(KFLAG.EQ.1)GOTO790
277          GRD1=-ALAMBD+(GRD/100.) $GRD2=-ALAMBD-(GRD/100.)
278          IF(Y(M).GT.GRD2.AND.Y(M).LT.GRD1)KMID=M
279          IF(Y(M).GT.GRD2.AND.Y(M).LT.GRD1)MCHECK=1
280          GOTO789
281      790 GRD1=-A+(GRD/100.) $GRD2=-A-(GRD/100.)
282          IF(X(M).GT.GRD2.AND.X(M).LT.GRD1)MCHECK=1
283          IF(X(M).GT.GRD2.AND.X(M).LT.GRD1)KMID=M
284      789 CONTINUE
285          IF(MCHECK.EQ.0)STSMID=((AMATRX(NUM,3,K)-AMATRX(1,3,K))/2.)
286         1  +AMATRX(1,3,K)
287          IF(MCHECK.EQ.1)STSMID=AMATRX(KMID,3,K)
288          DO 73 I=1,NUM
289       73 AMATRX(I,3,K)=AMATRX(I,3,K)-STSMID
290       58 CONTINUE
```

```
      DO 215 I=1,NUM
      IF(AMATRX(I,3,1).EQ.0.)AMATRX(I,3,1)=1.E-10
  215 PLOTY(I)=AMATRX(I,3,2)/AMATRX(I,3,1)
      STSMID=PLOTY(1) $SUP=SDN=STSMID
      IF(LNUM.EQ.4)DISPMX(LJR)=PLOTY(LI2)
      IF(LNUM.EQ.4)GOTO260
      DO 59 I=1,NUM
      IF(PLOTY(I).GT.SUP)SUP=PLOTY(I)
   59 IF(PLOTY(I).LT.SDN)SDN=PLOTY(I)
      IF(LNUM.EQ.1)DISPMX(LJR)=SDN
  260 IF(LNUM.EQ.2)DISPMX(LJR)=SUP
  400 CONTINUE
      STSMID=STRSMX(1)
      SUP=SDN=STSMID
      DO 28 I=1,NUM
      IF(STRSMX(I).GT.SUP)SUP=STRSMX(I)
   28 IF(STRSMX(I).LT.SDN)SDN=STRSMX(I)
      WRITE(7,29)SDN,SUP
   29 FORMAT(*MIN/MAX VALUES OF NORMALIZED STRESS*,/,
     1       *VERSUS SHEAR MODULUS CONTRAST*,/,
     1        E20.4, 5X, E20.4)
  417 LU=7 $LNLGX=1$LNLGY=1$NCLX=2$NCLY=2$WRITE(7,34)
   34 FORMAT(*PLOT TITLE, 80 CHARACTERS MAXIMUM*)
      READ(7,35)(LTITL(JM),JM=1,8)
   35 FORMAT(8A10)
      IF(IFLAG.EQ.1)LTITL2(1)=10HXZ STRESS-
      IF(IFLAG.EQ.2)LTITL2(2)=10HYZ STRESS-
      LTITL2(2)=10HNORMALIZED
      DO 36 I=3,8
   36 LTITL2(I)=10H
      KX(1)=10HLOG(MU2/MU $KY(1)=10HNORMALIZED
      KX(2)=10H1)            $KY(2)=10H STRESS
      XRT=PLOTX(NUM) $XLT=PLOTX(1)
      MAJX=5$MAJY=10$YLO=SDN$YUP=SUP$AA=YUP
      IF(YLO.EQ.AA)YUP=YUP+1$IF(YLO.EQ.AA)YLO=YLO-1
      WRITE(7,37)
   37 FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
      READ(7,38)CHARAC
   38 FORMAT(A1)
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO39
      WRITE(7,40)
   40 FORMAT(*MIN/MAX Y VALUES*)
      CALL GETNUM(AB) $YLO=AB(1)$YUP=AB(2)
   39 WRITE(7,41)
   41 FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
      READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO42
      WRITE(7,43)
   43 FORMAT(*MIN/MAX X VALUES*)
      CALL GETNUM(AB)$XLT=AB(1)$XRT=AB(2)
   42 WRITE(7,44)
   44 FORMAT(*SKIP?*)
      READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )CALL PLOTS(STRSMX,PLOTX,1,NUM)
      WRITE(7,46)
   46 FORMAT(*0=RE-START, 1=NEW PLOT*)
      READ(7,71)LCHECK
   71 FORMAT(I1)
```

```
      IF(LCHECK.EQ.0)GOTO63 $IF(LCHECK.EQ.1)GOTO417
      STSMID=PLCTY(1) $SUP=SDN=STSMID
      DO 52 I=1,NUM
      IF(STRNMX(I).GT.SUP)SUP=STRNMX(I)
   52 IF(STRNMX(I).LT.SDN)SDN=STRNMX(I)
      IF(IFLAG.EQ.1)CD=2HXZ $IF(IFLAG.EQ.2)CD=2HYZ
      WRITE(7,53)CD,SDN,SUP
   53 FORMAT(*MIN/MAX VALUES OF NORMALIZED *,A2,* STRAIN*,/,
     1       *VERSUS SHEAR MODULUS CONTRAST*,/,E20.4,5X,E20.4)
  418 KY(2)=10H STRAIN    $YLO=SDN $YUP=SUP$AA=YUP
      IF(YLO.EQ.AA)YUP=YUP+1 $IF(YLO.EQ.AA)YLO=YLO-1
      IF(IFLAG.EQ.1)LTITL2(1)=10HXZ STRAIN-
      IF(IFLAG.EQ.2)LTITL2(1)=10HYZ STRAIN-
      WRITE(7,37)$READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO550
      WRITE(7,40)$CALL GETNUM(AB)$YLO=AB(1)$YUP=AB(2)
  550 WRITE(7,41)$READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN .OR.CHARAC.EQ.1H )GOTO55
      WRITE(7,43)$CALL GETNUM(AB)$XLT=AB(1)$XRT=AB(2)
   55 WRITE(7,44)$READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN .OR.CHARAC.EQ.1H )CALL PLOTS(STRNMX,PLOTX,1,NUM)
   56 WRITE(7,46)$CALL GETNUM(AB)$LCHECK=AB(1)
      IF(LCHECK.EQ.0)GOTO63$IF(LCHECK.EQ.1)GOTO418
      IF(KFLAG.NE.IFLAG)WRITE(7,57)
   57 FORMAT(*NOTE--ONLY W(X) VS. X OR W(X) VS. Y IS COMPUTED*)
      IF(KFLAG.NE.IFLAG)GOTO63
      STSMID=DISPMX(1) $IF(LNUM.EQ.3)GOTO63
      SUP=SDN=STSMID
      DO 45 I=1,NUM
      IF(DISPMX(I).GT.SUP)SUP=DISPMX(I)
   45 IF(DISPMX(I).LT.SDN)SDN=DISPMX(I)
      WRITE(7,60)CD,SDN,SUP
   60 FORMAT(*MIN/MAX VALUES OF NORMALIZED DISPLACEMENT*,/,
     1       *VERSUS SHEAR MODULUS CONTRAST, COMPUTED*,/,
     1       *FROM *,A2,* STRAIN*,/,E20.4,5X,E20.4)
  419 KY(1)=10HDISPLACEME $KY(2)=10HNT
      YLO=SDN$YUP=SUP$AA=YUP$IF(YLO.EQ.AA)YLO=YUP+1
      IF(YLO.EQ.AA)YLO=YLO-1$WRITE(7,37)$READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO620
      WRITE(7,40)$CALL GETNUM(AB) $YLO=AB(1)$YUP=AB(2)
  620 WRITE(7,41)$READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO62
      WRITE(7,43)$CALL GETNUM(AB)$XLT=AB(1)$XRT=AB(2)
   62 WRITE(7,44)$READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )CALL PLOTS(DISPMX,PLOTX,1,NUM)
      WRITE(7,46)$CALL GETNUM(AB)$LCHECK=AB(1)
      IF(LCHECK.EQ.1)GOTO419
   63 WRITE(7,450)SETMU1,BGNMU2,ENDMU2,SETMU3
      WRITE(7,451)AMUHOM,AM,AK
      IF(IFLAG.EQ.1)CD1=2HXZ $IF(KFLAG.EQ.1)CD2=1HX
      IF(IFLAG.EQ.2)CD1=2HYZ $IF(KFLAG.EQ.2)CD2=1HY
      WRITE(7,49)CD1,CD2
  450 FORMAT(*MU1      = *,E20.4,5X,*MU2LOW = *,E20.4,/,
     1       *MU2HIGH = *,E20.4,5X,*MU3    = *,E20.4)
  451 FORMAT(*MUH      = *,E20.4,5X,*M       = *,E20.4,/,
     1       *K        = *,E20.4)
      BB=B*1.E+06
      WRITE(7,47)A,ALAMBD,D,BB
```

```
      47  FORMAT(*DISLOCATION POSITICN = -*,F10.3,*, -*,F10.3,* KMS.*,/,
     1        *SLAB THICKNESS       =  *,F10.3,* KMS.*,/,
     1        *BURGERS VECTOR       =  *,F10.3,* MMS.*)
      49  FORMAT(A2,* STRESS/STRAIN VS. *,A2)
          WRITE(7,64)
      64  FORMAT(*0=RE-START*,/,*1=STOP*,/,
     1        *2=RE-START WITH SAME VALUES, DIFFERENT PROFILE*)
          CALL GETNUM(AB) $LJK=AB(1)
          IF(LJK.EQ.0)GOTO48 $IF(LJK.EQ.2)GC TO 70
          STOP $END
          SUBROUTINE FUNC1(P,A,NU,D,G,ALAMBD,F)
          R=(((P+A+(2.*NU*D))**2)+((G+ALAMBD)**2))
          IF(R.EQ.0.)R=1.E-10$F=(P+A+(2.*NU*D))/R
          RETURN $END
          SUBROUTINE FUNC2(P,Q,R,S,A,NU,D,ALAMBD,F)
          T=(((R+A+(2.*NU*D))**2)+((S+ALAMBD)**2))
          IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
          RETURN $END
          SUBROUTINE FUNC3(P,Q,R,S,F)
          T=(((R+S)**2)+((P+Q)**2))
          IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
          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 $GCTC(1,2,3,5,1,4,3,4)K
        1 N=N*10+D-27  $S=4  $GOTC5
        2 M=-1 $S=4 $GOTO5
        3 P=I $S=4 $GOTO5
        4 IF(P.NE.0)F=10.**(I-P-1) $R(J)=N/F*M $GOTO6
        9 FORMAT(80R1)
          END
```


3333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333
3333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333
7777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777

LISTING OF DBLNRM

```
05 MAY 76    10.00.27          MCHUGH .DBLNRM

  1  DELETE(LGO,OUTPUT,DBLNRM)
  2  DBLNRM.
  3  CXIT.
  4  LIBCOPY(GRAPHIC,TXLGO/RR,TXLGO)
  5  LIBCOPY(JDRAT,NPLGO/RR,NPLGO)
  6  DELETE(LGO,OUTPUT,DBLNRM)
  7  RUN76(S)
  8  LINK(F=LGO,F=TXLGO,F=NPLGO,B=DBLNRM)
  9  DBLNRM.
 10  FIN.
 11  EOR
 12        PROGRAM DBLNRM(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
 13        COMMON/TVPOCL/TVPOL(8)
 14        COMMON/TVTUNE/ITUNE(30)
 15        COMMON/JPLOT/XLT,XRT,YLO,YUP,MAJX,MAJY,KX(2),KY(2),
 16       1LTITL(8),LU,LTF,LNLGX,LNLGY,NCLX,NCLY,LTITL2(8)
 17        DIMENSION IFET(8)
 18        DIMENSION X(200),Y(200),SXZ(200),SYZ(200),AB(30)
 19        DIMENSION PLOTY(200),PLOTX(200)
 20        DIMENSION CM(200,2,2)
 21        DATA YMIN,YMAX,XCUT,CUTMAX/-1.E+10,-1.E+10,-1.E+10,-1.E+10/
 22        CALL FET(5LTAPE7,IFET,8)
 23        IFET(2)=IFET(2).OR.0000 0010 0000 0000 0000B
 24        IFET(8)=IFET(8).OR.4000 0000 0000 0000 0000B
 25        CALL FET(5LTAPE7,IFET,-8)
 26    48  WRITE(7,15)
 27    15  FORMAT(*SHEAR MODULI--IN CGS UNITS--MU1,MU2,MU3*)
 28        CALL GETNUM(AB) $IAB2=AB(2) $IAB4=AB(4) $IAB6=AB(6)
 29        AMU1=AB(1)*(10.**IAB2) $AMU2=AB(3)*(10.**IAB4)
 30        AMU3=AB(5)*(10.**IAB6) $SETMU1=AMU1 $SETMU2=AMU2 $SETMU3=AMU3
 31        WRITE(7,95)
 32    95  FORMAT(*SHEAR MODULUS OF HOMOGENEOUS HALF-SPACE*)
 33        CALL GETNUM(AB) $AMUHOM=AB(1)*(10.**AB(2))
 34        WRITE(7,16)
 35    16  FORMAT(*POSITION OF DISLOCATION--IN KMS., (A,LAMBDA)*,/,
 36       1         *NOTE---A>0., LAMBDA>0.*)
 37        CALL GETNUM(AB) $A=AB(1) $ALAMBD=AB(2) $WRITE(7,17)
 38    17  FORMAT(*THICKNESS OF SLAB, IN KMS.--D>0.*)
 39        CALL GETNUM(AB) $D=AB(1)  $WRITE(7,900)
 40    900 FORMAT(*WIDTH OF SLIP PLANE, IN KMS.--W>0.*)
 41        CALL GETNUM(AB)$WIDTH=AB(1)$WRITE(7,18)
 42    18  FORMAT(*BURGERS VECTOR, IN MMS.--B*,/,
 43       1       *B<0 FOR RIGHT LATERAL SLIP*)
 44        CALL GETNUM(AB) $B=-AB(1)*1.E-06
 45    70  WRITE(7,19)
 46    19  FORMAT(*1=AMPLITUDE VS. X, 2=AMPLITUDE VS. Y*)
 47        CALL GETNUM(AB) $KFLAG=AB(1)$WRITE(7,20)
 48    20  FORMAT(*1=XZ STRESS, 2=YZ STRESS*)
 49        CALL GETNUM(AB)  $IFLAG=AB(1)
 50        IF(KFLAG.EQ.2)GOTO201
 51        WRITE(7,21) $CALL GETNUM(AB) $XMIN=AB(1)
 52        XMAX=AB(2) $ WRITE(7,22)  $CALL GETNUM(AB) $SET=-AB(1)
 53        GO TO 202
 54    201 WRITE(7,23) $CALL GETNUM(AB)
 55        YMAX=-AB(1) $YMIN=-AB(2) $WRITE(7,24)
 56        CALL GETNUM(AB) $SET=AB(1)
 57    21  FORMAT(*ENTER MINIMUM AND MAXIMUM X VALUES, IN KMS.*)
 58    22  FORMAT(*DEPTH OF PROFILE, IN KMS.*)
```

```
 59     23  FORMAT(*ENTER MINIMUM AND MAXIMUM Y VALUES, IN KMS.*)
 60      24 FORMAT(*X COORDINATE OF PROFILE, IN KMS.*)
 61     202 NUM=200 $IUPPER=20 $PI=3.14159
 62         IF(KFLAG.EQ.1)XINC=(XMAX-XMIN)/NUM
 63         IF(KFLAG.EQ.2)YINC=(YMAX-YMIN)/NUM
 64         WRITE(7,25) NUM,IUPPER
 65      25 FORMAT(*NOTE--NUMBER OF DATA POINTS = *,I5,/,
 66        1       *      UPPER LIMIT ON SERIES = *,I5)
 67         AM=(AMU3-AMU2)/(AMU3+AMU2)$IF(AM.EQ.0.)AM=1.E-20
 68         AK=(AMU1-AMU2)/(AMU1+AMU2) $IF(AK.EQ.0.)AK=1.E-20
 69         BOUND=-D $AL=-ALAMBD $AR=-A
 70         IF(KFLAG.EQ.1 .AND.AL.GT.SET)GOTO74
 71         GOTO75
 72      74 WRITE(7,76)
 73      76 FORMAT(*WARNING---PROFILE BELOW DISLOCATION LINE*,/,
 74        1       *SOLUTION INVALID, TRY AGAIN*)
 75         GOTO63
 76      75 IF(KFLAG.EQ.1)GOTO90
 77         RDIST=ABS(SET-(-A))
 78         IF(RDIST.LT.YINC)WRITE(7,91)
 79         IF(RDIST.LT.YINC)GOTO63
 80         GOTO92
 81      90 RDIST=ABS(SET-(-ALAMBD))
 82         IF(RDIST.LT.XINC)WRITE(7,91)
 83         IF(RDIST.LT.XINC)GOTO63
 84      91 FORMAT(*WARNING---PROFILE TOO CLOSE TO DISLOCATION CORE,*,/,
 85        1       *CHOOSE NEW PROFILE OR INCREASE GRID POINT DENSITY*)
 86      92 CONTINUE
 87         IF(KFLAG.EQ.1.AND.SET.GT.0.)WRITE(7,93)
 88         IF(KFLAG.EQ.1.AND.SET.GT.0.)GOTO63
 89      93 FORMAT(*WARNING---PROFILE ABOVE SURFACE OF HALF-SPACE,*,/,
 90        1       *SOLUTION INVALID, TRY AGAIN*)
 91         DO 230 K=1,2
 92         IF(K.EQ.2)GOTO231
 93         AMU1=AMU2=AMU3=AMUHOM
 94         GOTO221
 95     231 AMU1=SETMU1 $AMU2=SETMU2 $AMU3=SETMU3
 96     221 AM=(AMU3-AMU2)/(AMU3+AMU2) $IF(AM.EQ.0.)AM=1.E-20
 97         AK=(AMU1-AMU2)/(AMU1+AMU2) $IF(AK.EQ.0.)AK=1.E-20
 98         IF(K.EQ.2)ALAMBD=ALAMBD-WIDTH
 99         IF(K.EQ.2)B=-B
100         DO 910 MK=1,2
101         IF(MK.EQ.2)ALAMBD=ALAMBD+WIDTH
102         IF(MK.EQ.2)B=-B
103         DO 200 J=1,NUM
104         IF(KFLAG.EQ.1)X(J)=XMIN+(XINC*(J-1))
105         IF(KFLAG.EQ.1) Y(J)=SET
106         IF(KFLAG.EQ.2)X(J)=SET
107         IF(KFLAG.EQ.2)Y(J)=YMIN+(YINC*(J-1))
108         IF(X(J).EQ.AR.AND.Y(J).EQ.AL)GOTO32
109         GO TO 33
110      32 X(J)=X(J)+1.E-10 $Y(J)=Y(J)+1.E-10
111      33 IF(X(J) .LT.BOUND)GOTO103
112         IF(X(J).GE.BOUND.AND.X(J).LE.0.)GOTO102
113     101 IF(IFLAG .EQ. 2)GOTO12
114         SUM1=SUM2=0.
115         DO 1 I=1,IUPPER
116         NU=I-1
```

```
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
      TERM1=((AM*AK)**NU)*F1
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,I,D,ALAMBD,F2)
      TERM2=  (AM**I)*(AK**NU)  *F2
      SUM1=SUM1+TERM1     $SUM2=SUM2+TERM2
    1 CONTINUE
      SXZ(J)=(((-AMU1*B)/(2.*PI))*(1.-AK))*(SUM1+SUM2) $GOTO200
   12 SUM1=SUM2=SUM3=SUM4=0.
      DO 5 I=1,IUPPER
      NU=I-1
      CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
      CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMBD,F2)
      CALL FUNC1(X(J),-A,I,D,Y(J),ALAMBD,F3)
      CALL FUNC1(X(J),-A,I,D,Y(J),-ALAMBD,F4)
      TERM1=((AM*AK)**NU)*F1 $TERM2=((AM*AK)**NU)*F2
      TERM3=(AM**I)*(AK**NU)*F3 $TERM4=(AM**I)*(AK**NU)*F4
      SUM1=SUM1+TERM1 $SUM2=SUM2+TERM2 $SUM3=SUM3+TERM3
      SUM4=SUM4+TERM4
    5 CONTINUE
      SYZ(J)=((AMU1*B)*(1.-AK)/(2.*PI))*(SUM1+SUM3-SUM2-SUM4)
      GOTO200
  102 IF(IFLAG.EQ.2)GOTO13
      SUM1=0.
      DO 7 NU=1,IUPPER
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMBD,F2)
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,NU,D,ALAMBD,F3)
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMBD,F4)
      TERM=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4) $SUM1=SUM1+TERM
    7 CONTINUE
      CALL FUNC3(Y(J),ALAMBD,X(J),A,F1)
      CALL FUNC3(Y(J),ALAMBD,X(J),-A,F2)
      TERM2=F1+(AK*F2)
      SYZ(J)=((-AMU2*B)/(2.*PI))*(TERM2+SUM1) $GOTO200
   13 SUM1=SUM2=0.
      DO 3 NU=1,IUPPER
      CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
      CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMBD,F2)
      CALL FUNC1(X(J),-A,NU,D,Y(J),ALAMBD,F3)
      CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMBD,F4)
      TERM1=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4)
      SUM1=SUM1+TERM1
      CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMBD,F5)
      CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F6)
      CALL FUNC1(X(J),-A,NU,D,Y(J),-ALAMBD,F7)
      CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMBD,F8)
      TERM2=((AM*AK)**NU)*(F5+(AK*F6)+((1./AK)*F7)+F8)
      SUM2=SUM2+TERM2
    3 CONTINUE
      CALL FUNC3(X(J),A,Y(J),ALAMBD,F1)
      CALL FUNC3(X(J),-A,Y(J),ALAMBD,F2)
      CALL FUNC3(X(J),A,Y(J),-ALAMBD,F3)
      CALL FUNC3(X(J),-A,Y(J),-ALAMBD,F4)
      SYZ(J)=((AMU2*B)/(2.*PI))*((F1+(AK*F2)+SUM1)-(F3+(AK*F4)+SUM2))
      GO TO 200
  103 IF(IFLAG.EQ.2)GOTO14
      SUM1=0.
      DO 9 I=1,IUPPER
```

```
      NU=I-1
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMBD,F1)
      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMBD,F2)
      TERM=(F1+(AK*F2))*((AK*AM)**NU) $SUM1=SUM1+TERM
    9 CONTINUE
      SXZ(J)=((-AMU3*B)/(2.*PI))*(1.-AM)*SUM1 $GOTO200
   14 SUM1=SUM2=0.
      DO 11 I=1,IUPPER
      NU=I-1
      CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMBD,F1)
      CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMBD,F2)
      TERM1=((AM*AK)**NU)*(F1+(AK*F2))
      SUM1=SUM1+TERM1
      CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMBD,F1)
      CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F2)
      TERM2=((AM*AK)**NU)*(F1+(AK*F2)) $SUM2=SUM2+TERM2
   11 CONTINUE
      SYZ(J)=((AMU3*B)/(2.*PI))*(1.-AM)*(SUM1-SUM2)
  200 CONTINUE
      DO 920 IMK=1,NUM
      IF(IFLAG.EQ.1)CM(IMK,MK,K)=SXZ(IMK)
  920 IF(IFLAG.EQ.2)CM(IMK,MK,K)=SYZ(IMK)
  910 CONTINUE
  230 CONTINUE
      DO 930 I=1,NUM
      IF(IFLAG.EQ.1)SXZ(I)=(CM(I,1,2)+CM(I,2,2))/(CM(I,1,1)+CM(I,2,1))
  930 IF(IFLAG.EQ.2)SYZ(I)=(CM(I,1,2)+CM(I,2,2))/(CM(I,1,1)+CM(I,2,1))
      IF(IFLAG.EQ.1)STSMID=SXZ(1)
      IF(IFLAG.EQ.2)STSMID=SYZ(1)
      SUP=SDN=STSMID
      DO 26 I=1,NUM
      IF(IFLAG.EQ.1)GOTO27
      IF(SYZ(I).GT.SUP)SUP=SYZ(I)
      IF(SYZ(I).LT.SDN)SDN=SYZ(I)
      GO TO 26
   27 IF(SXZ(I).GT.SUP)SUP=SXZ(I)
      IF(SXZ(I).LT. SDN)SDN=SXZ(I)
   26 CONTINUE
      IF(IFLAG.EQ.1)WRITE(7,28) $IF(IFLAG.EQ.2)WRITE(7,29)
   28 FORMAT(*MIN/MAX VALUES OF NORMALIZED XZ STRESS*)
   29 FORMAT(*MIN/MAX VALUES OF NORMALIZED YZ STRESS*)
      WRITE(7,30)SDN,SUP
   30 FORMAT(E15.4,5X,E15.4)
      LUP=0
      DO 31 I=1,NUM
      IF(IFLAG.EQ.1)PLOTY(I)=SXZ(I)
      IF(IFLAG.EQ.2)PLOTY(I)=SYZ(I)
      IF(KFLAG.EQ.1)PLOTX(I)=X(I)
   31 IF(KFLAG.EQ.2)PLOTX(I)=Y(I)
      IF(KFLAG.EQ.2)GOTO83
      GOTO85
   83 DO 81 I=1,NUM
      SXZ(I)=PLOTY(I)
   81 SYZ(I)=PLOTX(I)
      DO 84 I=1,NUM
      L=NUM-I+1
      PLOTY(L)=SXZ(I)
   84 PLOTX(L)=SYZ(I)
```

```
      DO 82 I=1,NUM
      IF(KFLAG.EQ.2.AND.PLOTX(I).LT.AL)LUP=LUP+1
   82 IF(LUP.EQ.1)XCUT=PLOTX(I-1)
      LDN=0
      DO 86 I=1,NUM
      L=NUM-I+1
      IF(KFLAG.EQ.2.AND.PLOTX(L).GT.0.)LDN=LDN+1
   86 IF(LDN.EQ.1)CUTMAX=PLOTX(L-1)
   85 CONTINUE
  417 LU=7 $LNLGX=1 $LNLGY=1 $NCLX=2 $NCLY=2 $WRITE(7,34)
   34 FORMAT(*PLOT TITLE, 80 CHARACTERS MAXIMUM*)
      READ(7,35)(LTITL(JM),JM=1,8)
   35 FORMAT(8A10)
      IF(IFLAG.EQ.1)LTITL2(2)=10H XZ STRESS
      IF(IFLAG.EQ.2)LTITL2(2)=10H YZ STRESS
      LTITL2(1)=10HNORMALIZED
      DO 36 I=3,8
   36 LTITL2(I)=10H
      IF(KFLAG.EQ.1)KX(1)=10HX (KMS.)
      IF(KFLAG.EQ.2)KX(1)=10HY (KMS.)
      KX(2)=10H             $KY(1)=10HNORMALIZED    $KY(2)=10H STRESS
      XRT=PLOTX(NUM) $XLT=PLOTY(1) $MAJX=5 $MAJY=10 $YLC=SDN $YUP=SUP
      IF(KFLAG.EQ.2.AND.YMIN.LT.AL)XRT=XCUT
      IF(KFLAG.EQ.2.AND.YMAX.GT.0.)XLT=CUTMAX
      IF(KFLAG.EQ.2.AND.YMIN.GT.AL)XRT=PLOTX(NUM)
      AA=YUP$IF(YLO.EQ.AA)YUP=YUP+1.$IF(YLO.EQ.AA)YLC=YLC-1.
      WRITE(7,37)
   37 FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
      READ(7,38)CHARAC
   38 FORMAT(A1)
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO39
      WRITE(7,40)
   40 FORMAT(*MIN/MAX Y VALUES*)
      CALL GETNUM(AB) $YLO=AB(1) $YUP=AB(2)
   39 WRITE(7,41)
   41 FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
      READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO42
      WRITE(7,43)
   43 FORMAT(*MIN/MAX X VALUES*)
      CALL GETNUM(AB) $IF(KFLAG.EQ.1)XLT=AB(1)
      IF(KFLAG.EQ.1)XRT=AB(2)
      IF(KFLAG.EQ.2)XLT=-AB(2)
      IF(KFLAG.EQ.2)XRT=-AB(1)
   42 WRITE(7,44)
   44 FORMAT(*SKIP?*)
      READ(7,38)CHARAC
      IF(CHARAC.EQ.1HN .OR. CHARAC.EQ.1H )CALL PLOTS(PLOTY,PLOTX,1,NUM)
      WRITE(7,46)
   46 FORMAT(*0=RE-START, 1=NEW PLOT*)
      READ(7,71)LCHECK
   71 FORMAT(I1)
      IF(LCHECK.EQ.0)GOTO63$IF(LCHECK.EQ.1)GOTO417
      DO 210 K=1,2
      IF(K.EQ.2)GOTO211
      AMU1=AMU2=AMU3=AMUHOM
      GOTO212
  211 AMU1=SETMU1 $AMU2=SETMU2 $AMU3=SETMU3
```

```
291     212 DO 50 I=1,NUM
292         IF(X(I).LT.BOUND)AMU=AMU3
293         IF(X(I).GE.BOUND.AND.X(I).LE.0.)AMU=AMU2
294         IF(X(I).GT.0.)AMU=AMU1
295         IF(AMU.EQ.0.)AMU=1.E-10
296         IF(K.EQ.2)SXZ(I)=((CM(I,1,2)+CM(I,2,2))/AMU)*1.E+06
297         IF(K.EQ.1)SYZ(I)=((CM(I,1,1)+CM(I,2,1))/AMU)*1.E+06
298      50 CONTINUE
299     210 CONTINUE
300         DO 987 I=1,NUM
301         PLOTY(I)=SXZ(I)/SYZ(I)
302     987 CONTINUE
303         STSMID=PLOTY(1)
304         SUP=SDN=STSMID
305         DO 51 I=1,NUM
306         IF(PLOTY(I).GT.SUP)SUP=PLOTY(I)
307      51 IF(PLOTY(I).LT.SDN)SDN=PLOTY(I)
308         IF(IFLAG.EQ.1)WRITE(7,52) $IF(IFLAG.EQ.2)WRITE(7,53)
309      52 FORMAT(*MIN/MAX VALUES OF NORMALIZED XZ STRAIN*)
310      53 FORMAT(*MIN/MAX VALUES OF NORMALIZED YZ STRAIN*)
311         WRITE(7,30)SDN,SUP
312         IF(IFLAG.EQ.1)LTITL2(2)=10H XZ STRAIN
313         IF(IFLAG.EQ.2)LTITL2(2)=10H YZ STRAIN
314     418 KY(1)=10HNORMALIZED       $KY(2)=10H STRAIN
315         YLO=SDN $YUP=SUP
316         AA=YUP $IF(YLO.EQ.AA)YUP=YUP+1. $IF(YLO.EQ.AA)YLO=YLO-1.
317         WRITE(7,37)  $READ(7,38)CHARAC
318         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO55
319         WRITE(7,40) $CALL GETNUM(AB) $YLO=AB(1) $YUP=AB(2)
320      55 WRITE(7,41) $READ(7,38)CHARAC
321         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO301
322         WRITE(7,43) $CALL GETNUM(AB)
323         IF(KFLAG.EQ.1)XLT=AB(1)  $IF(KFLAG.EQ.1)XRT=AB(2)
324         IF(KFLAG.EQ.2)XLT=-AB(2)  $IF(KFLAG.EQ.2)XRT=-AB(1)
325     301 WRITE(7,44) $READ(7,38)CHARAC
326         IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )CALL PLOTS(PLOTY,PLOTX,1,NUM)
327      56 WRITE(7,46) $CALL GETNUM(AB)$LCHECK=AB(1)
328         IF(LCHECK.EQ.0)GOTO63$IF(LCHECK.EQ.1)GOTO418
329         IF(KFLAG.NE.IFLAG)WRITE(7,57)
330      57 FORMAT(*NOTE--ONLY W(X) VS. X OR W(Y) VS. Y IS COMPUTED*)
331         IF(KFLAG.NE.IFLAG)GOTO63
332         IF(KFLAG.EQ.1)DINC=XINC*1.E+06
333         IF(KFLAG.EQ.2)DINC=YINC*1.E+06
334         SUM1=SUM2=0.
335         DO 58 I=1,NUM
336         TERM1=2.*SXZ(I)*DINC*1.E-06 $TERM2=2.*SYZ(I)*DINC*1.E-06
337         SUM1=SUM1+TERM1 $SUM2=SUM2+TERM2
338         SXZ(I)=SUM1 $SYZ(I)=SUM2
339      58 CONTINUE
340         MCHECK=0
341         DO 789 M=1,NUM
342         IF(KFLAG.EQ.1)GRD=ABS((XMAX-XMIN)/NUM)
343         IF(KFLAG.EQ.2)GRD=ABS((YMAX-YMIN)/NUM)
344         IF(KFLAG.EQ.1)GOTO790
345         GRD1=-ALAMBD+(GRD/100.)  $GRD2=-ALAMBD-(GRD/100.)
346         IF(Y(M).GT.GRD2.AND.Y(M).LT.GRD1)KMID=M
347         IF(Y(M).GT.GRD2.AND.Y(M).LT.GRD1)MCHECK=1
348         GOTO789
```

```
 790 GRD1=-A+(GRD/100.)  $GRD2=-A-(GRD/100.)
     IF(X(M).GT.GRD2.AND.X(M).LT.GRD1)KMID=M
     IF(X(M).GT.GRD2.AND.X(M).LT.GRD1)MCHECK=1
 789 CONTINUE
     IF(MCHECK.EQ.0)S1MID=((SXZ(NUM)-SXZ(1))/2.)+SXZ(1)
     IF(MCHECK.EQ.0)S2MID=((SYZ(NUM)-SYZ(1))/2.)+SYZ(1)
     IF(MCHECK.EQ.1)S1MID=SXZ(KMID)   $IF(MCHECK.EQ.1)S2MID=SYZ(KMID)
     DO 73 I=1,NUM
     SXZ(I)=SXZ(I)-S1MID $SYZ(I)=SYZ(I)-S2MID
     IF(SYZ(I).EQ.0.)SYZ(I)=1.E-20
  73 PLOTY(I)=SXZ(I)/SYZ(I)
     STSMID=PLOTY(1)
     SUP=SDN=STSMID
     DO 59 I=1,NUM
     IF(PLOTY(I).GT.SUP)SUP=PLOTY(I)
  59 IF(PLOTY(I).LT.SDN)SDN=PLOTY(I)
     IF(WIDTH.LT.1.E+05)WRITE(7,600)
 600 FORMAT(*vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv*,/,
    1        *.......................................................*,/,
    1        *WARNING--CAUTION--WARNING*,/,
    1        *NORMALIZED DISPLACEMENTS MAY BE INCORRECT*,/,
    1        *CHECK DBLNRM DOCUMENTATION FOR EXPLANATION*,/,
    1        *.......................................................*,/,
    1        *vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv*)
     IF(IFLAG.EQ.1) WRITE(7,60) $IF(IFLAG.EQ.2)WRITE(7,61)
     IF(KFLAG.EQ.1.AND.SLT.NE.0.)WRITE(7,87)
  87 FORMAT(*CAUTION---THIS IS NOT THE TOTAL DISPLACEMENT*)
  60 FORMAT(*MIN/MAX VALUES OF NORMALIZED W(X) VS. X*)
  61 FORMAT(*MIN/MAX VALUES OF NORMALIZED W(Y) VS. Y*)
     WRITE(7,30) SDN,SUP
 419 IF(IFLAG.EQ.1)LTITL2(2)=10H W(X).VS.X
     IF(IFLAG.EQ.2)LTITL2(2)=10H W(Y).VS.Y
     KY(1)=10HDISPLACEME     $KY(2)=10HNT
     YLO=SDN  $YUP=SUP$AA=YUP$IF(YLO.EQ.AA)YUP=YUP+1.
     IF(YLO.EQ.AA)YLO=YLO-1.
     WRITE(7,37) $READ(7,38)CHARAC
     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO62
     WRITE(7,40) $CALL GETNUM(AB) $YLO=AB(1) $YUP=AB(2)
  62 WRITE(7,41) $READ(7,38)CHARAC
     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO303
     WRITE(7,43) $CALL GETNUM(AB)
     IF(KFLAG.EQ.1)XLT=AB(1)   $IF(KFLAG.EQ.1)XRT=AB(2)
     IF(KFLAG.EQ.2)XLT=-AB(2)   $IF(KFLAG.EQ.2)XRT=-AB(1)
 303 WRITE(7,44) $READ(7,38)CHARAC
     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H ) CALL PLOTS(PLOTY,PLOTX,1,NUM)
     WRITE(7,46)$CALL GETNUM(AB)$LCHECK=AB(1)$IF(LCHECK.EQ.1)GOTO419
  63 ALAMBD=ALAMBD-WIDTH $BB=-B
     AM=(SETMU3-SETMU2)/(SETMU3+SETMU2)
     AK=(SETMU1-SETMU2)/(SETMU1+SETMU2)
     WRITE(7,45)AMU1,AMU2,AMU3,AM,AK
     BB=B*1.E+06
     WRITE(7,47)A,ALAMBD,WIDTH,D,BB
     IF(IFLAG.EQ.1)CD1=2HXZ$IF(KFLAG.EQ.1)CD2=1HX
     IF(IFLAG.EQ.2)CD1=2HYZ$IF(KFLAG.EQ.2)CD2=1HY
     WRITE(7,49)CD1,CD2
  45 FORMAT(*MU1 = *,E20.4,5X,*MU2 = *,E20.4,/,
    1        *MU3 = *,E20.4,5X,*  M = *,E20.4,/,
    1        *  K = *,E20.4)
```

```
407     47  FORMAT(*DISLOCATION POSITION = -*,F10.3,*, -*,F10.3,* KMS.*,/,
408        1        *WIDTH OF SLIP ZONE    =  *,F10.3,/,
409        1        *SLAB THICKNESS        =  *,F10.3,* KMS.*,/,
410        1        *BURGERS VECTOR        =  *,F10.3,* MMS.*)
411     49  FORMAT(A2,* STRESS/STRAIN VS. *,A2)
412         WRITE(7,64)
413     64  FORMAT(*0=RE-START*,/,*1=STOP*,/,
414        1        *2=RE-START WITH SAME VALUES, DIFFERENT PROFILE*)
415         CALL GETNUM(AB) $LJK=AB(1)
416         IF(LJK.EQ.0)GOTO48 $IF(LJK.EQ.2)GO TO 70
417         STOP $END
418         SUBROUTINE FUNC1(P,A,NU,D,Q,ALAMBD,F)
419         R=(((P+A+(2.*NU*D))**2)+((Q+ALAMBD)**2))
420         IF(R.EQ.0.)R=1.E-10$F=(P+A+(2.*NU*D))/R
421         RETURN $END
422         SUBROUTINE FUNC2(P,Q,R,S,A,NU,D,ALAMBD,F)
423         T=(((R+A+(2.*NU*D))**2)+((S+ALAMBD)**2))
424         IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
425         RETURN $END
426         SUBROUTINE FUNC3(P,Q,R,S,F)
427         T=(((R+S)**2)+((P+Q)**2))
428         IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
429         RETURN $END
430         SUBROUTINE GETNUM(R)
431         DIMENSION R(1),L(80)
432         READ(7,9)L $I=J=0
433       6 J=J+1 $N=F=S=0 $M=F=1
434       5 I=I+1 $IF(I.GT.80)RETURN $D=L(I) $K=4
435         IF(D.EQ.38)K=2 $IF(D.GE.27.A.D.LE.36)K=1
436         IF(D.EQ.47)K=3 $K=K+S $GOTO(1,2,3,5,1,4,3,4)K
437       1 N=N*10+D-27  $S=4  $GOTO5
438       2 M=-1 $S=4 $GOTO5
439       3 P=I $S=4 $GOTO5
440       4 IF(P.NE.0)F=10.**(I-P-1) $R(J)=N/F*M $GOTO6
441       9 FORMAT(80A1)
442         END
```