

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

This report is preliminary and has not been edited or reviewed for conformity with Geological Survey standards and nomenclature.

Note: Some of the pages in this report may be difficult to read. This is the best copy that can be made from the original.

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

A)XPND, XPND01, XPND02

1)Documentation	A 1
a)Appendix A--Array Assignements	A11
b)Appendix B--Trigger Option	A16
c)Appendix C--Definitions of Some Parameters	A19
2)Examples of Program Operation	
a)XPND	A20
b)XPND01	A30
c)XPND02	A43
3)Program Listing	
a)XPND	A54
b)XPND01	A62
c)XPND02	A71

B)MAXVGT

1)Documentation	B 1
2)Examples of Program Operation	B 7
3)Program Listing	B28

C)ST3PHS, ST3PSZ, STNRML, SHRSTN, DBLNRM

1)Documentation	C 1
a)Description of Variables and Flags Occurring in Output	C10
2)Examples of Program Operation	
a)ST3PHS	C12
b)ST3PSZ	C20
c)STNRML	C27
d)SHRSTN	C34
e)DBLNRM	C40
3)Program Listing	
a)ST3PHS	C51
b)ST3PSZ	C58
c)STNRML	C65
d)SHRSTN	C73
e)DBLNRM	C81

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; program 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 x_1 axis is coincident with the strike of the fault, the x_2 axis is the horizontal axis perpendicular to x_1 , and the x_3 axis is vertical with its positive direction into the medium. In these programs x_1 increases to the northwest and x_2 increases to the northeast. The angle between x_1 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 $x_1 > 0$ 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 (XPND01) 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
- 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

XPNDØ1 {

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:

0 = 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

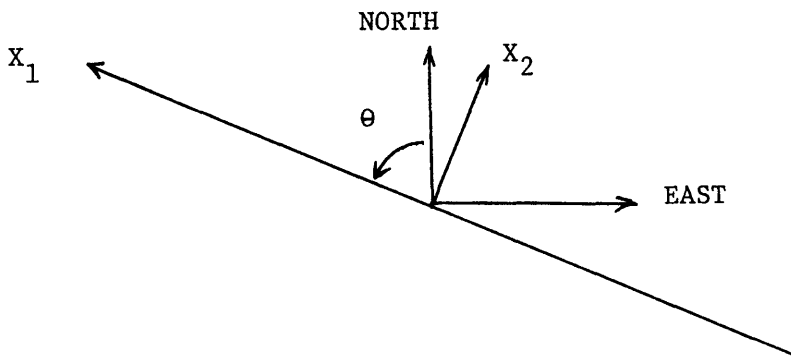
This program operates in the same fashion as, and requires responses identical to, XPNDØ1. However instead of the TRIGGER option responding 'D(M1) > C(M2)' it will ask for M1 and M2 such that 'D(M1) < C(M2)'. This modification allows the triggered zone to have coordinates that are less than the coordinates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



$\theta = \text{theta} - \text{assumed to be } 45.^{\circ}$

Figure 1

Geometry assumed in Programs XPND/XPND01

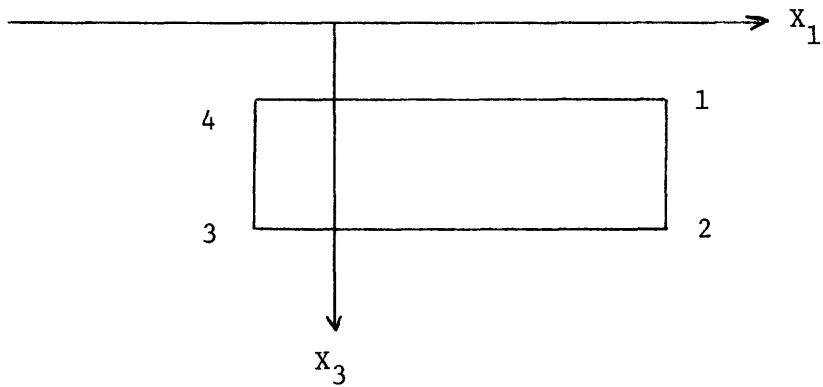


Figure 2

Geometry and notation used in Programs XPND/XPND01

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

APPENDIX A (cont'd)

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

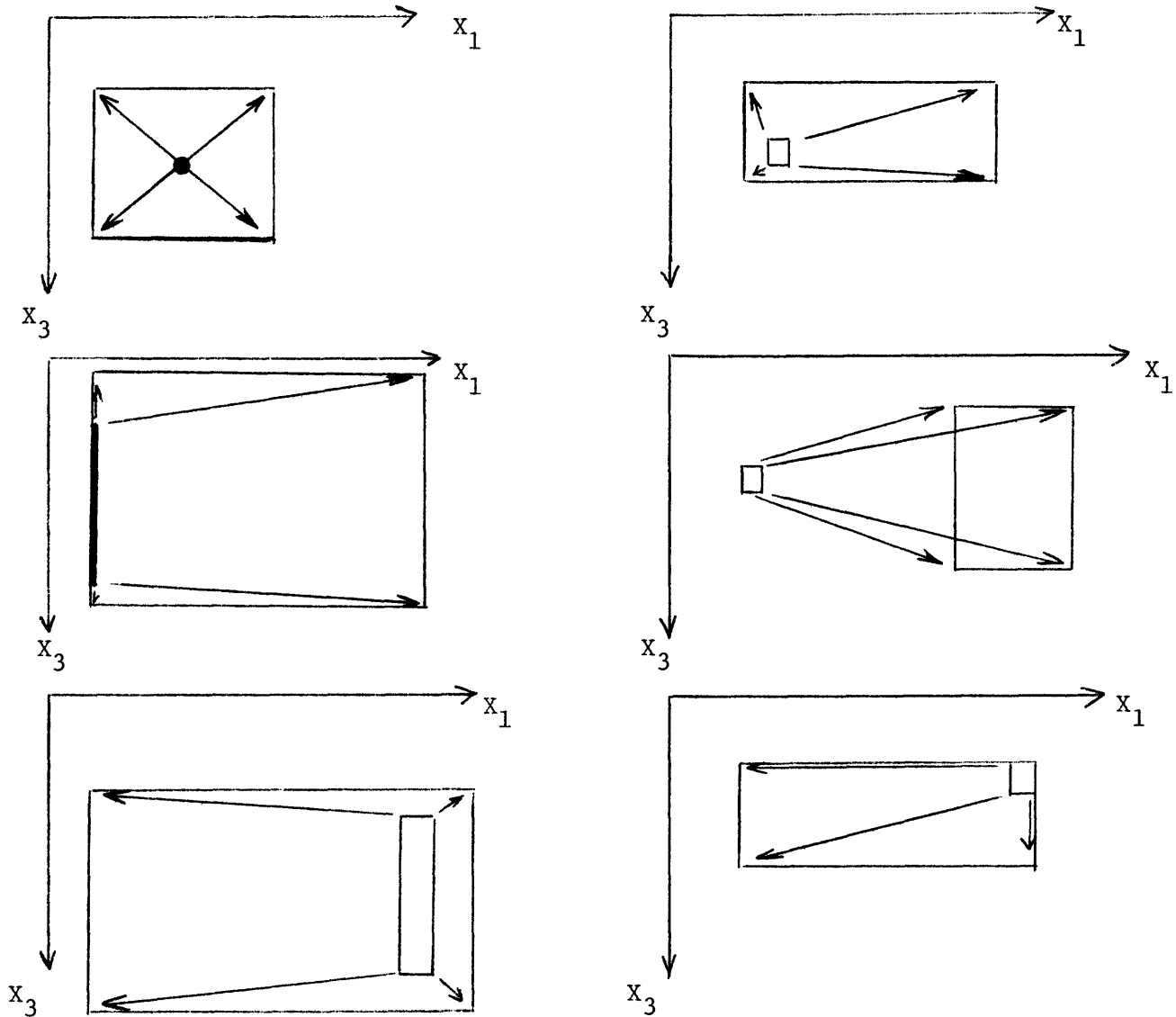
D(17) = U1

D(18) = U3

The coordinates are incremented linearly:

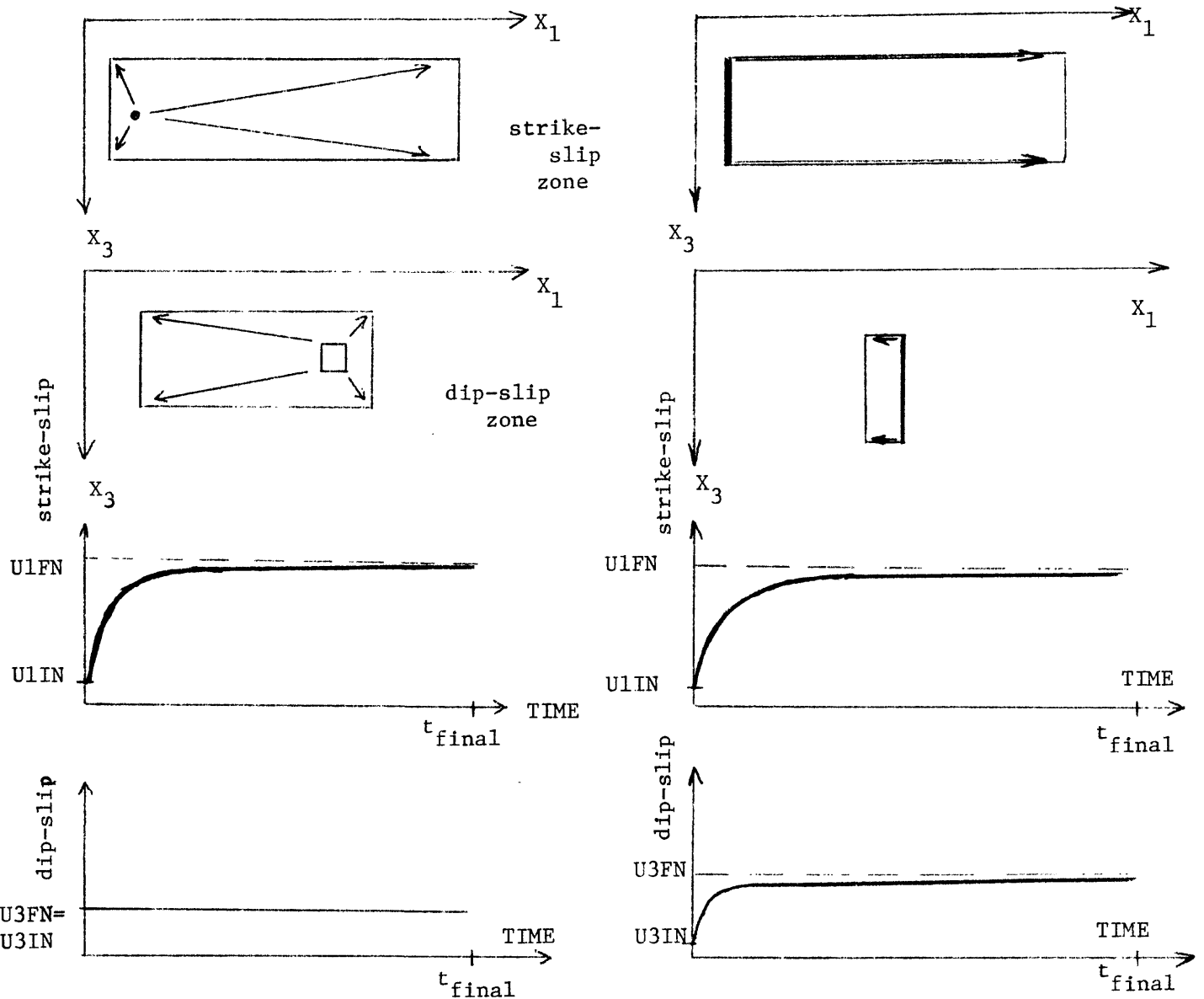
ie. $(\text{initial}) + [(\text{final}-\text{initial}) * (\text{time})]$

Examples of slip zone expansion:



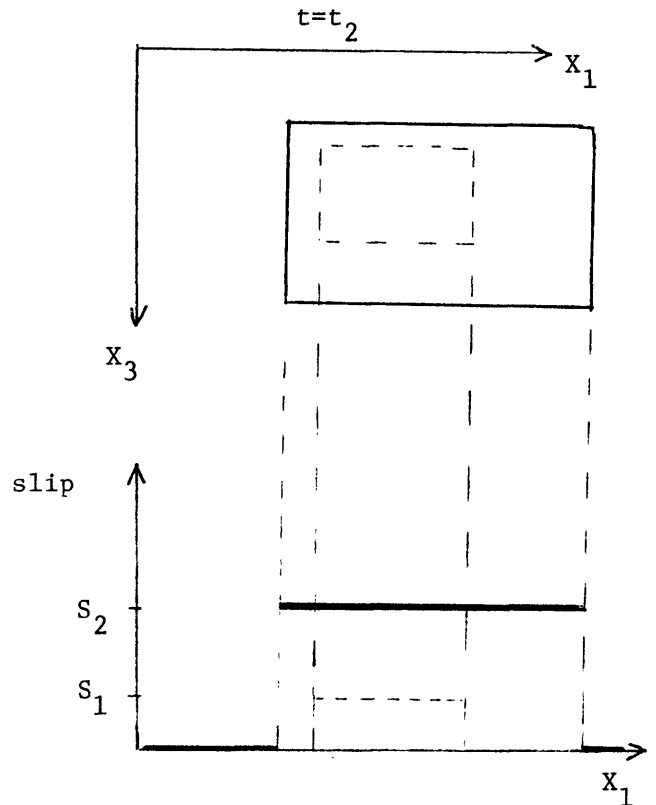
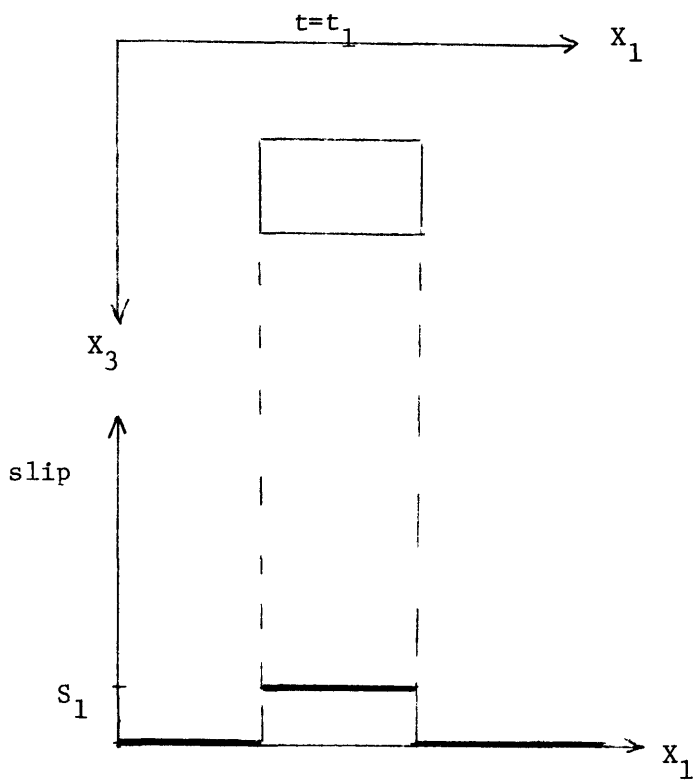
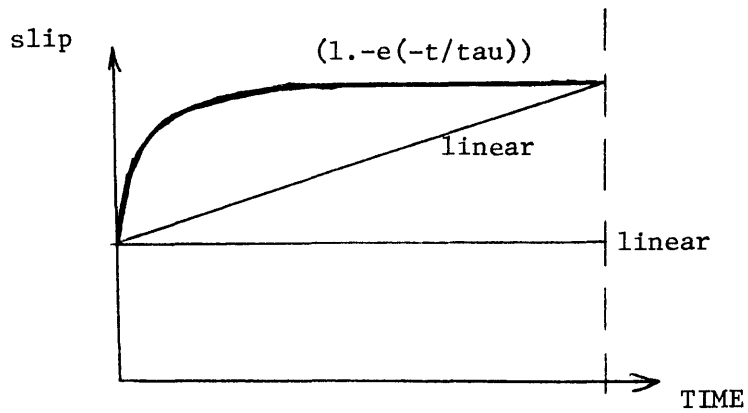
If the contraction option is entered the zone's growth will be the reverse of that shown above.

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:

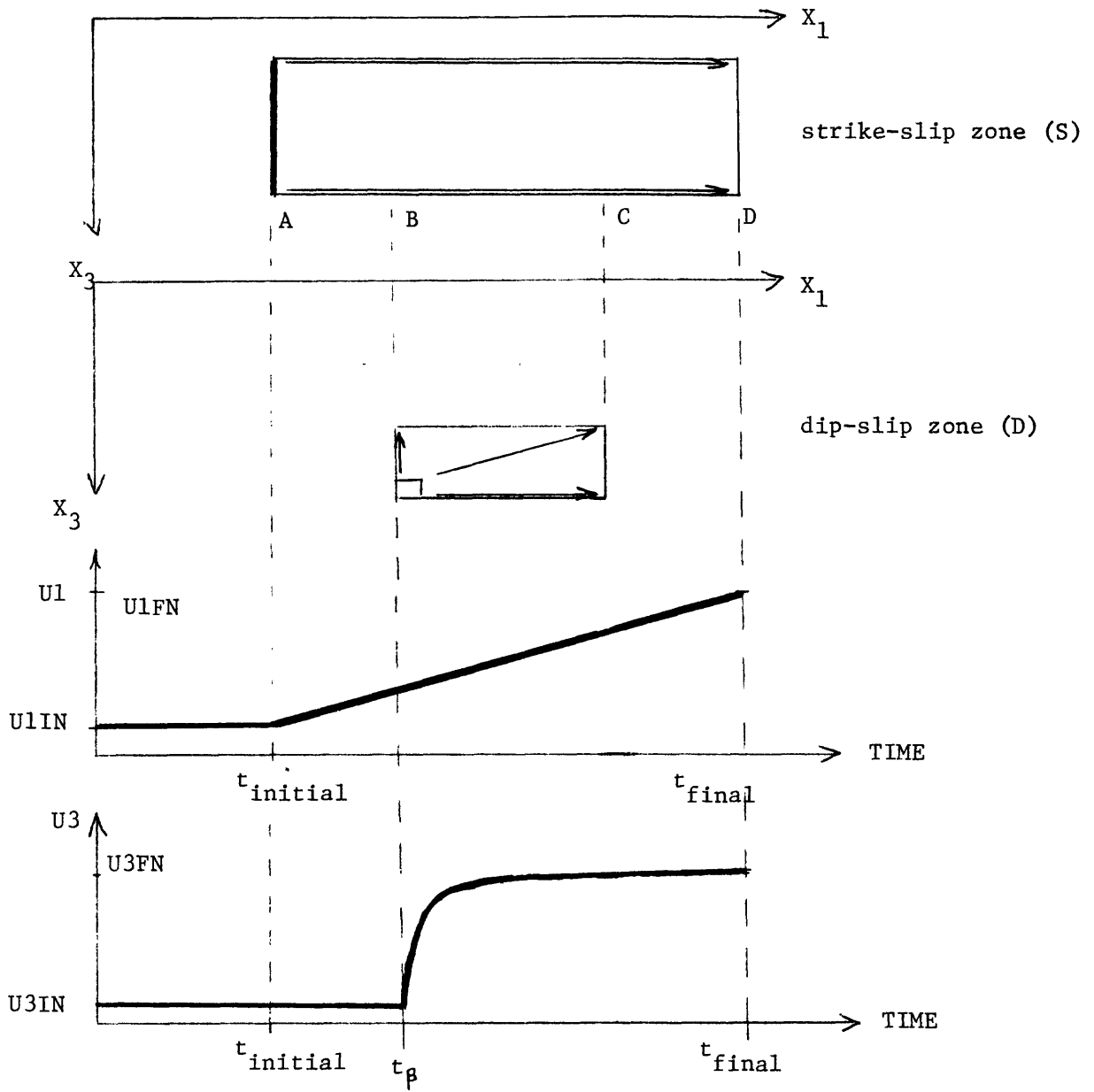


APPENDIX B

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:



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.

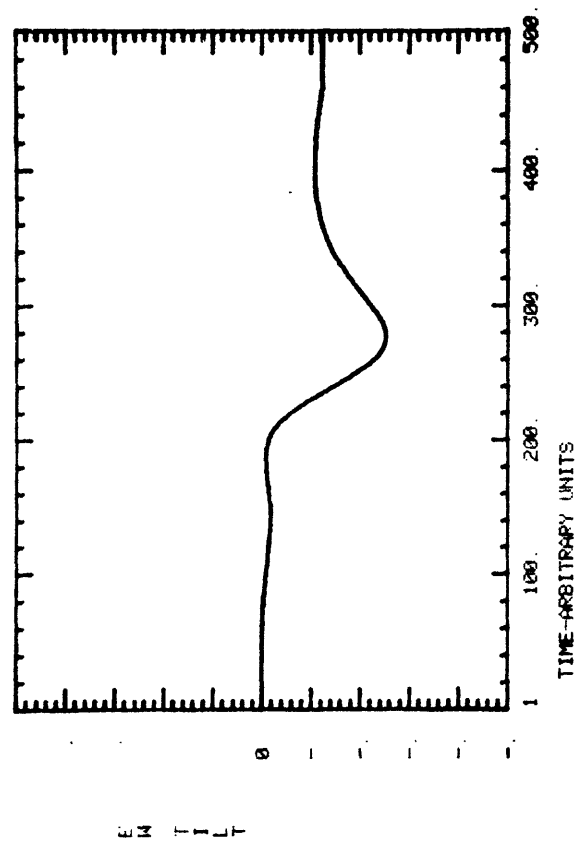
EXAMPLE FROM XPND


```

1 0=PE-START, 1=NEW PLOT
2 1
3 SET HORIZONTAL SCALE? Y OR NK=BLANK )
4 1
5 SET VERTICAL SCALE? Y OR NK=BLANK )
6 Y
7 MIN/MAX Y VALUES
8 .5 .5
9 SKIP PLOT OF NS TILT

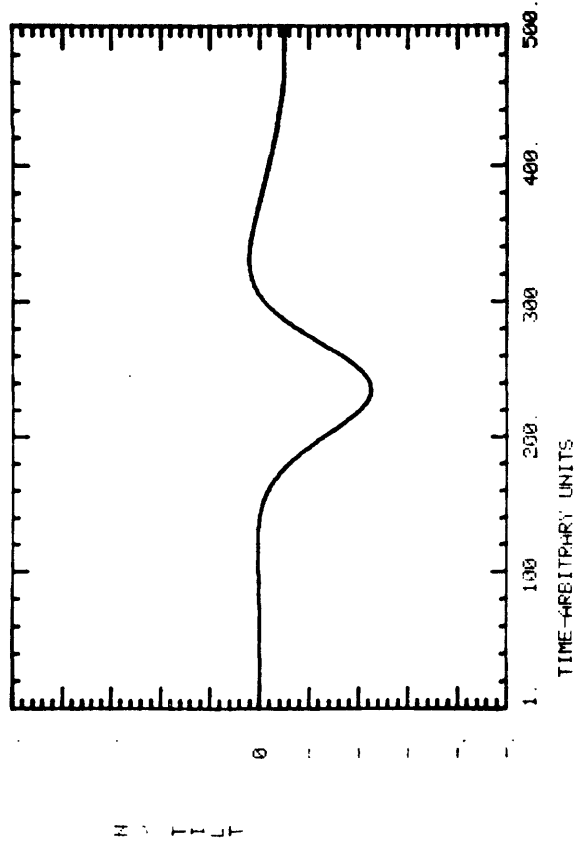
```

EXPANDING SLIP ZONES

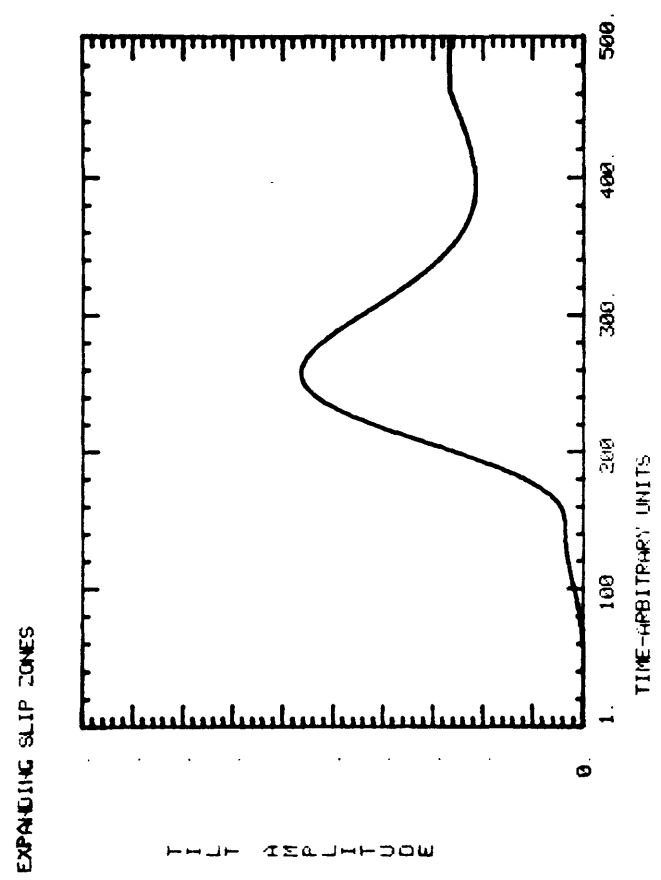


EXPANDING SLIP ZONES

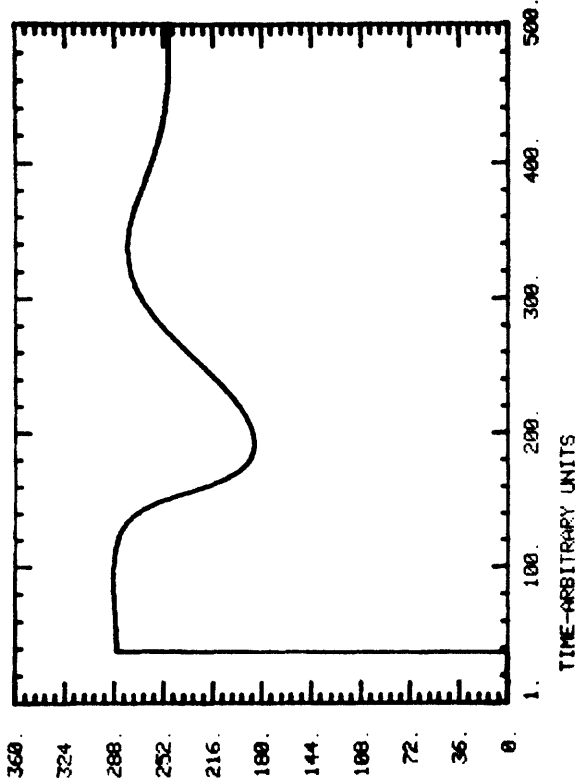
- 1. 0=RE-START, 1=NEW PLOT
- 2. SET HORIZONTAL SCALE? Y OR N(=BLANK)
- SET VERTICAL SCALE? Y OR N(=BLANK)
- Y:
- 0
- 5
- SKIP PLOT OF TILT AMPLITUDE?



0=RE-START. THEN PLOT
 2, SET HORIZONTAL SCALE? (UP H=BLANK)
 SET VERTICAL SCALE? (IF H=BLANK)
 Y, MIN-MAX Y VALUES
 U 2681
 4, IF PLOT OF TILT AZIMUTH



EXPANDING SLIP ZONES



T I L T A Z I M U T H

90	0	1.00	0	1.00	1.00	500	462	9
90	0	1.00	0	1.00	0	463	-70.667	229.000
1 10	0	1.00	0	1.00	1.10			
90	0	0	0	0	0			
0	0	0	0	0	0			
0	0	0	0	0	0			
2 00	0	0	0	0	2.00			
0	0	1.00	0	1.00	0			
0	0	-3.00	0	0	0			
1	2	3	38	462				
1	18	332	37	463				

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 TILT-METER COORDINATES ONLY
 7=STOP
 ?
 OK - ^EDIT

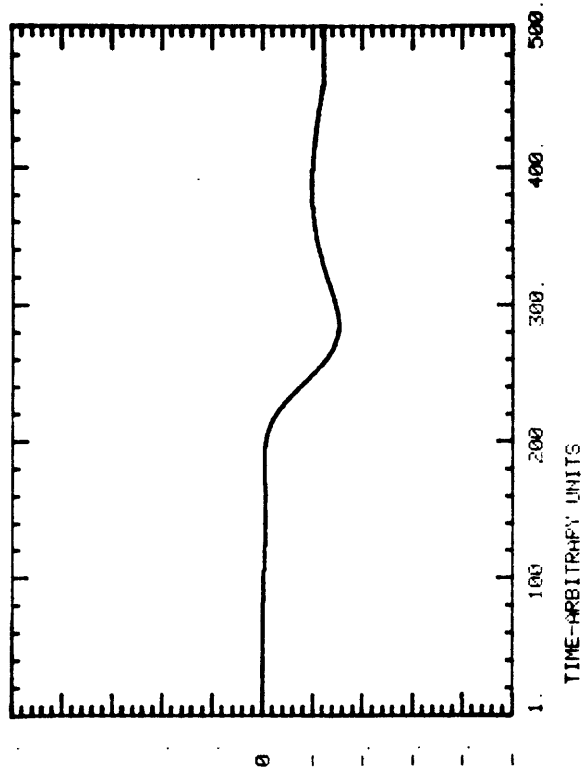
EXAMPLE FROM XPND01

A29a

COORDINATES OF STATION (X1,X2)
 371
 1 THE IN-SLIP COMPONENT INCREMENTED EXPONENTIALLY
 AND NORTH= 45 N30
 SPECIFY 2 CORNERS OF DISLOCATION SURFACE FOR DISPLAY
 1 D1X1=D2X1 2=D1X3=D4X3
 4=D2X3=D3X3 5=D3X1=D4X1
 9=D1X1=D2X1 10=D1X3=D4X3
 1=D2X3=D3X3 13=D3X1=D4X1
 1 101
 MIN/MAX VALUES OF EM 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 0 288.503
 NOTE: TILT AMPLITUDES ARE IN MICROGRADUANS
 AZIMUTH IN DEGREES)
 THE FOLLOWING ARE PLOTS OF THE EM AND NS
 COMPONENTS OF TILT, AND THE TILT AMPLITUDE
 AND AZIMUTH (MEASURED CLOCKWISE FROM NORTH)
 (ARE-START, 1=CONTINUE)
 11
 WRITE PLOT TITLE, 80 CHARACTERS
 XPND01/EXPANDING SLIP ZONES'
 SET HORIZONTAL SCALE? Y OR N(=BLANK)
 1
 SET VERTICAL SCALE? Y OR N(=BLANK)
 Y1
 MIN/MAX Y VALUES
 N=5 51
 -5 51
 SKIP PLOT OF EM TILT?

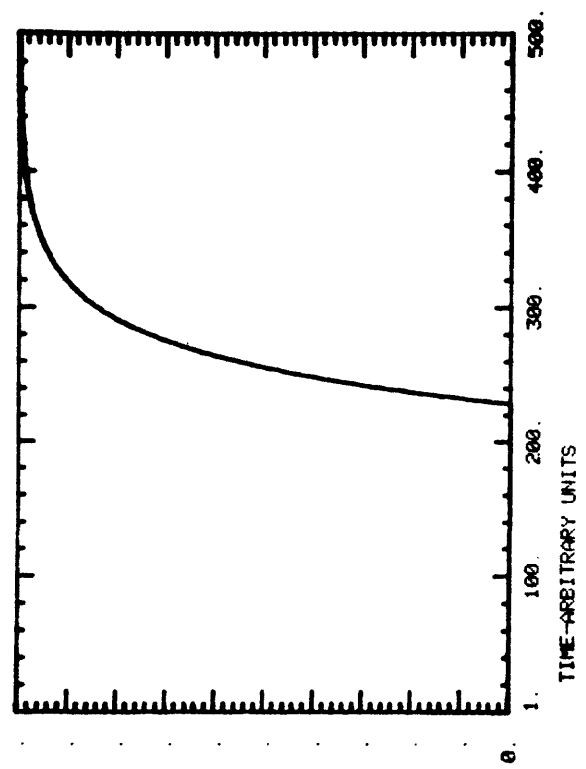
1-ZONE METHODS
 2-ZONE METHODS
 1 STRIKE-SLIP COMPONENT INCREMENTED EXPONENTIALLY
 2 DIP-SLIP COMPONENT INCREMENTED EXPONENTIALLY
 3 BOTH SLIP COMPONENTS INCREMENTED EXPONENTIALLY
 4 NEITHER SLIP COMPONENT INCREMENTED EXPONENTIALLY
 2!
 DO YOU WISH THE 'TRIGGER' OPTION?
 YES OR 'NO'
 YES
 SPECIFY 'TRIGGER' PARAMETERS IN THE FORM
 (DIP) (DIP) (DIP) (DIP)
 1 1
 SPECIFY WHICH SLIP COMPONENT IS TO BE =0.
 (DIP) (DIP) (DIP) (DIP)
 1 1 1 1
 STRIKE-SLIP=0, DIP-SLIP=0
 11
 SPECIFY INITIAL AND FINAL VALUES.
 RIGHT-LATERAL STRIKE-SLIP IS 0
 DIP-SLIP >0 FOR 'DIP' SIZE. 00001
 FORMAT=DIP:USER:USER
 1 3 0 51
 SPECIFY INITIAL AND FINAL COORDINATES
 OF DIP-SLIP ZONE CORNERS AS INDICATED
 D1X1=D1X3=D2X1=D2X3=D3X1=D3X3=D4X1=D4X3=D1X1FN
 D2X1FN D3X1FN D1X1FN D2X3FN D3X3FN D4X1FN
 1 0 1 9 1 1 0 1 51
 SPECIFY INITIAL AND FINAL VALUES
 OF STRIKE-SLIP ZONE COORDINATES AS INDICATED
 C1X1=D1X3=D2X3=D3X1=D3X3=D4X1FN C1X3FN C2X3FN C3X3FN
 3 3 0 0 2 0 1 0

*(100) EXPANDING SLIP ZONES

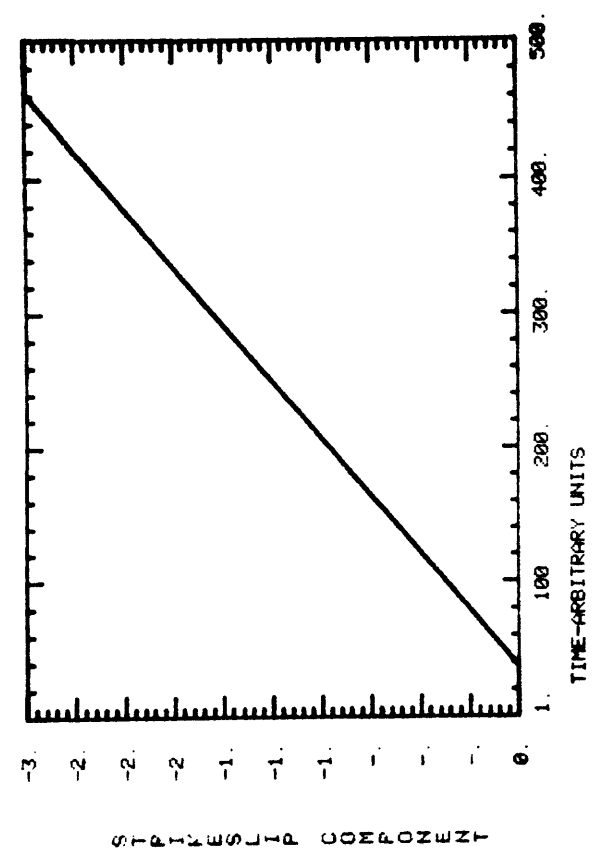


TYPE-START, 1=NEW PLOT
2=1
SET HORIZONTAL SCALE? Y OR N(=BLANK)
1
SET VERTICAL SCALE? Y OR N(=BLANK)
1
MIN MAX Y VALUES
: 5 51
SKIP PLOT OF NS TILT

31091/EXPANDING SLIP ZONES

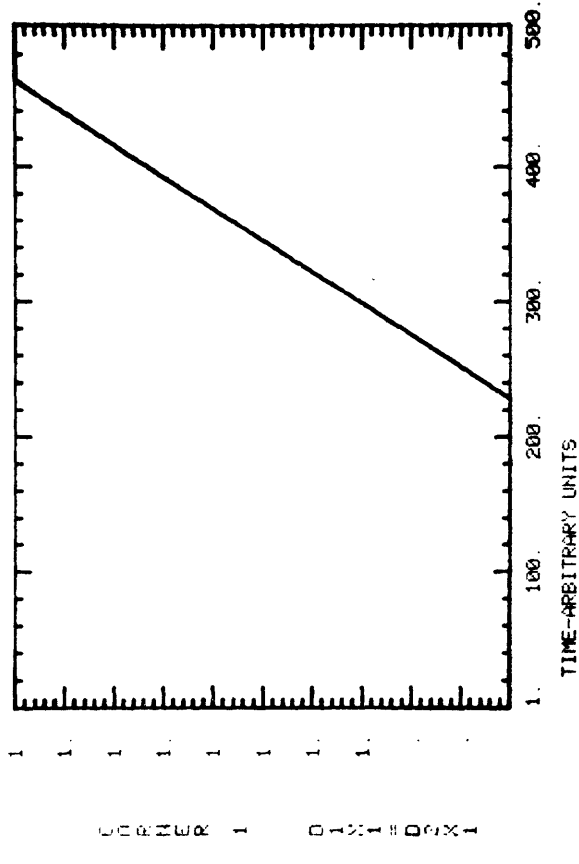


31091/EXPANDING SLIP ZONES

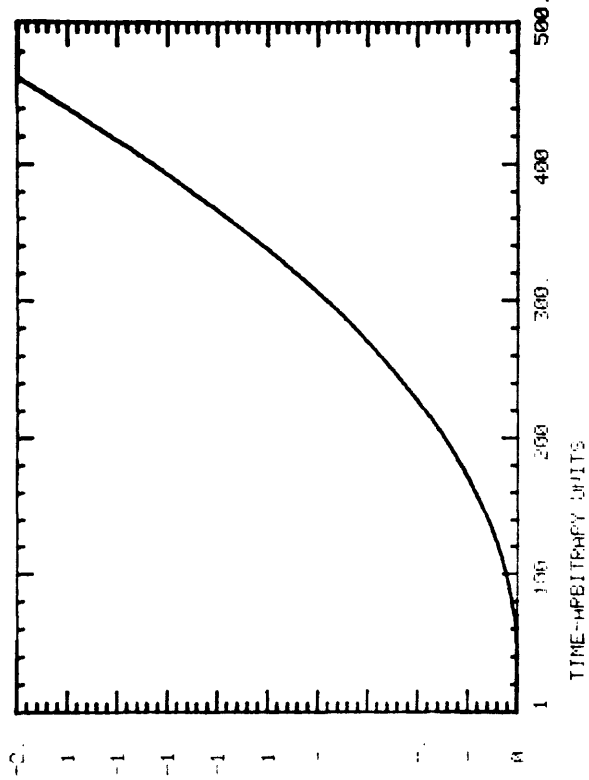


0=START
2=SKIP PLOT OF CORNER 17

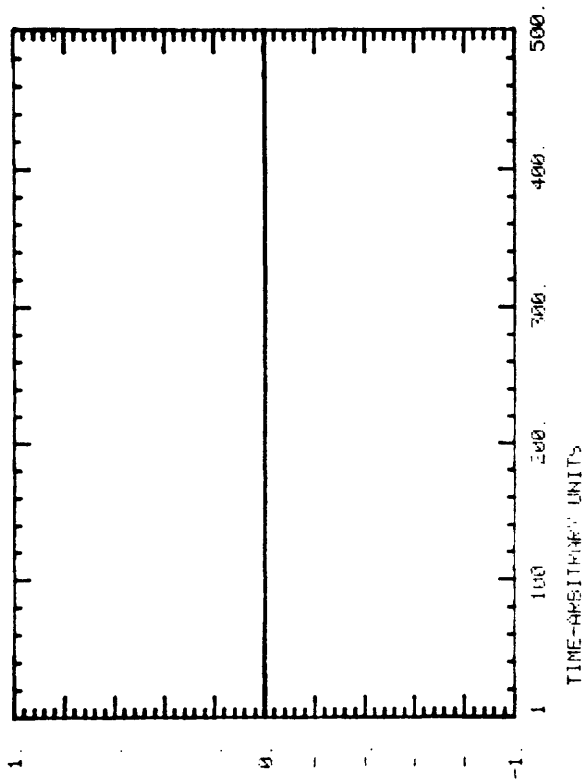
XPLOU: EXPANDING SLIP ZONES



SPR001 EXPANDING SLIP ZONES



ZC1001 EXPANDING SLIP ZONES



A4I

A4D

.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

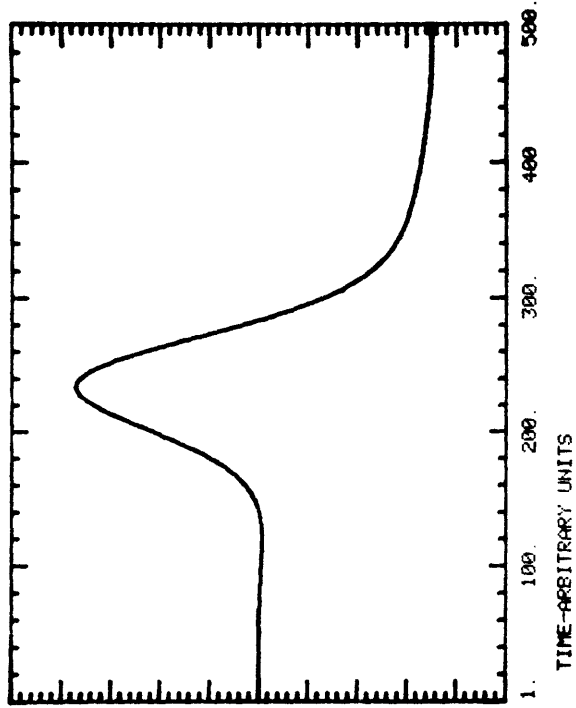
EXAMPLE FROM XPND02

A42a

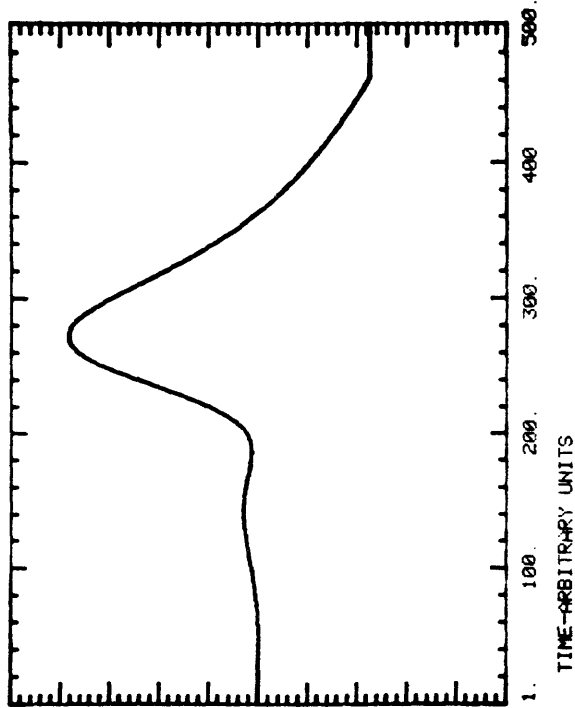
XP:002 EXPANDING SLIP ZONES

- 1 0=RE-START, 1=NEW PLOT
- 2 1 SET HORIZONTAL SCALE? Y OR N (=BLANK)
- 3 1 SET VERTICAL SCALE? Y OR N (=BLANK)
- 4 1 MIN/MAX Y VALUES
- 5 31 SKIP PLOT OF NS TILT

E
W
T
I
L
T



XPID02/EXPANDING SLIP ZONES



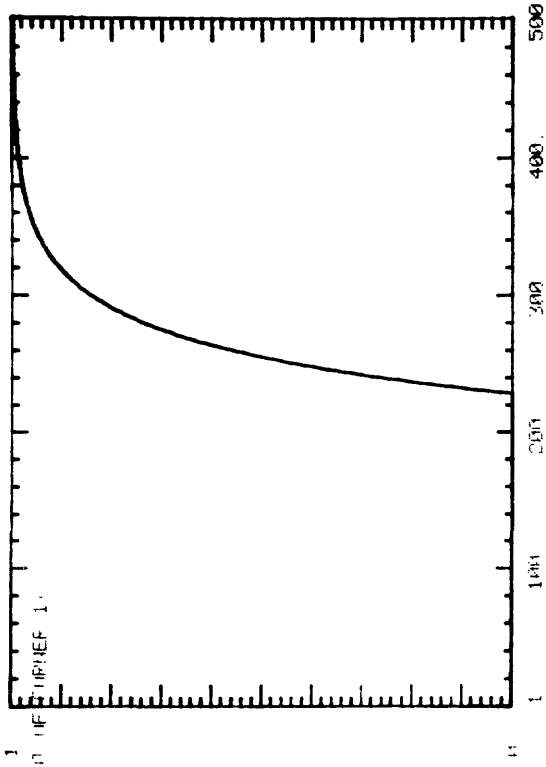
TILT

- 1 Q=RG--START, 1=NEW PLOT
- 2 SET HORIZONTAL SCALE? Y OR N(=BLANK)
- 3 SET VERTICAL SCALE? Y OR N(=BLANK)
- 4 SKIP PLOT OF TILT AMPLITUDE?
- 5 N=RG--START, 1=NEW PLOT
- 6 SET HORIZONTAL SCALE? Y OR N(=BLANK)
- 7 SET VERTICAL SCALE? Y OR N(=BLANK)
- 8 SKIP PLOT OF TILT AZIMUTH?
- 9 A=PE--START, 1=NEW PLOT
- 0 SKIP PLOT OF STRIKE SLIP COMPONENT?

XPN001: EMERGING SLIP ZONES

0 = 0.001 FT

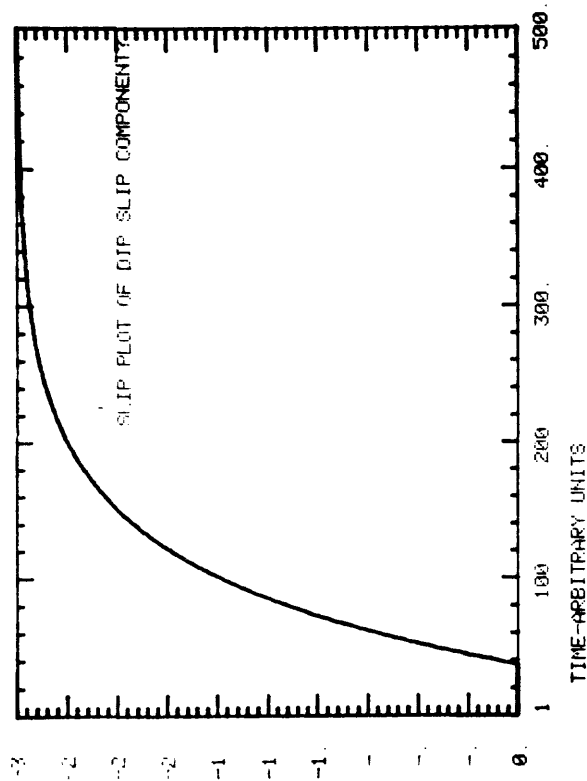
2 SKIN FOLD UP SURF 1.



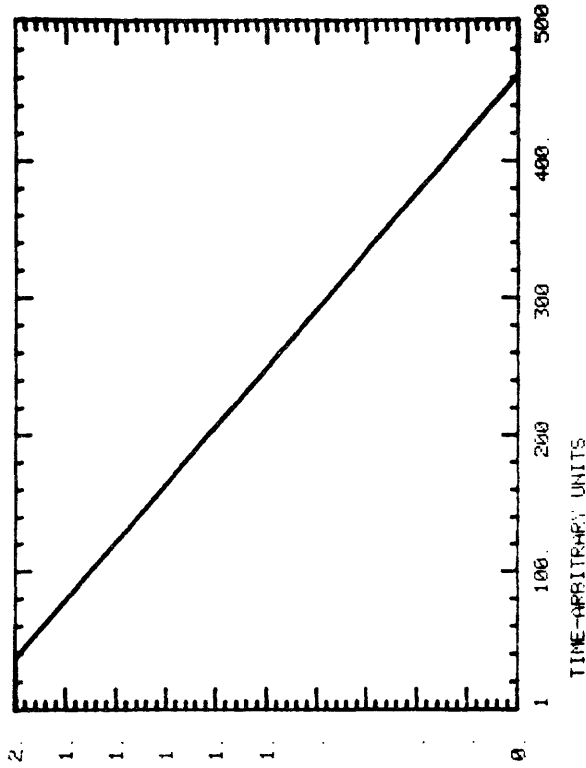
XPI002: EMERGING SLIP ZONES

0 = 0.001 FT

2 DTP RESULT COMPONENT

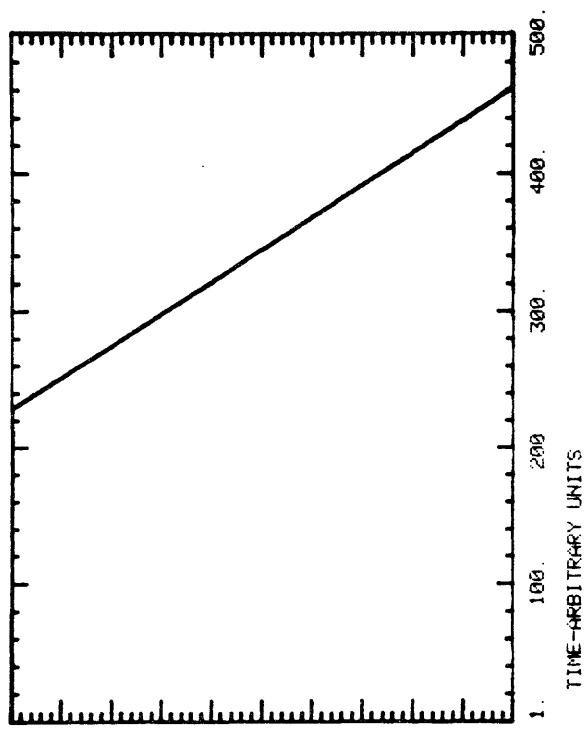


XPND02/EXPANDING SLIP ZONES



U3X1=04X1

XPND02/EXPANDING SLIP ZONES
SKIP PLOT OF CORNER 2?



U3X1=04X1

I
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

```

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=TAFETTY, TAPE7=TAPETTY)
13     COMMON/TVPCOL/TVPOL(8)
14     COMMON/TVTUNE/ITUNF(30)
15     COMMON/JPLCT/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 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., -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 FFT(5LTAPE7, IFET, -8)
25     20 CONTINUE
26     MSTART=9 $WRITE(7,12)
27     12 FORMAT(*1=ZONE EXPANS*,/,
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=(.075*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) $GOTO 11
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)

```

```

117     IF (TRIGGR.EQ.3HYFS.AND.D(M1).GE.C(M2).AND.M3.EQ.17.AND.LFLAG.EQ.4)
118     1 D(M3)=(((C(34)-C(33))/(KTIME))*(I-RTIME))+C(33)
119
120 156 CALL CMPTILT(D,X1,X2,X3,T1,T2)
121     IF(D(1).NE.D(3).OR.D(5).NE.D(7).OR.D(2).NE.D(8)
122     1 .OR.D(4).NE.D(6).OR.D(9).NE.D(11).OR.D(13).NE.D(15)
123     1 .OR.D(10).NE.D(16).OR.D(12).NE.D(14))WRITE(7,67)
124 67 FORMAT(*CORNERS NOT INCREMENTING PROPERLY*)
125     T(I)=I $TEW(I)=T2
126 103 TNS(I)=T1
127     K7=IJKL-1
128     DO 106 I=1,K7
129     TEW(I)=TEW(IJKL) $T(I)=I
130 106 TNS(I)=TNS(IJKL)
131     K8=NEND+1
132     DO 107 I=K8,NUM
133     T(I)=I $TEW(I)=TEW(NEND)
134 107 TNS(I)=TNS(NEND)
135     IF(IFLAG.LC. 2)GO TO 108
136     A=TEW(1) $B=TNS(1)
137     DO 109 I=1,NUM
138     TEW(I)=TEW(I)-A
139 109 TNS(I)=TNS(I)-B
140     GO TO 110
141 108 DO 111 I=1,NUM
142     TEW(I)=TEW(NUM)-TEW(I)
143 111 TNS(I)=TNS(NUM)-TNS(I)
144 110 CT=COS(THETA*.01745)
145     ST=SIN(THETA*.01745)
146     DO 21 I=1,NUM
147     B=TEW(I) $A=TNS(I)
148     TNS(I)=A*CT+B*ST
149     TEW(I)=-A*ST+B*CT
150     TAMP(I)=SQRT((TEW(I)**2)+(TNS(I)**2))
151     IF(TNS(I).EQ.0.)TNS(I)=1.E-20
152     TAZM(I)=(ATAN(TEW(I)/TNS(I)))*(180./3.1415926)
153     IF(TNS(I).LT.0.)TAZM(I)=TAZM(I)+180.
154     IF(TAZM(I).LT.0.)TAZM(I)=TAZM(I)+360.
155 21 IF(TAZM(I).GT.360.)TAZM(I)=TAZM(I)-360.
156     TEWMID=TEW(1) $TNSMID=TNS(1) $TAMPID=TAMP(1) $TAZMID=TAZM(1)
157     TEWMIN=TEWMAX=TEWMID $TNSMIN=TNSMAX=TNSMID
158     TAMPIN=TAMPX=TAMPID $TAZMIN=TAZMX=TAZMID
159     DO 160 I=1,NUM
160     IF (TEW(I) .LT. TEWMIN)TEWMIN=TEW(I)
161     IF (TEW(I) .GT. TEWMAX)TEWMAX=TEW(I)
162     IF (TNS(I) .LT. TNSMIN)TNSMIN=TNS(I)
163     IF (TNS(I) .GT. TNSMAX)TNSMAX=TNS(I)
164     IF (TAMP(I) .LT. TAMPIN)TAMPIN=TAMP(I)
165     IF (TAMP(I) .GT. TAMPX)TAMPX=TAMP(I)
166     IF (TAZM(I) .LT. TAZMIN)TAZMIN=TAZM(I)
167     IF (TAZM(I) .GT. TAZMX)TAZMX=TAZM(I)
168 160 CONTINUE
169     WRITE(7,17)TEWMIN,TEWMAX,TNSMIN,TNSMAX,TAMPIN,TAMPX,TAZMIN,TAZMX
170 17 FORMAT(*MIN./MAX. VALUES OF EW COMPONENT*,2X,E10.3,2X,E10.3,/,
171 1 *MIN./MAX. VALUES OF NS COMPONENT*,2X,E10.3,2X,E10.3,/,
172 1 *MIN./MAX. VALUES OF AMPLITUDE*,5X,E10.3,2X,E10.3,/,
173 1 *MIN./MAX. VALUES OF AZIMUTH*,3X,F10.3,2X,F10.3,/,
174 1 *(NOTE TILT AMPLITUDES ARE IN MICRORADIANS*,/,
175 1 *AZIMUTH IN DEGREES)*)

```

```

175      WRITE(7,170)
176 170  FORMAT(*THE FOLLOWING ARE PLOTS OF THE EW AND NS*,/,
177      1  *COMPONENTS OF TILT, AND THE TILT AMPLITUDE*,/,
178      1  *AND AZIMUTH (MEASURED CLOCKWISE FROM NORTH)*,/,
179      1  *0=RE-START, 1=CONTINUE*)
180      CALL GETNUM(A) $IKFLAG=A(1)
181      IF(IKFLAG.EQ. 0) GO TO 113
182      LU=7 $LNLGX=1 $LNLY=1 $NCLX=2 $NCLY=2
183      WRITE(7,199)
184 199  FORMAT(*WRITE PLOT TITLE, 80 CHARACTERS*)
185      READ(7,200)(LTITL(JM),JM=1,8)
186 200  FORMAT(8A10)
187      KX(1)=10HTIME-ARBIT
188      KX(2)=10HRARY UNITS
189      KY(1)=10H EW TILT
190      KY(2)=10H          $MAJX=5 $MAJY=10 $LTITL2(1)=1
191 903  XLT=T(1)$XRT=T(NUM) $YLO=TEWMIN $YUP=TEWMAX
192      WRITE(7,70)
193      70  FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
194      READ(7,66)CHARAC $IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GO TO 71
195      WRITE(7,72)
196      72  FORMAT(*MIN/MAX X VALUES*)
197      CALL GETNUM(A) $XLT=A(1) $XRT=A(2)
198      71  WRITE(7,73)
199      73  FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
200      READ(7,66)CHAPAC $IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GO TO 74
201      WRITE(7,75)
202      75  FORMAT(*MIN/MAX Y VALUES*)
203      CALL GETNUM(A) $YLO=A(1) $YUP=A(2)
204      74  IF(TEWMIN.EQ.TEWMAX)YLO=YUP-1.
205          IF(TEWMIN.EQ.TEWMAX)YUP=YUP+1.
206          WRITE(7,1)
207      1  FORMAT(*SKIP PLOT OF EW TILT?*)
208      READ(7,66)IJVAR
209 66  FORMAT(A1)
210      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TEW,T,1,NUM)
211      WRITE(7,19) $CALL GETNUM(A)$IKFLAG=A(1)
212      IF(IKFLAG.EQ.0)GOTO113$IF(IKFLAG.EQ.1)GOTO903
213 900  KY(1)=10H NS TILT      $XLT=T(1)$XRT=T(NUM)
214      KY(2)=10H
215      YLO=TNSMIN $YUP=TNSMAX $WRITE(7,70) $READ(7,66)CHARAC
216      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO710
217      WRITE(7,72)$CALL GETNUM(A)$XLT=A(1)$XRT=A(2)
218 710  WRITE(7,73)$READ(7,66)CHAPAC
219      IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO740
220      WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
221 740  IF(YLO.EQ.YUP)AAA=YLO$IF(YLO.EQ.AAA)YUP=YUP+1.
222      IF(YLO.EQ.AAA)YLO=YLO-1.
223      WRITE(7,910) $READ(7,66)IJVAR
224 910  FORMAT(*SKIP PLOT OF NS TILT?*)
225      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TNS,T,1,NUM)
226      WRITE(7,19)
227 19  FORMAT(*0=RE-START, 1=NEW FLOT*)
228      CALL GETNUM(A) $IKFLAG=A(1)
229      IF(IKFLAG.EQ.0)GO TO 113 $IF(IKFLAG.EQ.1)GOTO900
230 911  KY(1)=10HTILT AMPLI $XLT=T(1) $XRT=T(NUM)
231      KY(2)=10HTUDE
232      YLO=TAMPN $YUP=TAMPX $WRITE(7,70)$READ(7,66)CHARAC

```

```

233     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO741
234     WRITE(7,72)$CALL GETNUM(A) $XLT=A(1)$XRT=A(2)
235     741  WRITE(7,73)$READ(7,66)CHARAC
236     IF(CHAPAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO742
237     WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
238     742  IF(YLO.EQ.YUP)AAA=YLO$IF(YLO.EQ.AAA)YUP=YUP+1.
239     IF(YLO.EQ.AAA)YLO=YLO-1.
240     WRITE(7,912) $READ(7,66)IJVAR
241     912  FORMAT(*SKIP PLOT OF TILT AMPLITUDE?*)
242     IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAMP,T,1,NUM)
243     WRITE(7,19)$CALL GETNUM(A) $IKFLAG=A(1)
244     IF(IKFLAG.EQ.0)GOTO113$IF(IKFLAG.EQ.1)GOTO911
245     913  KY(1)=10HTILT AZIMU $XLT=T(1)$XRT=T(NUM)
246     KY(2)=10HTH $YLO=TAZMMN $YUP=TAZMMX
247     WRITE(7,70) $READ(7,66)CHARAC
248     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO743
249     WRITE(7,72)$CALL GETNUM(A)$XLT=A(1)$XRT=A(2)
250     743  WRITE(7,73)$READ(7,66)CHARAC
251     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO744
252     WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
253     744  IF(YLO.EQ.YUP)AAA=YLO$IF(YLO.EQ.AAA)YUP=YUP+1.
254     IF(YLO.EQ.AAA)YLO=YLO-1.
255     WRITE(7,914) $READ(7,66)IJVAR
256     914  FORMAT(*SKIP PLOT OF TILT AZIMUTH?*)
257     IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAZM,T,1,NUM)
258     WRITE(7,68)(C(I),I=1,36)
259     68  FORMAT(9(4F12.2,/))
260     WRITE(7,69)IFLAG,IKFLAG,LFLAG,IJKL,NUM,NEND,M1,M2,M3,KTIME,K7,K8,
261     1  TAU,RTIME
262     69  FORMAT(7I10,/,5I10,2F10.3)
263     113  WRITE(7,115)
264     115  FORMAT(*1=RE-START WITH ALL NEW VALUES*,/,
265     1  *2=RE-START WITH NEW STRIKE-SLIP VALUE AND ZONE COORDINATES*,/,
266     1  *3=RE-START WITH NEW DIP-SLIP VALUE AND ZONE COORDINATES*,/,
267     1  *4=RE-START WITH NEW STRIKE-SLIP VALUE ONLY*,/,
268     1  *5=RE-START WITH NEW DIP-SLIP VALUE ONLY*,/,
269     1  *6=RE-START WITH NEW TILT METER COORDINATES ONLY*,/,
270     1  *7=STOP*)
271     CALL GETNUM(A) $MSTART=A(1)
272     IF(MSTART.EQ.2)GOTO116 $IF(MSTART.EQ.3)GOTO117
273     IF(MSTART.EQ.4)GOTO118 $IF(MSTART.EQ.5)GOTO119
274     IF(MSTART.EQ.6)GOTO120 $IF(MSTART.EQ.7)GOTO630
275     GO TO 20
276     116  WRITE(7,12)
277     CALL GETNUM(A) $IFLAG=A(1) $WRITE(7,15)
278     CALL GETNUM(A) $LFLAG=A(1)
279     WRITE(7,23)
280     23  FORMAT(*U1IN,U1FN*)
281     CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
282     CALL WRITE4(TRIGGER,M1,M2,M3,KCHECK)
283     GO TO 100
284     117  WRITE(7,12)
285     CALL GETNUM(A) $IFLAG=A(1)
286     WRITE(7,15)
287     CALL GETNUM(A) $LFLAG=A(1)
288     WRITE(7,25)
289     25  FORMAT(*U3IN,U3FN*)
290     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(C(17),X1,X2,X3,C(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,DE1)
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,DE3)
330     CALL DPSPTL(D(18),X1,X2,X3,D(7),D(8),DA4,DE4)
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     P=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)

```

```

349      1 -(X2**2))/(R*((X2**2)+(P3**2)))+(((X1-P1)**2)+(P3**2))
350      1 /(R**3))+((X2**2)+(P3**2))/(R*((X1-P1)**2)+(X2**2)))
351      RETURN
352      END
353      SUBROUTINE WRITE1
354      WRITE(7,1)
355      1 FORMAT(*SPECIFY INITIAL AND FINAL SLIP VALUES,*,/,
356      1 *RIGHT-LATERAL STRIKE-SLIP IS <0.*/,,
357      1 *DIP-SLIP >0. FOR *X2>0. SIDE* DOWN*,/,
358      1 *FORMAT=U1IN,U1FN,U3IN,U3FN*)
359      RETURN
360      END
361      SUBROUTINE WRITE2(C)
362      DIMENSION C(40),A(30)
363      WRITE(7,3)
364      3 FORMAT(*SPECIFY INITIAL AND FINAL COORDINATES*,/,
365      1 *OF DIP-SLIP ZONE CORNERS AS INDICATED.*/,,
366      1 *D1X1IN,D1X3IN,D2X3IN,D3X1IN,U1X1FN,D1X3FN,D2X3FN,D3X1FN*)
367      CALL GETNUM(A)  IC(1)=A(1)  IC(2)=A(2)  IC(4)=A(3)  IC(5)=A(4)
368      C(9)=A(5)  IC(10)=A(6)  IC(12)=A(7)  IC(13)=A(8)
369      C(3)=C(1)  IC(7)=C(5)  IC(8)=C(4)  IC(8)=C(2)
370      C(11)=C(9)  IC(16)=C(10)  IC(14)=C(12)  IC(15)=C(13)
371      RETURN
372      END
373      SUBROUTINE WRITE3(C)
374      REAL A(30)
375      DIMENSION C(40)
376      WRITE(7,8)
377      8 FORMAT(*SPECIFY INITIAL AND FINAL VALUES *,/,
378      1 *OF STRIKE-SLIP ZONE COORDINATES AS INDICATED*,/,
379      1 *C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN*)
380      CALL GETNUM(A)  IC(17)=A(1)  IC(18)=A(2)  IC(20)=A(3)  IC(21)=A(4)
381      C(25)=A(5)  IC(26)=A(6)  IC(28)=A(7)  IC(29)=A(8)
382      C(19)=C(17)  IC(22)=C(20)  IC(23)=C(21)  IC(24)=C(18)
383      C(27)=C(25)  IC(30)=C(28)  IC(31)=C(29)  IC(32)=C(26)
384      RETURN
385      END
386      SUBROUTINE WRITE4(TRIGGR,M1,M2,M3,KCHECK)
387      REAL A(30)
388      M1=100  M2=100  M3=100
389      WRITE(7,150)
390      150 FORMAT(*DO YOU WISH THE *TRIGGER* OPTION?*,/,
391      1 **YES* OR *NO ***)
392      READ(7,151)TRIGGR
393      151 FORMAT(A3)
394      KCHECK=0
395      IF(TRIGGR.EQ.3)GO TO 160
396      WRITE(7,157)
397      157 FORMAT(*SPECIFY *TRIGGER* PARAMETERS IN THE FORM*,/,
398      1 *D(M1).GF.C(M2)**)
399      CALL GETNUM(A)  M1=A(1)  M2=A(2)
400      WRITE(7,158)
401      158 FORMAT(*SPECIFY WHICH SLIP COMPONENT IS TO BE =0.*/,,
402      1 *WHEN D(M1).LT.C(M2)*,/,
403      1 *STRIKE-SLIP=S, DIP-SLIP=C*)
404      READ(7,159)SLIP
405      159 FORMAT(A1)
406      IF(SLIP.EQ.1)M3=17

```



```

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,E=XPND)
9  XPND.
10 FIN.
11 FOR
12   PROGRAM XPND(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
13   COMMON/TVPCOL/TVPUL(8)
14   COMMON/TVTUNE/ITUNE(30)
15   COMMON/JPLOT/XLT,XPT,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 C(20),C(40),A(30),AREAS(500),AREAD(500)
21   DATA KTIME,FTIME,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).OF.0000 0010 0000 0000 0000B
24   IFET(8)=IFET(8).CR.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 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(TFIGGR,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=ANGL: 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*)

```



```

59      CALL GETNUM(A) $MVAR1=A(1) $MVAR2=A(2)
60      IF(C(35).EQ.0..AND.C(36).EQ.0.)GO TO 151
61      IF(C(33).EQ.0..AND.C(34).EQ.0.)GO TO 152
62      GC TO 153
63      151 DO 154 I=1,16
64      154 C(I)=0.
65      GO TO 153
66      152 DO 155 I=16,32
67      155 C(I)=0.
68      153 CONTINUE
69      NEND=NUM-IJKL $TAU=- (NEND-IJKL)/6.
70      DO 103 I=IJKL,NEND
71      DO 104 K=1,16
72      IF(K.LE.8)L1=K $IF(K.LE.8)L2=K+8
73      IF(K.GT.8.AND.K.LE.16)L1=K+8
74      IF(K.GT.8.AND.K.LE.16)L2=K+16
75      104 D(K)=(((C(L2)-C(L1))/(NEND-IJKL))*(I-IJKL))+C(L1)
76      IF(LFLAG.EQ.1.OR.LFLAG.EQ.3)D(17)=((1.-EXP((I-IJKL)/TAU))*(C(34)-C
77      1(33)))+C(33)
78      IF(LFLAG.EQ.2.OR.LFLAG.EQ.3)D(18)=((1.-EXP((I-IJKL)/TAU))*(C(36)-C
79      1(35)))+C(35)
80      IF(LFLAG.EQ.1)D(18)=(((C(36)-C(35))/(NEND-IJKL))*(I-IJKL))+C(35)
81      IF(LFLAG.EQ.2)D(17)=(((C(34)-C(33))/(NEND-IJKL))*(I-IJKL))+C(33)
82      IF(LFLAG.EQ.4)D(17)=(((C(34)-C(33))/(NEND-IJKL))*(I-IJKL))+C(33)
83      IF(LFLAG.EQ.4)D(18)=(((C(36)-C(35))/(NEND-IJKL))*(I-IJKL))+C(35)
84      IF(TRIGGR.NE.3HYES)GO TO 156
85      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2))GO TO 114
86      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2))KCHECK=KCHECK+1
87      IF(KCHECK.EQ.1)KTIME=NEND-I
88      IF(KCHECK.EQ.1)RTIME=I $IF(M3.EQ.18)GOTO 180
89      DO 181 K=9,16
90      L1=K+8 $L2=K+16
91      181 D(K)=(((C(L2)-C(L1))/(KTIME))*(I-RTIME))+C(L1)
92      GO TO 114
93      180 DO 182 K=1,8
94      L1=K $L2=K+8
95      182 D(K)=(((C(L2)-C(L1))/(KTIME))*(I-RTIME))+C(L1)
96      114 IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2))D(M3)=0.
97      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.M3.EQ.17)GO TO 9
98      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.M3.EQ.18)GO TO 10
99      GO TO 11
100     9 DO 16 M=9,16
101     16 D(M)=C(M+8) $GOTO 11
102     10 DO 18 M=1,8
103     18 D(M)=C(M)
104     11 IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.C(M3).EQ.0.)GO TO 156
105     IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.1
106     1 .AND.M3.EQ.17)D(M3)=((1.-EXP((I-RTIME)/
107     1 (-KTIME/6.)))*C(34)-C(33))+C(33)
108     IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.3
109     1 .AND.M3.EQ.17)D(M3)=((1.-EXP((I-RTIME)/(-KTIME/6.)))*C(34)-C(33
110     1)))+C(33)
111     IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.2
112     1 .AND.M3.EQ.18)D(M3)=((1.-EXP((I-RTIME)/
113     1 (-KTIME/6.)))*C(36)-C(35))+C(35)
114     IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.LFLAG.EQ.3.AND.
115     1 M3.EQ.18)D(M3)=((1.-EXP((I-RTIME)/(-KTIME/6.)))*C(36)-C(35))+C(
116     135)

```

```

117     IF (TRIGGER.EQ.3.HYES.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 (TRIGGER.EQ.3.HYES.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 (TRIGGER.EQ.3.HYES.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 (TRIGGER.EQ.3.HYES.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(0,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(I).GT.360.) TAZM(I)=TAZM(I)-360.
171     TEWMID=TEW(1) $TNSMID=TNS(1) $TAMPID=TAMP(1) $TAZMID=TAZM(1)
172     C4X1MD=C4X1(1) $D4X1MD=D4X1(1)
173     TEWMIN=TEWMAX=TEWMID $TNSMIN=TNSMAX=TNSMID
174     TAMPMN=TAMPMX=TAMPID $TAZMNN=TAZMNN=TAZMID

```

```

175      C4X1MN=C4X1MX=C4X1MD $D4X1MN=D4X1MX=D4X1MD
176      DO 169 I=1,NUM
177      IF(C4X1(I).LT.C4X1MN)C4X1MN=C4X1(I)
178      IF(C4X1(I).GT.C4X1MX)C4X1MX=C4X1(I)
179      IF(D4X1(I).LT.D4X1MN)D4X1MN=D4X1(I)
180      IF(D4X1(I).GT.D4X1MX)D4X1MX=D4X1(I)
181      IF (TEW(I) .LT. TEWMIN)TEWMIN=TEW(I)
182      IF (TEW(I) .GT. TEWMAX)TEWMAX=TEW(I)
183      IF (TNS(I) .LT. TNSMIN)TNSMIN=TNS(I)
184      IF (TNS(I) .GT. TNSMAX)TNSMAX=TNS(I)
185      IF (TAMP(I) .LT. TAMPMN)TAMPMN=TAMP(I)
186      IF (TAMP(I) .GT. TAMPMX)TAMPMX=TAMP(I)
187      IF(TAZM(I) .LT. TAZMMN)TAZMMN=TAZM(I)
188      IF(TAZM(I) .GT. TAZMMX)TAZMMX=TAZM(I)
189      160 CONTINUE
190      WRITE(7,17)TEWMIN,TEWMAX,TNSMIN,TNSMAX,TAMPMN,TAMPMX,TAZMMN,TAZMMX
191      17 FORMAT(*MIN./MAX. VALUES OF EW COMPONENT*,2X,E10.3,2X,E10.3,/,
192      1 *MIN./MAX. VALUES OF NS COMPONENT*,2X,E10.3,2X,E10.3,/,
193      1 *MIN./MAX. VALUES OF AMPLITUDE*,5X,E10.3,2X,E10.3,/,
194      1 *MIN./MAX. VALUES OF AZIMUTH*,3X,F10.3,2X,F10.3,/,
195      1 *(NOTE TILT AMPLITUDES ARE IN MICRORADIANS*,/,
196      1 *AZIMUTH IN DEGREES*)
197      WRITE(7,170)
198      170 FORMAT(*THE FOLLOWING ARE PLOTS OF THE EW AND NS*,/,
199      1 *COMPONENTS OF TILT, AND THE TILT AMPLITUDE*,/,
200      1 *AND AZIMUTH (MEASURED CLOCKWISE FROM NORTH)*,/,
201      1 *0=RE-START, 1=CONTINUE*)
202      CALL GETNUM(A) $IKFLAG=A(1)
203      IF(IKFLAG.EQ.0) GO TO 113
204      LU=7 $LNLGX=1 $LNLY=1 $NCLX=2 $NCLY=2
205      WRITE(7,199)
206      199 FORMAT(*WRITE PLOT TITLE, 80 CHARACTERS*)
207      READ(7,200){LTITL(JM),JM=1,8)
208      200 FORMAT(8A10)
209      KX(1)=10HTIME-ARBIT
210      KX(2)=10HRRARY UNITS
211      KY(1)=10H EW TILT
212      KY(2)=10H $MAJX=5 $MAJY=10 $LTITL2(1)=1
213      903 XLT=T(1)$XRT=T(NUM) $YLO=TEWMIN $YUP=TEWMAX
214      WRITE(7,70)
215      70 FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
216      READ(7,66)CHARAC $IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H)GO TO 71
217      WRITE(7,72)
218      72 FORMAT(*MIN/MAX X VALUES*)
219      CALL GETNUM(A) $XLT=A(1) $XRT=A(2)
220      71 WRITE(7,73)
221      73 FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
222      READ(7,66)CHARAC $IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H)GC TO 74
223      WRITE(7,75)
224      75 FORMAT(*MIN/MAX Y VALUES*)
225      CALL GETNUM(A) $YLO=A(1) $YUP=A(2)
226      74 IF(TEWMIN.EQ.TEWMAX)YLO=YUP-1.
227      IF(TEWMIN.EQ.TEWMAX)YUP=YUP+1.
228      WRITE(7,1)
229      1 FORMAT(*SKIP PLOT OF EW TILT?*)
230      READ(7,66)IJVAR
231      66 FORMAT(A1)
232      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H)CALL PLOTS(TEW,T,1,NUM)

```

```

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

```

```

291      YLO=C(35) $YUP=C(36)
292      IF(C(35).EQ.C(36))YLO=YUP-1.
293      IF(C(35).EQ.C(36))YUP=YUP+1.
294      WRITE(7,917) $READ(7,66)IJVAR
295  917  FORMAT(*SKIP PLOT OF DIP SLIP COMPONENT?*)
296      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(SLIPU3,T,1,NUM)
297      WRITE(7,919) $CALL GETNUM(A) $IKFLAG=A(1)
298  919  FORMAT(*0=RE-START*)
299      IF(IKFLAG.EQ.0)GO TO 113
300      KY(1)=10HCORNER 1
301      IF(MVAR1.EQ.1)KY(2)=10H D1X1=D2X1
302      IF(MVAR1.EQ.2)KY(2)=10H D1X3=D4X3
303      IF(MVAR1.EQ.4)KY(2)=10H D2X3=D3X3
304      IF(MVAR1.EQ.5)KY(2)=10H D3X1=D4X1
305      IF(MVAR1.EQ.9)KY(2)=10H C1X1=C2X1
306      IF(MVAR1.EQ.10)KY(2)=10H C1X3=C4X3
307      IF(MVAR1.EQ.12)KY(2)=10H C2X3=C3X3
308      IF(MVAR1.EQ.13)KY(2)=10H C3X1=C4X1
309      YLO=C4X1MN $YUP=C4X1MX
310      IF(C4X1MN.EQ.C4X1MX)YLO=YUP-1.
311      IF(C4X1MN.EQ.C4X1MX)YUP=YUP+1.
312      WRITE(7,918) $READ(7,66)IJVAR
313  918  FORMAT(*SKIP PLOT OF CORNER 1?*)
314      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(C4X1,T,1,NUM)
315      KY(1)=10HCORNER 2
316      IF(MVAR2.EQ.1)KY(2)=10H D1X1=D2X1
317      IF(MVAR2.EQ.2)KY(2)=10H D1X3=D4X3
318      IF(MVAR2.EQ.4)KY(2)=10H D2X3=D3X3
319      IF(MVAR2.EQ.5)KY(2)=10H D3X1=D4X1
320      IF(MVAR2.EQ.9)KY(2)=10H C1X1=C2X1
321      IF(MVAR2.EQ.10)KY(2)=10H C1X3=C4X3
322      IF(MVAR2.EQ.12)KY(2)=10H C2X3=C3X3
323      IF(MVAR2.EQ.13)KY(2)=10H C3X1=C4X1
324      YLO=D4X1MN $YUP=D4X1MX
325      IF(D4X1MN.EQ.D4X1MX)YLO=YUP-1.
326      IF(D4X1MN.EQ.D4X1MX)YUP=YUP+1.
327      WRITE(7,920) $READ(7,66)IJVAR
328  920  FORMAT(*SKIP PLOT OF CORNER 2?*)
329      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(D4X1,T,1,NUM)
330      KY(1)=10HS.S. AREA
331      YLO=(C(23)-C(17))*(C(20)-C(18))
332      YUP=(C(31)-C(25))*(C(28)-C(26))
333      ABC=YLO $IF(YLO.EQ.YUP)ABC=YLO-1. $IF(YLO.EQ.YUP)YUP=YUP+1.
334      YLO=ABC $WRITE(7,921) $READ(7,66)IJVAR
335  921  FORMAT(*SKIP PLOT OF STRIKE SLIP AREA?*)
336      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(AREAS,T,1,NUM)
337      KY(1)=10HD.S. AREA
338      YLO=(C(7)-C(1))*(C(4)-C(2))
339      YUP=(C(15)-C(9))*(C(12)-C(10))
340      ABC=YLO $IF(YLO.EQ.YUP)ABC=YLO-1. $IF(YLO.EQ.YUP)YUP=YUP+1.
341      YLO=ABC $WRITE(7,922) $READ(7,66)IJVAR
342  922  FORMAT(*SKIP PLOT OF DIP SLIP AREA?*)
343      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(AREAD,T,1,NUM)
344      WRITE(7,68)(C(I),I=1,36)
345      68  FORMAT(9(4F12.2,/))
346      WRITE(7,69)IFLAG,IKFLAG,LFLAG,IJKL,NUM,NENC,M1,M2,M3,KTIME,K7,K8,
347      1  TAU,RTIME
348      69  FORMAT(7I10,/,5I10,2F10.3)

```

```

349 113 WRITE (7,115)
350 115 FORMAT(*1=RE-START WITH ALL NEW VALUES*,/,
351 1 *2=RE-START WITH NEW STRIKE-SLIP VALUE AND ZONE COORDINATES*,/,
352 1 *3=RE-START WITH NEW DIP-SLIP VALUE AND ZONE COORDINATES*,/,
353 1 *4=RE-START WITH NEW STRIKE-SLIP VALUE ONLY*,/,
354 1 *5=RE-START WITH NEW DIP-SLIP VALUE ONLY*,/,
355 1 *6=RE-START WITH NEW TILTMETER COORDINATES ONLY*,/,
356 1 *7=STOP*)
357 CALL GETNUM(A) $MSTART=A(1)
358 IF(MSTART.EQ.2)GOTO116 $IF(MSTART.EQ.3)GOTO117
359 IF(MSTART.EQ.4)GOTO118 $IF(MSTART.EQ.5)GOTO119
360 IF(MSTART.EQ.6)GOTO120 $IF(MSTART.EQ.7)GOTO630
361 GO TO 20
362 116 WRITE (7,12)
363 CALL GETNUM(A) $IFLAG=A(1) $WRITE(7,15)
364 CALL GETNUM(A) $LFLAG=A(1)
365 WRITE(7,23)
366 23 FORMAT(*U1IN,U1FN*)
367 CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
368 CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
369 GO TO 100
370 117 WRITE(7,12)
371 CALL GETNUM(A) $IFLAG=A(1)
372 WRITE(7,15)
373 CALL GETNUM(A) $LFLAG=A(1)
374 WRITE(7,25)
375 25 FORMAT(*U3IN,U3FN*)
376 CALL GETNUM(A) $C(35)=A(1) $C(36)=A(2)
377 CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
378 GO TO 122
379 118 WRITE(7,12)
380 CALL GETNUM(A) $IFLAG=A(1)
381 WRITE(7,15)
382 CALL GETNUM(A) $LFLAG=A(1)
383 WRITE(7,23)
384 CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
385 CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
386 GO TO 123
387 119 WRITE(7,12)
388 CALL GETNUM(A) $IFLAG=A(1)
389 WRITE(7,15)
390 CALL GETNUM(A) $LFLAG=A(1)
391 WRITE(7,25)
392 CALL GETNUM(A) $C(35)=A(1) $C(36)=A(2)
393 CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
394 GO TO 123
395 120 WRITE(7,4)
396 CALL GETNUM(A) $X1=A(1) $X2=A(2)
397 CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
398 GO TO 123
399 630 STOP
400 END
401 SUBROUTINE CMPTLT(D,X1,X2,X3,T1,T2)
402 DIMENSION D(20)
403 A1=A2=A3=A4=B1=B2=B3=B4=C.
404 DA1=DA2=CA3=DA4=DB1=DB2=DB3=DB4=0.
405 U1=D(17)
406 U3=D(18)

```

```

407     IF(U1.EQ.0.)GO TO 1
408     CALL TILT(D(17),X1,X2,X3,C(11),D(12),A1,B1)
409     CALL TILT(D(17),X1,X2,X3,D(9),C(10),A2,B2)
410     CALL TILT(D(17),X1,X2,X3,D(13),D(14),A3,B3)
411     CALL TILT(D(17),X1,X2,X3,C(15),D(16),A4,B4)
412     1 IF(U3.EQ.0.)GO TO 2
413     CALL DPSFTL(D(18),X1,X2,X3,D(3),D(4),DA1,DE1)
414     CALL CPSFTL(D(18),X1,X2,X3,D(1),D(2),DA2,DE2)
415     CALL DPSFTL(D(18),X1,X2,X3,D(5),D(6),DA3,DE3)
416     CALL DPSFTL(D(18),X1,X2,X3,D(7),D(8),DA4,DE4)
417     2 T1=A1-A2-A3+A4+DA1-DA2-DA3+DA4
418     T2=B1-B2-B3+B4+DB1-DB2-DB3+DB4
419     RETURN
420     END
421     SUBROUTINE TILT(U1,X1,X2,X3,P1,P3,T1,T2)
422     R=SQRT((X1-P1)**2+X2**2+(X3-P3)**2)
423     RP=R+P3
424     T1=(U1/12.5664)*(X2*(X1-P1)*(R*RP-(R+2.*P3)*(2.*R+P3)))/
425     1 (R**3*RP**2)
426     T2=(U1/12.5664)*(X2**2*(R*RP-(R+2.*P3)*(2.*R+P3))/(R**3*RP**2)
427     1 +(R+2.*P3)/(R*RP))
428     RETURN
429     END
430     SUBROUTINE DPSFTL(U3,X1,X2,X3,P1,P3,DT1,DT2)
431     R=SQRT(((X1-P1)**2)+(X2**2)+((X3-P3)**2))
432     DT1=(U3/6.28318)*((X2*P3)/R)*((1./(R**2))-(1./(((X1-P1)
433     1 **2)+(X2**2))))
434     DT2=(U3/6.28318)*(((X1-P1)*P3)/((X2**2)+(P3**2)))*(((P3**2)
435     1 -(X2**2))/(R*((X2**2)+(P3**2)))+(((X1-P1)**2)+(P3**2))
436     1 /((R**3))+((X2**2)+(P3**2))/(R*((X1-P1)**2)+(X2**2)))
437     RETURN
438     END
439     SUBROUTINE WRITE1
440     WRITE(7,1)
441     1 FORMAT(*SPECIFY INITIAL AND FINAL SLIP VALLES,*,/,
442     1 *RIGHT-LATERAL STRIKE-SLIP IS <0.*/,,
443     1 *DIP-SLIP >0. FOR *X2>0. SIDE* DOWN*/,,
444     1 *FORMAT=U1IN,U1FN,U3IN,U3FN*)
445     RETURN
446     END
447     SUBROUTINE WRITE2(C)
448     DIMENSION C(40),A(30)
449     WRITE(7,3)
450     3 FORMAT(*SPECIFY INITIAL AND FINAL COORDINATES*,/,
451     1 *OF DIP-SLIP ZONE CORNERS AS INDICATED.*/,,
452     1 *D1X1IN,D1X3IN,D2X3IN,D3X1IN,U1X1FN,D1X3FN,D2X3FN,D3X1FN*)
453     CALL GETNUM(A) %C(1)=A(1) %C(2)=A(2) %C(4)=A(3) %C(5)=A(4)
454     C(9)=A(5) %C(10)=A(6) %C(12)=A(7) %C(13)=A(8)
455     C(3)=C(1) %C(7)=C(5) %C(6)=C(4) %C(8)=C(2)
456     C(11)=C(9) %C(16)=C(10) %C(14)=C(12) %C(15)=C(13)
457     RETURN
458     END
459     SUBROUTINE WRITE3(C)
460     REAL A(30)
461     DIMENSION C(40)
462     WRITE(7,8)
463     8 FORMAT(*SPECIFY INITIAL AND FINAL VALUES *,/,
464     1 *OF STRIKE-SLIP ZONE COORDINATES AS INDICATED*/,,

```

```

465      1 *C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN*)
466      CALL GETNUM(A) $C(17)=A(1) $C(18)=A(2) $C(20)=A(3) $C(21)=A(4)
467      C(25)=A(5) $C(26)=A(6) $C(28)=A(7) $C(29)=A(8)
468      C(19)=C(17) $C(22)=C(20) $C(23)=C(21) $C(24)=C(18)
469      C(27)=C(25) $C(30)=C(28) $C(31)=C(29) $C(32)=C(26)
470      RETURN
471      END
472      SUBROUTINE WRITE4 (TRIGGR,M1,M2,M3,KCHECK)
473      REAL A(30)
474      M1=100 $M2=100 $M3=100
475      WRITE(7,150)
476      150 FORMAT(*DO YOU WISH THE *TRIGGER* OPTION?*,/,
477      1 *YES* OR *NO *)
478      READ(7,151)TRIGGR
479      151 FORMAT(A3)
480      KCHECK=0
481      IF(TRIGGR.EQ.3)GO TO 160
482      WRITE(7,157)
483      157 FORMAT(*SPECIFY *TRIGGER* PARAMETERS IN THE FORM*,/,
484      1 *D(M1).GE.C(M2)* )
485      CALL GETNUM(A) $M1=A(1) $M2=A(2)
486      WRITE(7,158)
487      158 FORMAT(*SPECIFY WHICH SLIP COMPONENT IS TO BE =0.*/,
488      1 *WHEN D(M1).LT.C(M2)*,/,
489      1 *STRIKE-SLIP=C, DIP-SLIP=D*)
490      READ(7,159)SLIP
491      159 FORMAT(A1)
492      IF(SLIP.EQ.1)M3=17
493      IF(SLIP.EQ.1)M3=18
494      160 CONTINUE
495      RETURN
496      END
497      SUBROUTINE GETNUM(R)
498      DIMENSION R(1),L(80)
499      READ(7,9)L $ I=J=0
500      6 J=J+1 $ N=P=S=0 $ M=F=1
501      5 I=I+1 $ IF(I.GT.80)RETURN $ D=L(I) $ K=4
502      IF(D.EQ.38)K=2 $ IF(D.GE.27.A.D.LE.36)K=1
503      IF(D.EQ.47)K=3 $ K=K+S $ GOTO(1,2,3,5,1,4,3,4)K
504      1 N=N*10+D-27 $ S=4 $ GOTO 5
505      2 M=-1 $ S=4 $ GOTO 5
506      3 P=I $ S=4 $ GOTO 5
507      4 IF(P.NE.0)F=10.** (I-P-1) $ R(J)=N/F*M $ GOTO 6
508      9 FORMAT(80F1)
509      END

```


LISTING OF XPND02

A70a

```

1  DELETE(LGO,OUTPUT,XPND)
2  XPND.
3  CXIT.
4  LIBCOPY(GRAPHIC, TXLGO/RR, TXLGO)
5  LIBCOPY(JDRAT, NPLGO/FP, 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(TAPE TTY=201, FILM=TAFETTY, TAPE7=TAFETTY)
13     COMMON/TVPCOL/TVPOL(8)
14     COMMON/TVTUNE/ITUNE(30)
15     COMMON/JPLOT/XLT, XRT, YLU, 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 SLIPU1(500), SLIPU3(500), O4X1(500), O4X1(500)
20     DIMENSION D(20), C(40), A(30), AREAS(500), AREAD(500)
21     DATA KTIME, FTIME, 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 COMPONENT INCREMENTED EXPONENTIALLY*,/,
33     1 *3=BOTH SLIP COMPONENTS INCREMENTED EXPONENTIALLY*,/,
34     1 *4=NEITHER SLIP COMPONENT INCREMENTED EXPONENTIALLY*)
35     CALL GETNUM(A) $IFLAG=A(1)
36     CALL WRITE4(TFIGGER, 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*)

```

```

59      CALL GETNUM(A) $MVAR1=A(1) $MVAR2=A(2)
60      IF(C(35).EQ.0..AND.C(36).EQ.0.)GO TO 151
61      IF(C(33).EQ.0..AND.C(34).EQ.0.)GO TO 152
62      GO TO 153
63      151 DO 154 I=1,16
64      154 C(I)=0.
65      GO TO 153
66      152 DO 155 I=16,32
67      155 C(I)=0.
68      153 CONTINUE
69      NEND=NUM-IJKL $TAU=- (NEND-IJKL)/6.
70      DO 103 I=IJKL,NEND
71      DO 104 K=1,16
72      IF(K.LE.8)L1=K $IF(K.LE.8)L2=K+8
73      IF(K.GT.8.AND.K.LE.16)L1=K+8
74      IF(K.GT.8.AND.K.LE.16)L2=K+16
75      104 D(K)=(((C(L2)-C(L1))/(NEND-IJKL))*(I-IJKL))+C(L1)
76      IF(LFLAG.EQ.1.OR.LFLAG.EQ.3)D(17)=((1.-EXP((I-IJKL)/TAU))*(C(34)-C
77      1(33)))+C(33)
78      IF(LFLAG.EQ.2.OR.LFLAG.EQ.3)D(18)=((1.-EXP((I-IJKL)/TAU))*(C(36)-C
79      1(35)))+C(35)
80      IF(LFLAG.EQ.1)D(18)=(((C(36)-C(35))/(NEND-IJKL))*(I-IJKL))+C(35)
81      IF(LFLAG.EQ.2)D(17)=(((C(34)-C(33))/(NEND-IJKL))*(I-IJKL))+C(33)
82      IF(LFLAG.EQ.4)D(17)=(((C(34)-C(33))/(NEND-IJKL))*(I-IJKL))+C(33)
83      IF(LFLAG.EQ.4)D(18)=(((C(36)-C(35))/(NEND-IJKL))*(I-IJKL))+C(35)
84      IF(TRIGGR.NE.3HYES)GO TO 156
85      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2))GO TO 114
86      IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2))KCHECK=KCHECK+1
87      IF(KCHECK.EQ.1)KTIME=NEND-I
88      IF(KCHECK.EQ.1)RTIME=I $IF(M3.EQ.18)GOTO 180
89      DO 181 K=9,16
90      L1=K+8 $L2=K+16
91      181 D(K)=(((C(L2)-C(L1))/(KTIME))*(I-RTIME))+C(L1)
92      GO TO 114
93      180 DO 182 K=1,8
94      L1=K $L2=K+8
95      182 D(K)=(((C(L2)-C(L1))/(KTIME))*(I-RTIME))+C(L1)
96      114 IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2))D(M3)=0.
97      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.M3.EQ.17)GO TO 9
98      IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.M3.EQ.18)GO TO 10
99      GO TO 11
100     9 DO 16 M=9,16
101     16 D(M)=C(M+8) $GOTO 11
102     10 DO 18 M=1,8
103     18 D(M)=C(M)
104     11 IF(TRIGGR.EQ.3HYES.AND.D(M1).GE.C(M2).AND.C(M3).EQ.0.)GO TO 156
105     IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2) .AND.LFLAG.EQ.1
106     1 .AND.M3.EQ.17)D(M3)=((1.-EXP((I-RTIME)/
107     1 (-KTIME/6.)))*(C(34)-C(33)))+C(33)
108     IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.LFLAG.EQ.3
109     1 .AND. M3.EQ.17)D(M3)=((1.-EXP((I-RTIME)/(-KTIME/6.)))*(C(34)-C(33
110     1)))+C(33)
111     IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.LFLAG.EQ.2
112     1 .AND.M3.EQ.18)D(M3)=((1.-EXP((I-RTIME)/
113     1 (-KTIME/6.)))*(C(36)-C(35)))+C(35)
114     IF(TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.LFLAG.EQ.3.AND.
115     1 M3.EQ.18)D(M3)=((1.-EXP((I-RTIME)/(-KTIME/6.)))*(C(36)-C(35)))+C(
116     1 35)

```

```

117     IF (TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.LFLAG.EG.1
118 1 .AND.M3.EG.18) D(M3) = (((C(36)-C(35))/
119 1 (KTIME)) * (I-RTIME)) + C(35)
120     IF (TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.LFLAG.EG.4.AND.M3.EG.18)
121 1 D(M3) = (((C(36)-C(35))/(KTIME)) * (I-RTIME)) + C(35)
122     IF (TRIGGR.EG.3HYES.AND.D(M1).LT.C(M2).AND.LFLAG.EG.2
123 1 .AND.M3.EG.17) D(M3) = (((C(34)-C(33))/
124 1 (KTIME)) * (I-RTIME)) + C(33)
125     IF (TRIGGR.EQ.3HYES.AND.D(M1).LT.C(M2).AND.M3.EG.17.AND.LFLAG.EG.4)
126 1 D(M3) = (((C(34)-C(33))/(KTIME)) * (I-RTIME)) + C(33)
127 156 CALL CMPTLT(D,X1,X2,X3,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) = D(MVAR1) $D4X1(I) = D(MVAR2) $T(I) = I $TFW(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.EG.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     TEW(I) = -A * ST + B * CT
164     TAMP(I) = SQRT((TEW(I)**2) + (TNS(I)**2))
165     IF (TNS(I).EQ.0.) TNS(I) = 1.E-20
166     TAZM(I) = (ATAN(TEW(I)/TNS(I))) * (180./3.1415926)
167     IF (TNS(I).LT.0.) TAZM(I) = TAZM(I) + 180.
168     IF (TAZM(I).LT.0.) TAZM(I) = TAZM(I) + 360.
169 21 IF (TAZM(I).GT.360.) TAZM(I) = TAZM(I) - 360.
170     TEWMID = TEW(1) $TNSMID = TNS(1) $TAMPMD = TAMP(1) $TAZMMD = TAZM(1)
171     C4X1MD = C4X1(1) $D4X1MD = D4X1(1)
172     TFWMIN = TFWMAX = TEWMID $TNSMIN = TNSMAX = TNSMID
173     TAMPMN = TAMPMX = TAMPMD $TAZMMN = TAZMMX = TAZMMD
174     C4X1MN = C4X1MX = C4X1MD $D4X1MN = D4X1MX = D4X1MD

```

```

175      DO 160 I=1,NUM
176      IF(C4X1(I).LT.C4X1MN)C4X1MN=C4X1(I)
177      IF(C4X1(I).GT.C4X1MX)C4X1MX=C4X1(I)
178      IF(D4X1(I).LT.D4X1MN)D4X1MN=D4X1(I)
179      IF(D4X1(I).GT.D4X1MX)D4X1MX=D4X1(I)
180      IF (TEW(I) .LT. TEWMIN)TEWMIN=TEW(I)
181      IF (TEW(I) .GT. TEWMAX)TEWMAX=TEW(I)
182      IF (TNS(I) .LT. TNSMIN)TNSMIN=TNS(I)
183      IF (TNS(I) .GT. TNSMAX)TNSMAX=TNS(I)
184      IF (TAMP(I) .LT. TAMPMN)TAMPMN=TAMP(I)
185      IF (TAMP(I) .GT. TAMPMX)TAMPMX=TAMP(I)
186      IF(TAZM(I) .LT. TAZMMN)TAZMMN=TAZM(I)
187      IF(TAZM(I) .GT. TAZMMX)TAZMMX=TAZM(I)
188      160 CONTINUE
189      WRITE (7,17)TEWMIN,TEWMAX,TNSMIN,TNSMAX,TAMPMN,TAMPMX,TAZMMN,TAZMMX
190      17 FORMAT(*MIN./MAX. VALUES OF EW COMPONENT*,2X,E10.3,2X,E10.3,/,
191      1 *MIN./MAX. VALUES OF NS COMPONENT*,2X,E10.3,2X,E10.3,/,
192      1 *MIN./MAX. VALUES OF AMPLITUDE*,5X,E10.3,2X,E10.3,/,
193      1 *MIN./MAX. VALUES OF AZIMLTH*,3X,F10.3,2X,F10.3,/,
194      1 *(NOTE TILT AMPLITUDES ARE IN MICRORADIANS*,/,
195      1 *AZIMUTH IN DEGREES*)
196      WRITE (7,170)
197      170 FORMAT(*THE FOLLOWING ARE PLOTS OF THE EW AND NS*,/,
198      1 *COMPONENTS OF TILT, AND THE TILT AMPLITUDE*,/,
199      1 *AND AZIMUTH (MEASURED CLOCKWISE FROM NORTH).*,/,
200      1 *0=RE-START, 1=CONTINUE*)
201      CALL GETNUM(A) $IKFLAG=A(1)
202      IF(IKFLAG.EQ. 0) GO TO 113
203      LU=7 $LNLGX=1 $LNLGY=1 $NCLX=2 $NCLY=2
204      WRITE (7,199)
205      199 FORMAT(*WRITE PLOT TITLE, 80 CHARACTERS*)
206      READ (7,200)(LTITL(JM),JM=1,8)
207      200 FORMAT(8A10)
208      KX(1)=10H TIME-ARBIT
209      KX(2)=10H PARY UNITS
210      KY(1)=10H EW TILT
211      KY(2)=10H $MAJX=5 $MAJY=10 $LTITL2(1)=1
212      903 XLT=T(1)$XPT=T(NUM) $YLO=TEWMIN $YUP=TEWMAX
213      WRITE (7,70)
214      70 FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
215      READ (7,66)CHARAC $IF (CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GO TO 71
216      WRITE (7,72)
217      72 FORMAT(*MIN/MAX X VALUES*)
218      CALL GETNUM(A) $XLT=A(1) $XRT=A(2)
219      71 WRITE (7,73)
220      73 FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
221      READ (7,66)CHARAC $IF (CHARAC .EQ.1HN.OR.CHARAC.EQ.1H )GO TO 74
222      WRITE (7,75)
223      75 FORMAT(*MIN/MAX Y VALUES*)
224      CALL GETNUM(A) $YLO=A(1) $YUP=A(2)
225      74 IF(TEWMIN.EQ.TEWMAX)YLO=YUP-1.
226      IF(TEWMIN.EQ.TEWMAX)YUP=YUP+1.
227      WRITE (7,1)
228      1 FORMAT(*SKIP PLOT OF EW TILT?*)
229      READ (7,66)IJVAR
230      66 FORMAT(A1)
231      IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TEW,T,1,NUM)
232      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) $READ(7,66)CHARAC
237     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC710
238     WRITE(7,72)$CALL GETNUM(A)$XLT=A(1)$XRT=A(2)
239     710 WRITE(7,73)$READ(7,66)CHARAC
240     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC740
241     WRITE(7,75)$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 FLOT*)
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=TAMPXN $YUP=TAMPX $WRITE(7,70)$READ(7,66)CHARAC
254     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC741
255     WRITE(7,72)$CALL GETNUM(A) $XLT=A(1)$XRT=A(2)
256     741 WRITE(7,73)$READ(7,66)CHARAC
257     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC742
258     WRITE(7,75)$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) $READ(7,66)IJVAR
262     912 FORMAT(*SKIP PLOT OF TILT AMPLITUDE?*)
263     IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAMP,1,1,NUM)
264     WRITE(7,19)$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)=10HPTH           $YLO=TAZMMN $YUP=TAZMMX
268     WRITE(7,70) $READ(7,66)CHARAC
269     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC743
270     WRITE(7,72)$CALL GETNUM(A)$XLT=A(1)$XRT=A(2)
271     743 WRITE(7,73)$READ(7,66)CHARAC
272     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC744
273     WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
274     744 IF(YLO.EQ.YUP)AAA=YLO$IF(YLO.EQ.AAA)YLP=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)=10HSTRIKESLIP $KY(2)=10HCOMPONENT
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)=10HSLIP SLIP C $KY(2)=10HCOMPONENT
290     YLO=C(35) $YUP=C(36)

```

```

291     IF(C(35).EQ.C(36))YLO=YUP-1.
292     IF(C(35).EQ.C(36))YUP=YUP+1.
293     WRITE(7,917) $FEAD(7,66)IJVAR
294 917  FORMAT(*SKIP PLOT OF DIP SLIP COMPONENT?*)
295     IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(SLIPU3,T,1,NUM)
296     WRITE(7,919) $CALL GETNUM(A) $IKFLAG=A(1)
297 919  FORMAT(*0=RE-START*)
298     IF(IKFLAG.EQ.0)GO TO 113
299     KY(1)=10HCORNER 1
300     IF(MVAR1.EQ.1)KY(2)=10H D1X1=D2X1
301     IF(MVAR1.EQ.2)KY(2)=10H D1X3=D4X3
302     IF(MVAR1.EQ.4)KY(2)=10H D2X3=D3X3
303     IF(MVAR1.EQ.5)KY(2)=10H D3X1=D4X1
304     IF(MVAR1.EQ.9)KY(2)=10H C1X1=C2X1
305     IF(MVAR1.EQ.10)KY(2)=10H C1X3=C4X3
306     IF(MVAR1.EQ.12)KY(2)=10H C2X3=C3X3
307     IF(MVAR1.EQ.13)KY(2)=10H C3X1=C4X1
308     YLO=C4X1MN $YUP=C4X1MX
309     IF(C4X1MN.EQ.C4X1MX)YLO=YUP-1.
310     IF(C4X1MN.EQ.C4X1MX)YUP=YUP+1.
311     WRITE(7,918) $READ(7,66)IJVAR
312 918  FORMAT(*SKIP PLOT OF CORNER 1?*)
313     IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(C4X1,T,1,NUM)
314     KY(1)=10HCORNER 2
315     IF(MVAR2.EQ.1)KY(2)=10H D1X1=D2X1
316     IF(MVAR2.EQ.2)KY(2)=10H D1X3=D4X3
317     IF(MVAR2.EQ.4)KY(2)=10H D2X3=D3X3
318     IF(MVAR2.EQ.5)KY(2)=10H D3X1=D4X1
319     IF(MVAR2.EQ.9)KY(2)=10H C1X1=C2X1
320     IF(MVAR2.EQ.10)KY(2)=10H C1X3=C4X3
321     IF(MVAR2.EQ.12)KY(2)=10H C2X3=C3X3
322     IF(MVAR2.EQ.13)KY(2)=10H C3X1=C4X1
323     YLO=D4X1MN $YUP=D4X1MX
324     IF(D4X1MN.EQ.D4X1MX)YLO=YUP-1.
325     IF(D4X1MN.EQ.D4X1MX)YUP=YUP+1.
326     WRITE(7,920) $READ(7,66)IJVAR
327 920  FORMAT(*SKIP PLOT OF CORNER 2?*)
328     IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(D4X1,T,1,NUM)
329     KY(1)=10HS.S. AREA
330     YLO=(C(23)-C(17))*(C(20)-C(18))
331     YUP=(C(31)-C(25))*(C(28)-C(26))
332     ABC=YLO $IF(YLO.EQ.YUP)ABC=YLO-1. $IF(YLO.EQ.YUP)YUP=YUP+1.
333     YLO=ABC $WRITE(7,921) $READ(7,66)IJVAR
334 921  FORMAT(*SKIP PLOT OF STRIKE SLIP AREA?*)
335     IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(AREAS,T,1,NUM)
336     KY(1)=10HD.S. AREA
337     YLO=(C(7)-C(1))*(C(4)-C(2))
338     YUP=(C(15)-C(9))*(C(12)-C(10))
339     ABC=YLO $IF(YLO.EQ.YUP)ABC=YLO-1. $IF(YLO.EQ.YUP)YUP=YUP+1.
340     YLO=ABC $WRITE(7,922) $READ(7,66)IJVAR
341 922  FORMAT(*SKIP PLOT OF DIP SLIP AREA?*)
342     IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(AREAD,T,1,NUM)
343     WRITE(7,68)(C(I),I=1,36)
344 68  FORMAT(9(4F12.2,/))
345     WRITE(7,69)IFLAG,IKFLAG,LFLAG,IJKL,NUM,NENC,M1,M2,M3,KTIME,K7,K8,
346     1 TAU,PTIME
347 69  FORMAT(7I10,/,5I10,2F10.3)
348 113  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)GOTC117
358 IF(MSTART.EQ.4)GOTC118 $IF(MSTART.EQ.5)GOTC119
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

```



```

407 CALL TILT(D(17),X1,X2,X3,C(11),D(12),A1,B1)
408 CALL TILT(D(17),X1,X2,X3,D(9),D(10),A2,B2)
409 CALL TILT(D(17),X1,X2,X3,D(13),D(14),A3,B3)
410 CALL TILT(D(17),X1,X2,X3,D(15),D(16),A4,B4)
411 1 IF(U3.EQ.0.)GO TO 2
412 CALL DPSPTL(D(18),X1,X2,X3,D(3),D(4),DA1,DE1)
413 CALL CPSFTL(D(18),X1,X2,X3,D(1),D(2),DA2,DB2)
414 CALL DPSPTL(D(18),X1,X2,X3,D(5),D(6),CA3,DE3)
415 CALL DPSPTL(D(18),X1,X2,X3,D(7),D(8),CA4,DE4)
416 2 T1=A1-A2-A3+A4+DA1-DA2-DA3+DA4
417 T2=B1-B2-B3+B4+DB1-DB2-DB3+DB4
418 RETURN
419 END
420 SUBROUTINE TILT(U1,X1,X2,X3,P1,P3,T1,T2)
421 R=SQRT((X1-P1)**2+X2**2+(X3-P3)**2)
422 RP=R+P3
423 T1=(U1/12.5664)*(X2*(X1-P1)*(R*RP-(R+2.*P3)*(2.*R+P3)))/
424 1 (R**3*RP**2)
425 T2=(U1/12.5664)*(X2**2*(R*RP-(R+2.*P3)*(2.*R+P3))/(R**3*RP**2)
426 1 +(R+2.*P3)/(R*RP))
427 RETURN
428 END
429 SUBROUTINE DPSPTL(U3,X1,X2,X3,P1,P3,DT1,DT2)
430 R=SQRT(((X1-P1)**2)+(X2**2)+((X3-P3)**2))
431 DT1=(U3/6.28318)*((X2*P3)/R)*((1./(R**2))-(1./(((X1-P1)
432 1 **2)+(X2**2))))
433 DT2=(U3/6.28318)*(((X1-P1)*P3)/((X2**2)+(P3**2)))*(((P3**2)
434 1 -(X2**2))/(R*((X2**2)+(P3**2)))+(((X1-P1)**2)+(P3**2))
435 1 /R**3)+((X2**2)+(P3**2))/(R*((X1-P1)**2)+(X2**2)))
436 RETURN
437 END
438 SUBROUTINE WRITE1
439 WRITE(7,1)
440 1 FORMAT(*SPECIFY INITIAL AND FINAL SLIP VALUES*,/,
441 1 *RIGHT-LATERAL STRIKE-SLIP IS <0.*,/,
442 1 *DIP-SLIP >0. FOR *X2>0. SIDE* DOWN*,/,
443 1 *FORMAT=U1IN,U1FN,U3IN,U3FN*)
444 RETURN
445 END
446 SUBROUTINE WRITE2(C)
447 DIMENSION C(40),A(30)
448 WRITE(7,3)
449 3 FORMAT(*SPECIFY INITIAL AND FINAL COORDINATES*,/,
450 1 *OF DIP-SLIP ZONE CORNERS AS INDICATED.*,/,
451 1 *D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,U3X1FN*)
452 CALL GETNUM(A) %C(1)=A(1) %C(2)=A(2) %C(4)=A(3) %C(5)=A(4)
453 C(9)=A(5) %C(10)=A(6) %C(12)=A(7) %C(13)=A(8)
454 C(3)=C(1) %C(7)=C(5) %C(6)=C(4) %C(8)=C(2)
455 C(11)=C(9) %C(16)=C(10) %C(14)=C(12) %C(15)=C(13)
456 RETURN
457 END
458 SUBROUTINE WRITE3(C)
459 REAL A(30)
460 DIMENSION C(40)
461 WRITE(7,8)
462 8 FORMAT(*SPECIFY INITIAL AND FINAL VALUES *,/,
463 1 *OF STRIKE-SLIP ZONE COORDINATES AS INDICATED*,/,
464 1 *C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN*)

```

```

465 CALL GETNUM(A) C(17)=A(1) C(18)=A(2) C(20)=A(3) C(21)=A(4)
466 C(25)=A(5) C(26)=A(6) C(28)=A(7) C(29)=A(8)
467 C(19)=C(17) C(22)=C(20) C(23)=C(21) C(24)=C(18)
468 C(27)=C(25) C(30)=C(28) C(31)=C(29) C(32)=C(26)
469 RETURN
470 END
471 SUBROUTINE WRITE4 (TRIGGR,M1,M2,M3,KCHECK)
472 REAL A(30)
473 M1=100 M2=100 M3=100
474 WRITE(7,150)
475 150 FORMAT(*DO YOU WISH THE *TRIGGER* OPTION?*,/,
476 1 *YES* OF *NO *)
477 READ(7,151)TRIGGR
478 151 FORMAT(A3)
479 KCHECK=0
480 IF(TRIGGR.EQ.3)GO TO 160
481 WRITE(7,157)
482 157 FORMAT(*SPECIFY *TRIGGER* PARAMETERS IN THE FORM*,/,
483 1 *D(M1).LT.C(M2)*
484 CALL GETNUM(A) M1=A(1) M2=A(2)
485 WRITE(7,158)
486 158 FORMAT(*SPECIFY WHICH SLIP COMPONENT IS TO BE =0.*,/,
487 1 *WHEN D(M1).GE.C(M2)*,/,
488 1 *STRIKE-SLIP=S, DIP-SLIP=D*)
489 READ(7,159)SLIP
490 159 FORMAT(A1)
491 IF(SLIP.EQ.1)M3=17
492 IF(SLIP.EQ.1)M3=18
493 160 CONTINUE
494 RETURN
495 END
496 SUBROUTINE GETNUM(F)
497 DIMENSION R(1),L(30)
498 READ(7,9)L R I=J=0
499 6 J=J+1 R N=P=S=0 M=F=1
500 5 I=I+1 R IF(I.GT.80)RETURN R D=L(I) R K=4
501 IF(D.EQ.36)K=2 R IF(D.GE.27.A.D.LF.36)K=1
502 IF(D.EQ.47)K=3 R K=K+S R GOTO(1,2,3,5,1,4,3,4)K
503 1 N=N*10+D-27 R S=4 R GOTO 5
504 2 M=-1 R S=4 R GOTO 5
505 3 P=I R S=4 R GOTO 5
506 4 IF(P.NE.0)F=10.** (I-P-1) R R(J)=N/F*M R GOTO 6
507 9 FOFMAT(80F1)
508 END

```

MAXVGT

Documentation for Program 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 approximately 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 component incremented exponentially
- 3 = both-slip components incremented 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 incremented 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 X PND/X PND01). 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.

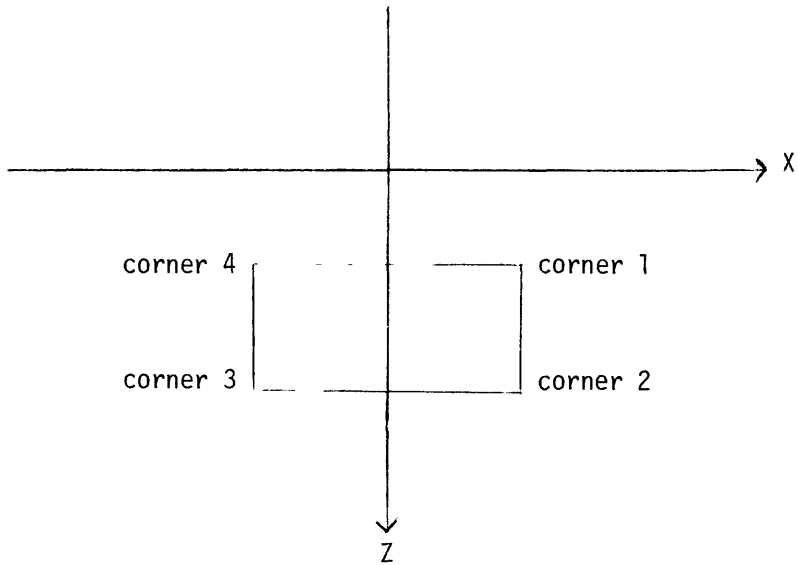
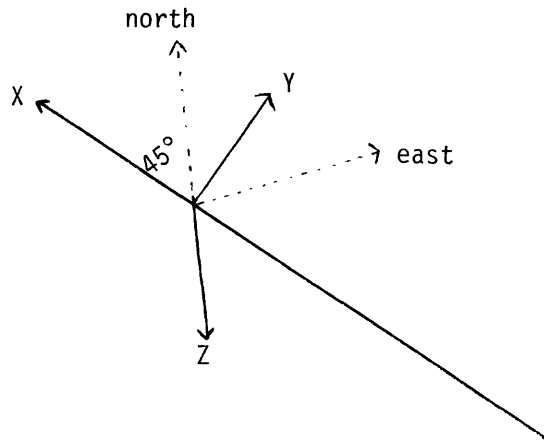


Figure 1

geometry and notation used in program MAXVGT

EXAMPLE FROM MAXVGT

```

1=ZONE EXPANSION
2=ZONE CONTRACTION
1
1=570E-01 COMPONENT REPRESENTED EXPONENTIALLY
2=0E+00 SLIP COMPONENTS REPRESENTED EXPONENTIALLY
3=0E+00 SLIP COMPONENTS REPRESENTED EXPONENTIALLY
4=NETEFFECT SLIP COMPONENT REPRESENTED EXPONENTIALLY
1
DO YOU WISH THE TRIGGER TO BE
YES (Y) NO (N)
1
1=0E+00 INITIAL AND FINAL SLIP VALUES
2=0E+00 LATERAL STRIKE-SLIP
3=0E+00 SLIP NO. FOR 182-90 STOOD DOWN
FORMAT=U1IN,U1FN,U2IN,U2FN
U 1 3 0 0
1
SPECIFY INITIAL AND FINAL VALUES
OF STRIKE-SLIP ZONE COORDINATES AS INDICATED
CENTIN,IX3IN,CZ3XIN,C3RTH,C3S1FN,C1X2FN,C2X3FN,C3X1FN
0 0 0 2 0 1 0
1
COORDINATES OF STATION (X1,X2)
1 37
1
ENTER MAXWELL AND UOIGT TIME CONSTANTS
.1 .1
1
1=STRIKE-SLIP BETWEEN STRIKE OF FAULT
AND NORTH: 45.000 DEGREES
U1I 1111: VALUES OF EM COMPONENT -4.992E-01 6.435E-02
U1N 1111: VALUES OF NS COMPONENT -4.146E-01 2.589E-01
U1I 1111: VALUES OF AMPLITUDE 0 5.016E-01
U1N 1111: VALUES OF AMPLITUDE 0 322.899
1
1=0E+00 TILT AMPLITUDES ARE IN MICRORADIANS
2=0E+00 TILT IN DEGREES
THE FOLLOWING ARE PLOTS OF THE EM AND NS
COMPONENTS OF TILT, AND THE TILT AMPLITUDE
(ALL AZIMUTH (MEASURED CLOCKWISE FROM NORTH).
U=0E+00 START, 1=CONTINUE
1
WRITE PLOT TITLE, 80 CHARACTERS
LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE!
SET HORIZONTAL SCALE: Y OR N (=BLANK)
1
SET VERTICAL SCALE: Y OR N (=BLANK)
Y
MIN/MAX Y VALUES
-1 1
1
SKIP PLOT OF EM TILT --- MAXWELL MODEL ?

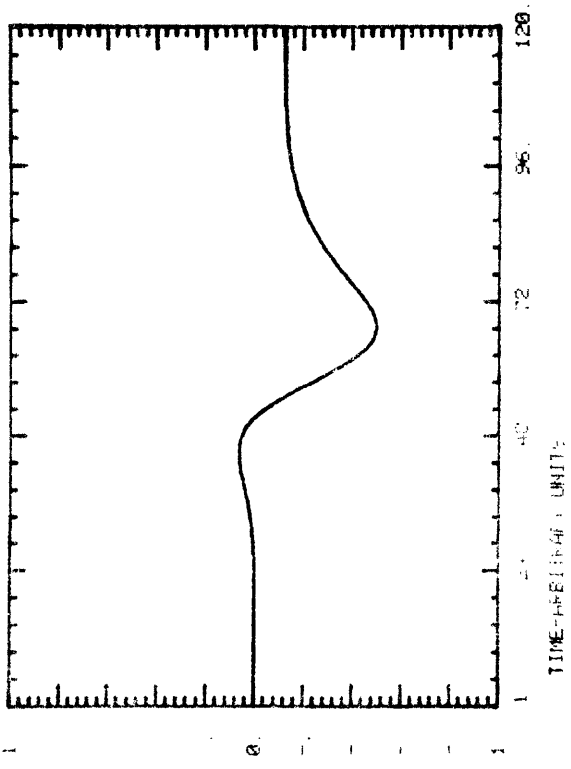
```

```

1=ZONE EXPANSION
2=ZONE CONTRACTION
1
1=570E-01 COMPONENT REPRESENTED EXPONENTIALLY
2=0E+00 SLIP COMPONENTS REPRESENTED EXPONENTIALLY
3=0E+00 SLIP COMPONENTS REPRESENTED EXPONENTIALLY
4=NETEFFECT SLIP COMPONENT REPRESENTED EXPONENTIALLY
1
DO YOU WISH THE TRIGGER TO BE
YES (Y) NO (N)
1
1=0E+00 INITIAL AND FINAL SLIP VALUES
2=0E+00 LATERAL STRIKE-SLIP
3=0E+00 SLIP NO. FOR 182-90 STOOD DOWN
FORMAT=U1IN,U1FN,U2IN,U2FN
U 1 3 0 0
1
SPECIFY INITIAL AND FINAL VALUES
OF STRIKE-SLIP ZONE COORDINATES AS INDICATED
CENTIN,IX3IN,CZ3XIN,C3RTH,C3S1FN,C1X2FN,C2X3FN,C3X1FN
0 0 0 2 0 1 0
1
COORDINATES OF STATION (X1,X2)
1 37
1
ENTER MAXWELL AND UOIGT TIME CONSTANTS
.1 .1
1

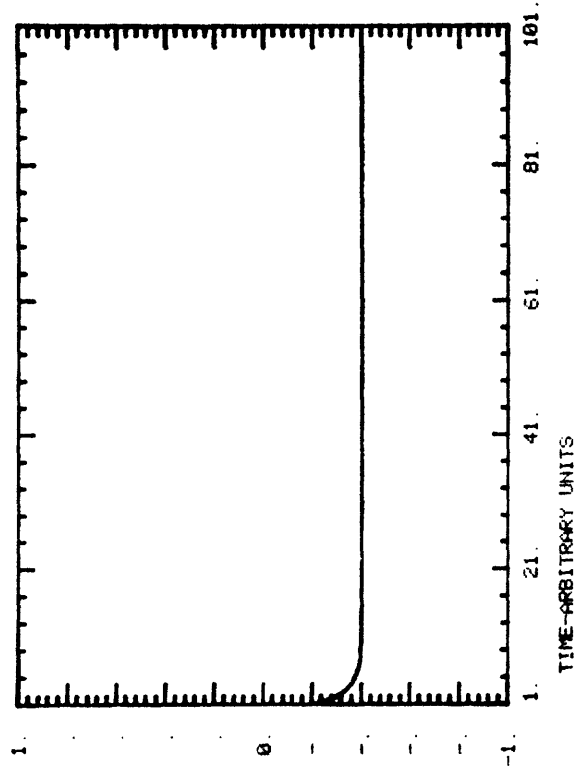
```

INITIAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
 MACQUELLE MODEL



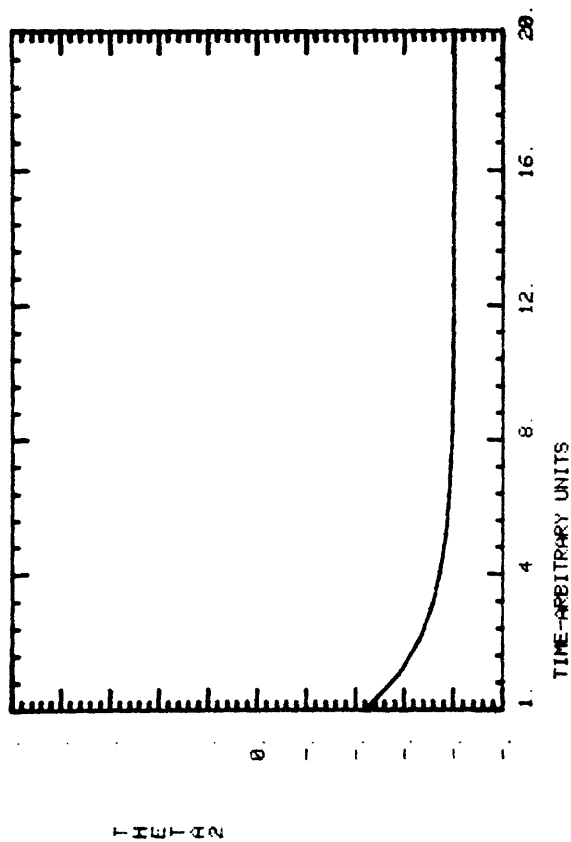
- 1) DATA PLOT, Z-RETURN
- 2) INITIAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
- 3) HORIZONTAL SCALES (FOR Z-RETURN)
- 4) VERTICAL SCALES (FOR HORIZONTAL)
- 5) INITIAL VALUE
- 6) TIME OF NO TILT --- MACQUELLE MODEL

LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
 IN A WELL MODEL



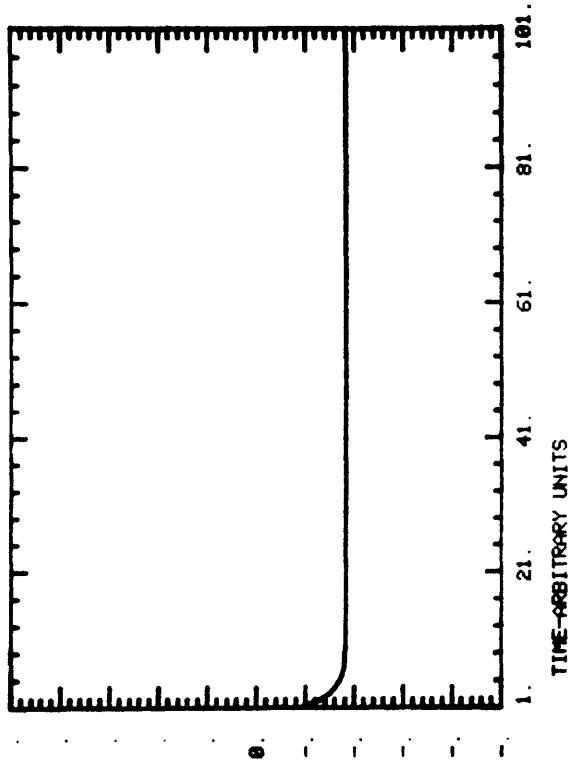
1 WHEN PLOT, 2=RETURN
 1.1
 1.2 SET HORIZONTAL SCALE? Y OR NK=BLANK)
 1.3
 1.4 MIN-MAX X VALUES
 1.5
 1.6 SET VERTICAL SCALE? Y OR NK=BLANK)
 1.7
 1.8 MIN-MAX Y VALUES
 1.9 -5 5
 2 SKIP PLOT OF THETA2 --- MAXWELL MODEL ?

LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
 MAXWELL MODEL



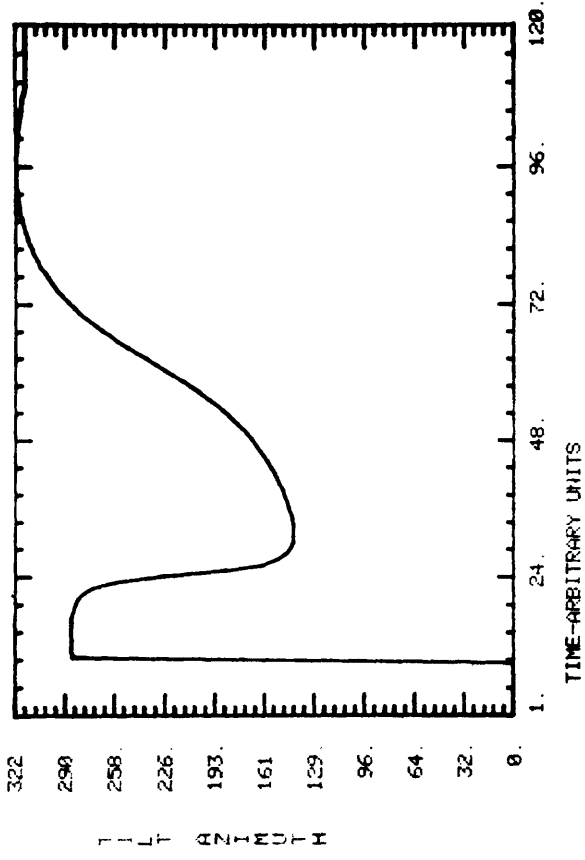
1) FROM PLOT, 2=RETURN
 2) HORIZONTAL SCALE? Y OR N:=BLANK
 3) VERTICAL SCALE? Y OR N:=BLANK
 4) Y VALUES
 5) PLOT OF THETA1 ----- MAXWELL MODEL

LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
MAXWELL MODEL

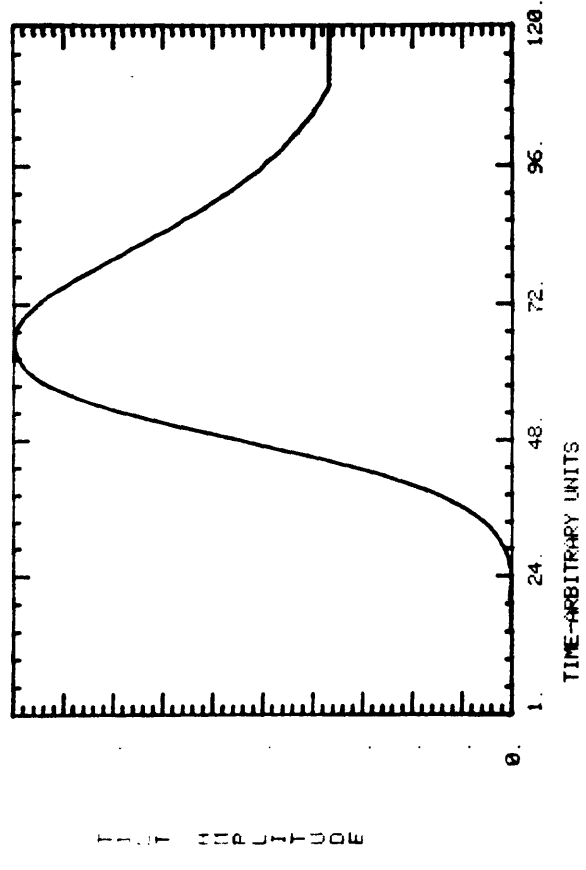


0:--START, 1--NEW PLOT, 2--RETURN
 2: ENTER ZONE INDEX, CAUTION--VALUE MUST BE
 BLANK EN 10 AND 110, 999=CONTINUE
 999: SLT HORIZONTAL SCALE? Y OR N (=BLANK)
 1: SET VERTICAL SCALE? Y OR N (=BLANK)
 1: SKIP PLOT OF TILT AMPLITUDE ---MAXWELL MODEL ?

LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
MULLER MODEL



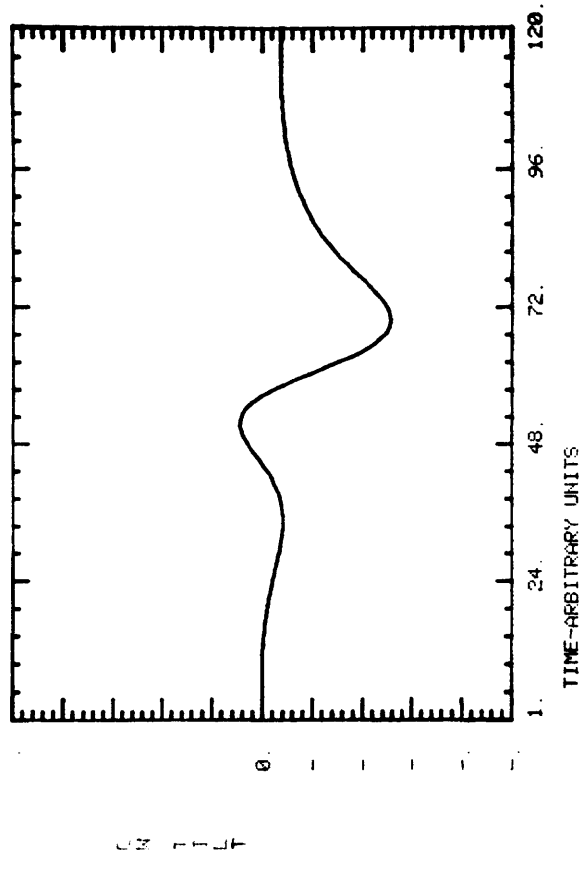
LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
MULLER MODEL



ALPHAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
 1.01 MODEL

```

0=DE START, 1=NEW PLOT, 2=RETURN
2.1
MIN MAX VALUES OF EW COMPONENT 2 555E-01 4 500E-02
MIN MAX VALUES OF NS COMPONENT 3 658E-01 8 534E-02
MIN MAX VALUES OF AMPLITUDE 0
MIN MAX VALUES OF AZIMUTH 0
MIN MAX VALUES OF ALTITUDE 325 434
CONT TILT AMPLITUDES ARE IN MICRORADIANS
AZIMUTH IN DEGREES
PL FOLLOWING ARE PLOT OF 04 EW AND NS
COMPONENTS OF TILT, AND TILT AMPLITUDE
AND AZIMUTH (MEASURED POSITIVE FROM NORTH).
PL START, 1=CONTINUE
1.1
WRITE PLOT TITLE, 80 CHARACTERS
ALPHAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE!
SET HORIZONTAL SCALE? Y OR N=(BLANK)
SET VERTICAL SCALE? Y OR N=(BLANK)
Y1
MIN/MAX Y VALUES
- 5 51
SKIP PLOT OF EW TILT --- VOIQT MODEL ?
  
```

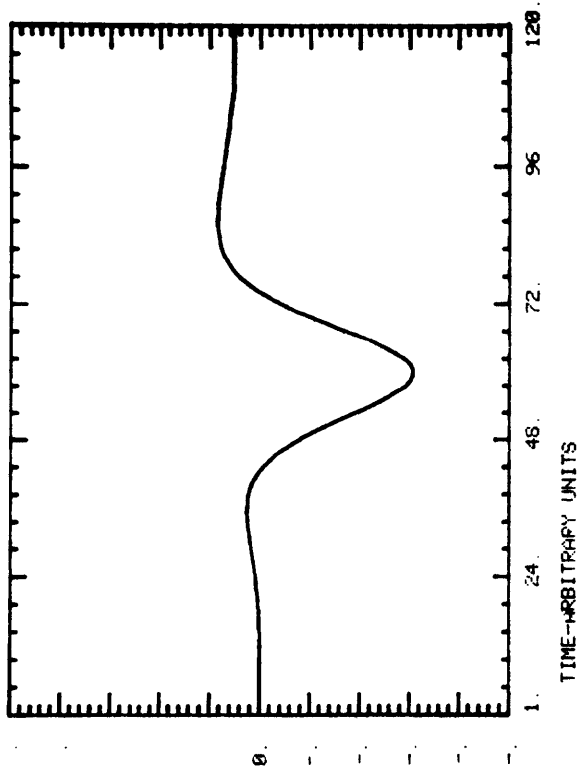


```

1 1=NEW PLOT, 2=RETURN
2 1
3 1 SET HORIZONTAL SCALE? Y OR N (=BLANK)
4 1
5 1 SET VERTICAL SCALE? Y OR N (=BLANK)
6 1
7 1 MIN/MAX Y VALUES
8 1 -5 51
9 1 SLIP PLOT OF NS TILT ---- VOIGT MODEL
10 1
11 1
12 1
13 1
14 1
15 1
16 1
17 1
18 1
19 1
20 1
21 1
22 1
23 1
24 1
25 1
26 1
27 1
28 1
29 1
30 1
31 1
32 1
33 1
34 1
35 1
36 1
37 1
38 1
39 1
40 1
41 1
42 1
43 1
44 1
45 1
46 1
47 1
48 1
49 1
50 1
51 1
52 1
53 1
54 1
55 1
56 1
57 1
58 1
59 1
60 1
61 1
62 1
63 1
64 1
65 1
66 1
67 1
68 1
69 1
70 1
71 1
72 1
73 1
74 1
75 1
76 1
77 1
78 1
79 1
80 1
81 1
82 1
83 1
84 1
85 1
86 1
87 1
88 1
89 1
90 1
91 1
92 1
93 1
94 1
95 1
96 1
97 1
98 1
99 1
100 1
101 1
102 1
103 1
104 1
105 1
106 1
107 1
108 1
109 1
110 1
111 1
112 1
113 1
114 1
115 1
116 1
117 1
118 1
119 1
120 1
121 1
122 1
123 1
124 1
125 1
126 1
127 1
128 1
129 1
130 1
131 1
132 1
133 1
134 1
135 1
136 1
137 1
138 1
139 1
140 1
141 1
142 1
143 1
144 1
145 1
146 1
147 1
148 1
149 1
150 1
151 1
152 1
153 1
154 1
155 1
156 1
157 1
158 1
159 1
160 1
161 1
162 1
163 1
164 1
165 1
166 1
167 1
168 1
169 1
170 1
171 1
172 1
173 1
174 1
175 1
176 1
177 1
178 1
179 1
180 1
181 1
182 1
183 1
184 1
185 1
186 1
187 1
188 1
189 1
190 1
191 1
192 1
193 1
194 1
195 1
196 1
197 1
198 1
199 1
200 1

```

LATERAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
VOIGT MODEL

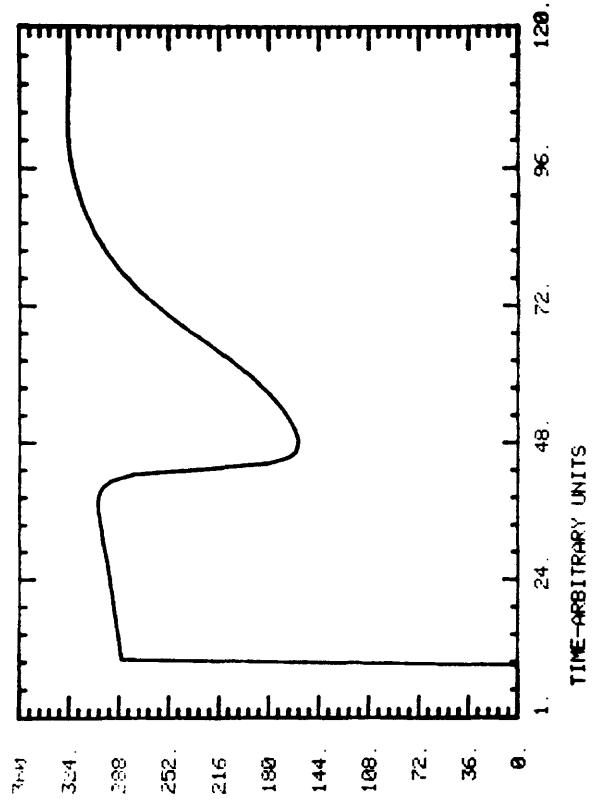


```

0=PC PLANT. 1=NEW PLOT. 2=RETURN
1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.
46.
47.
48.
49.
50.
51.
52.
53.
54.
55.
56.
57.
58.
59.
60.
61.
62.
63.
64.
65.
66.
67.
68.
69.
70.
71.
72.
73.
74.
75.
76.
77.
78.
79.
80.
81.
82.
83.
84.
85.
86.
87.
88.
89.
90.
91.
92.
93.
94.
95.
96.
97.
98.
99.
100.
101.
102.
103.
104.
105.
106.
107.
108.
109.
110.
111.
112.
113.
114.
115.
116.
117.
118.
119.
120.
121.
122.
123.
124.
125.
126.
127.
128.
129.
130.
131.
132.
133.
134.
135.
136.
137.
138.
139.
140.
141.
142.
143.
144.
145.
146.
147.
148.
149.
150.
151.
152.
153.
154.
155.
156.
157.
158.
159.
160.
161.
162.
163.
164.
165.
166.
167.
168.
169.
170.
171.
172.
173.
174.
175.
176.
177.
178.
179.
180.
181.
182.
183.
184.
185.
186.
187.
188.
189.
190.
191.
192.
193.
194.
195.
196.
197.
198.
199.
200.
201.
202.
203.
204.
205.
206.
207.
208.
209.
210.
211.
212.
213.
214.
215.
216.
217.
218.
219.
220.
221.
222.
223.
224.
225.
226.
227.
228.
229.
230.
231.
232.
233.
234.
235.
236.
237.
238.
239.
240.
241.
242.
243.
244.
245.
246.
247.
248.
249.
250.
251.
252.
253.
254.
255.
256.
257.
258.
259.
260.
261.
262.
263.
264.
265.
266.
267.
268.
269.
270.
271.
272.
273.
274.
275.
276.
277.
278.
279.
280.
281.
282.
283.
284.
285.
286.
287.
288.
289.
290.
291.
292.
293.
294.
295.
296.
297.
298.
299.
300.
301.
302.
303.
304.
305.
306.
307.
308.
309.
310.
311.
312.
313.
314.
315.
316.
317.
318.
319.
320.
321.
322.
323.
324.
325.
326.
327.
328.
329.
330.
331.
332.
333.
334.
335.
336.
337.
338.
339.
340.
341.
342.
343.
344.
345.
346.
347.
348.
349.
350.
351.
352.
353.
354.
355.
356.
357.
358.
359.
360.
361.
362.
363.
364.
365.
366.
367.
368.
369.
370.
371.
372.
373.
374.
375.
376.
377.
378.
379.
380.
381.
382.
383.
384.
385.
386.
387.
388.
389.
390.
391.
392.
393.
394.
395.
396.
397.
398.
399.
400.
401.
402.
403.
404.
405.
406.
407.
408.
409.
410.
411.
412.
413.
414.
415.
416.
417.
418.
419.
420.
421.
422.
423.
424.
425.
426.
427.
428.
429.
430.
431.
432.
433.
434.
435.
436.
437.
438.
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456.
457.
458.
459.
460.
461.
462.
463.
464.
465.
466.
467.
468.
469.
470.
471.
472.
473.
474.
475.
476.
477.
478.
479.
480.
481.
482.
483.
484.
485.
486.
487.
488.
489.
490.
491.
492.
493.
494.
495.
496.
497.
498.
499.
500.
501.
502.
503.
504.
505.
506.
507.
508.
509.
510.
511.
512.
513.
514.
515.
516.
517.
518.
519.
520.
521.
522.
523.
524.
525.
526.
527.
528.
529.
530.
531.
532.
533.
534.
535.
536.
537.
538.
539.
540.
541.
542.
543.
544.
545.
546.
547.
548.
549.
550.
551.
552.
553.
554.
555.
556.
557.
558.
559.
560.
561.
562.
563.
564.
565.
566.
567.
568.
569.
570.
571.
572.
573.
574.
575.
576.
577.
578.
579.
580.
581.
582.
583.
584.
585.
586.
587.
588.
589.
590.
591.
592.
593.
594.
595.
596.
597.
598.
599.
600.
601.
602.
603.
604.
605.
606.
607.
608.
609.
610.
611.
612.
613.
614.
615.
616.
617.
618.
619.
620.
621.
622.
623.
624.
625.
626.
627.
628.
629.
630.
631.
632.
633.
634.
635.
636.
637.
638.
639.
640.
641.
642.
643.
644.
645.
646.
647.
648.
649.
650.
651.
652.
653.
654.
655.
656.
657.
658.
659.
660.
661.
662.
663.
664.
665.
666.
667.
668.
669.
670.
671.
672.
673.
674.
675.
676.
677.
678.
679.
680.
681.
682.
683.
684.
685.
686.
687.
688.
689.
690.
691.
692.
693.
694.
695.
696.
697.
698.
699.
700.
701.
702.
703.
704.
705.
706.
707.
708.
709.
710.
711.
712.
713.
714.
715.
716.
717.
718.
719.
720.
721.
722.
723.
724.
725.
726.
727.
728.
729.
730.
731.
732.
733.
734.
735.
736.
737.
738.
739.
740.
741.
742.
743.
744.
745.
746.
747.
748.
749.
750.
751.
752.
753.
754.
755.
756.
757.
758.
759.
760.
761.
762.
763.
764.
765.
766.
767.
768.
769.
770.
771.
772.
773.
774.
775.
776.
777.
778.
779.
780.
781.
782.
783.
784.
785.
786.
787.
788.
789.
790.
791.
792.
793.
794.
795.
796.
797.
798.
799.
800.
801.
802.
803.
804.
805.
806.
807.
808.
809.
810.
811.
812.
813.
814.
815.
816.
817.
818.
819.
820.
821.
822.
823.
824.
825.
826.
827.
828.
829.
830.
831.
832.
833.
834.
835.
836.
837.
838.
839.
840.
841.
842.
843.
844.
845.
846.
847.
848.
849.
850.
851.
852.
853.
854.
855.
856.
857.
858.
859.
860.
861.
862.
863.
864.
865.
866.
867.
868.
869.
870.
871.
872.
873.
874.
875.
876.
877.
878.
879.
880.
881.
882.
883.
884.
885.
886.
887.
888.
889.
890.
891.
892.
893.
894.
895.
896.
897.
898.
899.
900.
901.
902.
903.
904.
905.
906.
907.
908.
909.
910.
911.
912.
913.
914.
915.
916.
917.
918.
919.
920.
921.
922.
923.
924.
925.
926.
927.
928.
929.
930.
931.
932.
933.
934.
935.
936.
937.
938.
939.
940.
941.
942.
943.
944.
945.
946.
947.
948.
949.
950.
951.
952.
953.
954.
955.
956.
957.
958.
959.
960.
961.
962.
963.
964.
965.
966.
967.
968.
969.
970.
971.
972.
973.
974.
975.
976.
977.
978.
979.
980.
981.
982.
983.
984.
985.
986.
987.
988.
989.
990.
991.
992.
993.
994.
995.
996.
997.
998.
999.
1000.

```

VERTICAL AND VERTICAL MIGRATION OF STRIKE-SLIP ZONE
 UOIGT MODEL



B25

B26

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

7)

OK - ^EDIT

LISTING OF MAXVGT

```

1  DELETE(LGO,OUTPUT,MAXVGT)
2  MAXVGT.
3  CXIT.
4  LIBCOPY(GRAPHIC, TXLGO/PR, TXLGO)
5  LIBCOPY(JDRAT, NPLGO/RR, NPLGO)
6  DELETE(LGO,CUTPUT,MAXVGT)
7  RUN76(S)
8  LINK(F=LGO,F=TXLGO,F=NPLGO,B=MAXVGT)
9  MAXVGT.
10 FIN.
11 FOR
12     PROGRAM MAXVGT(TAPE7=201,FILM=TAPETTY,TAPE7=TAPE1TY)
13     COMMON/TVPCOL/TVPUL(8)
14     COMMON/TVTUNE/ITUNE(30)
15     COMMON/JFLT(XLT,XFT,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 D(20),C(40),A(30)
20     DIMENSION THETA1(120,101),THETA2(120,101),T(101)
21     DATA KTIME,FTIME,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,IFCT,-8)
26     20 CONTINUE
27     MSTART=3 $WRITE(7,12)
28     12 FORMAT(*1=ZONE EXPANS*,/,
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) $IFLAG=A(1)
36     CALL WRITE4(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) $THETA=45.
50     1234 FORMAT(*ENTER MAXWELL AND VOIGT TIME CONSTANTS*)
51     CALL GETNUM(A)$D(19)=A(1)$L(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 $IJKL=(.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) $GOTO 11
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.EG.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.EG.4)
123     1 D(M3)=(((C(34)-C(33))/(KTIME))*(I-RTIME))+C(33)
124     156 CALL CMFTLT(D,X1,X2,X3,COUNTN,I,INDEX,THETA1,THETA2)
125     T(I)=I
126     103 CCNTINUF
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     TF(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*.61745)
163     ST=SIN(THETA*.61745)
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(I)+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) $STAMPMD=TAMP(1) $TAZMPD=TAZM(1)

```



```

175      TEWMIN=TEWMAX=TEWMIC $TNSMIN=TNSMAX=TNSMID
176      TAMPMN=TAMPX=TAMPMD $TAZMMN=TAZMMX=TAZMMD
177      DO 160 I=1,NUM
178      IF (TEW(I) .LT. TEWMIN)TEWMIN=TEW(I)
179      IF (TEW(I) .GT. TEWMAX)TEWMAX=TEW(I)
180      IF (TNS(I) .LT. TNSMIN)TNSMIN=TNS(I)
181      IF (TNS(I) .GT. TNSMAX)TNSMAX=TNS(I)
182      IF (TAMP(I) .LT. TAMPMN)TAMPMN=TAMP(I)
183      IF (TAMP(I) .GT. TAMPX)TAMPX=TAMP(I)
184      IF (TAZM(I) .LT. TAZMMN)TAZMMN=TAZM(I)
185      IF (TAZM(I) .GT. TAZMMX)TAZMMX=TAZM(I)
186      160 CONTINUE
187      WRITE(7,17)TEWMIN,TEWMAX,TNSMIN,TNSMAX,TAMPMN,TAMPX,TAZMMN,TAZMMX
138      17 FORMAT(*MIN./MAX. VALUES OF EW COMPONENT*,2X,E10.3,2X,E10.3/,
189      1 *MIN./MAX. VALUES OF NS COMPONENT*,2X,E10.3,2X,E10.3/,
190      1 *MIN./MAX. VALUES OF AMPLITUDE*,5X,E10.3,2X,E10.3/,
191      1 *MIN./MAX. VALUES OF AZIMUTH*,3X,F10.3,2X,F10.3/,
192      1 *(NOTE TILT AMPLITUDES ARE IN MICRORADIANS*,/,
193      1 *AZIMUTH IN DEGREES*)
194      WRITE(7,170)
195      170 FORMAT(*THE FOLLOWING ARE PLOTS OF THE EW AND NS*,/,
196      1 *COMPONENTS OF TILT, AND THE TILT AMPLITUDE*,/,
197      1 *AND AZIMUTH (MEASURED CLOCKWISE FROM NORTH).*,/,
198      1 *0=RE-START, 1=CONTINUE*)
199      CALL GETNUM(A) $IKFLAG=A(1)
200      IF(IKFLAG.EQ. 0) GO TO 113
201      LU=7 $LNLGX=1 $LNLY=1 $NCLX=2 $NCLY=2
202      WRITE(7,199)
203      199 FORMAT(*WRITE PLOT TITLE, 80 CHARACTERS*)
204      READ(7,200)(LTITL(JM),JM=1,8)
205      200 FORMAT(8A10)
206      202 IF(COUNTF.EQ.1)LTITL2(1)=10HMAXWELL MC
207      IF(COUNTR.EQ.1)LTITL2(2)=10HDEL
208      IF(COUNTF.EQ.2)LTITL2(1)=10HVOIGT MODE
209      IF(COUNTR.EQ.2)LTITL2(2)=10HCL
210      DO 8 MM=3,8
211      8 LTITL2(MM)=10H
212      KX(1)=10HTIME-ARBIT
213      KX(2)=10HFARY UNITS
214      KY(1)=10HEW TILT
215      KY(2)=10H          $MAJX=5 $MAJY=10 $XLT=T(1)
216      XRT=T(NUM) $YLO=TEWMIN $YUP=TEWMAX
217      WRITE(7,70)
218      70 FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
219      READ(7,66)CHARAC $IF(CHARAC.EQ.1)N.OR.CHARAC.EQ.1H)GO TO 71
220      WRITE(7,72)
221      72 FORMAT(*MIN/MAX X VALUES*)
222      CALL GETNUM(A) $XLT=A(1) $XRT=A(2)
223      71 WRITE(7,73)
224      73 FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
225      READ(7,66)CHAFAC $IF(CHARAC.EQ.1)N.OR.CHARAC.EQ.1H)GC TO 74
226      WRITE(7,75)
227      75 FORMAT(*MIN/MAX Y VALUES*)
228      CALL GETNUM(A) $YLO=A(1) $YUP=A(2)
229      74 AAA=YUP $IF(YLO.EQ.AAA)YUP=YUP+1.$IF(YLO.EQ.AAA)YLC=YLO-1.
230      WRITE(7,1)KY(1),LTITL2(1),LTITL2(2)
231      1 FORMAT(*SKIP PLOT OF *,A10,*--- *,2A10,*?*)
232      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(IPS.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)YLC=YLO-1.
247     WRITE(7,1)KY(1),LTITL2(1),LTITL2(2) $READ(7,66)IJVAR
248     IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TNS,T,1,NUM)
249     WRITE(7,19)
250     19  FORMAT(*0=FE-START, 1=NEW PLOT, 2=RETURN*)
251     CALL GETNUM(A)$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(1,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)YLC=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)=10H THETA1      $XLT=TT(1)$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)YLC=YLO-1.
290     WRITE(7,1)KY(1),LTITL2(1),LTITL2(2) $READ(7,66)IJVAR

```

```

291     IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TNS,TT,1,INDEX)
292     WRITE(7,19)$CALL GETNUM(A)$IF(A(1).EQ.0.)GOTO113
293     IF(A(1).EQ.1.)GOTO217
294     GOTO32
295     31 CONTINUE
296     218 KY(1)=10HTILT AMPLI   $KY(2)=10HTUDE
297     YLC=TAMP MN $YUP=TAMP MX $XLT=T(1) $XRT=T(NUM)
298     WRITE(7,76) $FEAD(7,66)CHARAC
299     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC211
300     WRITE(7,72)$CALL GETNUM(A)$XLT=A(1)$XRT=A(2)
301     211 WRITE(7,73)$READ(7,66)CHARAC
302     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO212
303     WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
304     212 AAA=YUP$IF(YLO.EQ.AAA)YUP=YUP+1.$IF(YLO.EQ.AAA)YLC=YLO-1.
305     WRITE(7,213)KY(1),KY(2),LTITL2(1),LTITL2(2)$READ(7,66)IJVAR
306     213 FORMAT(*SKIP PLOT OF *,2A10,*---*,2A10,*?*)
307     IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAMP,T,1,NUM)
308     WRITE(7,201)$CALL GETNUM(A)$IF(A(1).EQ.1.)GOTO218
309     220 KY(1)=10HTILT AZIMB   $KY(2)=10HTH           $YLO=TAZMMN
310     YUP=TAZMMX $XLT=T(1) $XRT=T(NUM)
311     WRITE(7,70)$READ(7,66)CHARAC
312     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO214
313     WRITE(7,72)$CALL GETNUM(A)$XLT=A(1)$XRT=A(2)
314     214 WRITE(7,73)$READ(7,66)CHARAC
315     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC215
316     WRITE(7,75)$CALL GETNUM(A)$YLO=A(1)$YUP=A(2)
317     215 AAA=YUP$IF(YLO.EQ.AAA)YUP=YUP+1.$IF(YLO.EQ.AAA)YLC=YLO-1.
318     WRITE(7,213)KY(1),KY(2),LTITL2(1),LTITL2(2)$READ(7,66)IJVAR
319     IF(IJVAR.EQ.1HN.OR.IJVAR.EQ.1H )CALL PLOTS(TAZM,T,1,NUM)
320     WRITE(7,19)$CALL GETNUM(A)
321     IF(A(1).EQ.0.)GOTO113$IF(A(1).EQ.1.)GOTO220
322     113 COUNTR=CCUNTR+1
323     IF(COUNTR.EQ.2)GOTO22
324     WRITE(7,115)
325     115 FORMAT(*1=RE-START WITH ALL NEW VALUES*,/,
326     1 *2=RE-START WITH NEW STRIKE-SLIP VALUE AND ZONE COORDINATES*,/,
327     1 *3=RE-START WITH NEW DIP-SLIP VALUE AND ZONE COORDINATES*,/,
328     1 *4=RE-START WITH NEW STRIKE-SLIP VALUE ONLY*,/,
329     1 *5=RE-START WITH NEW DIP-SLIP VALUE ONLY*,/,
330     1 *6=RE-START WITH NEW TILTMETER COORDINATES ONLY*,/,
331     1 *7=STOP*)
332     CALL GETNUM(A) $MSTART=A(1)
333     IF(MSTART.EQ.2)GOTO116 $IF(MSTART.EQ.3)GOTO117
334     IF(MSTART.EQ.4)GOTO118 $IF(MSTART.EQ.5)GOTC119
335     IF(MSTART.EQ.6)GOTO120 $IF(MSTART.EQ.7)GOTO630
336     GO TO 20
337     116 WRITE(7,12)
338     CALL GETNUM(A) $IFLAG=A(1) $WRITE(7,15)
339     CALL GETNUM(A) $LFLAG=A(1)
340     WRITE(7,23)
341     23  FORMAT(*U1IN,U1FN*)
342     CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
343     CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
344     GO TO 100
345     117 WRITE(7,12)
346     CALL GETNUM(A) $IFLAG=A(1)
347     WRITE(7,15)
348     CALL GETNUM(A) $LFLAG=A(1)

```

```

349      WRITE(7,25)
350      25  FORMAT(*U3IN,U3FN*)
351      CALL GETNUM(A) $C(35)=A(1) $C(36)=A(2)
352      CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
353      GO TO 122
354      118 WRITE(7,12)
355      CALL GETNUM(A) $IFLAG=A(1)
356      WRITE(7,15)
357      CALL GETNUM(A) $LFLAG=A(1)
358      WRITE(7,23)
359      CALL GETNUM(A) $C(33)=A(1) $C(34)=A(2)
360      CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
361      GO TO 123
362      119 WRITE(7,12)
363      CALL GETNUM(A) $IFLAG=A(1)
364      WRITE(7,15)
365      CALL GETNUM(A) $LFLAG=A(1)
366      WRITE(7,25)
367      CALL GETNUM(A) $C(35)=A(1) $C(36)=A(2)
368      CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
369      GO TO 123
370      120 WRITE(7,4)
371      CALL GETNUM(A) $X1=A(1) $X2=A(2)
372      CALL WRITE4(TRIGGR,M1,M2,M3,KCHECK)
373      GO TO 123
374      630 STOP
375      END
376      SUBROUTINE CMPTLT(D,X1,X2,X3,COUNT,I,INDEX,THETA1,THETA2)
377      DIMENSION D(20),THETA1(120,101),THETA2(120,101)
378      TAUM=D(19) $TALV=D(20)
379      IF(COUNT.EQ.2)GOTO3
380      DO 4 K=1,INDEX
381      TIME=((K-1.)/(INDEX/6.))/TAUM
382      A1=A2=A3=A4=B1=B2=B3=B4=0.
383      DA1=DA2=DA3=DA4=DB1=DB2=DB3=DB4=0.
384      U1=D(17) $U3=D(18)
385      IF(U1.EQ.0.)GOTO1
386      CALL MAXTLT(TIME,D(17),X1,X2,X3,D(11),D(12),A1,B1)
387      CALL MAXTLT(TIME,D(17),X1,X2,X3,D(9),D(10),A2,B2)
388      CALL MAXTLT(TIME,D(17),X1,X2,X3,D(13),D(14),A3,B3)
389      CALL MAXTLT(TIME,D(17),X1,X2,X3,D(15),D(16),A4,B4)
390      1  IF(U3.EQ.0.)GO TO 2
391      CALL MAXDPS(TIME,D(18),X1,X2,X3,D(3),D(4),CA1,CB1)
392      CALL MAXDPS(TIME,D(18),X1,X2,X3,D(1),D(2),CA2,CB2)
393      CALL MAXDPS(TIME,D(18),X1,X2,X3,D(5),D(6),CA3,CB3)
394      CALL MAXDPS(TIME,D(18),X1,X2,X3,D(7),D(8),CA4,CB4)
395      2  THETA1(I,K)=A1-A2-A3+A4+DA1-DA2-DA3+DA4
396      4  THETA2(I,K)=B1-B2-B3+B4+DB1-DB2-DB3+DB4
397      GOT05
398      3  DO 6 K=1,INDEX
399      TIME=((K-1.)/(INDEX/6.))/TALV
400      A1=A2=A3=A4=B1=B2=B3=B4=0.
401      DA1=DA2=DA3=DA4=DB1=DB2=DB3=DB4=0.
402      U1=D(17) $U3=D(18)
403      IF(U1.EQ.0.)GOTO7
404      CALL VGTTLT(TIME,D(17),X1,X2,X3,D(11),D(12),A1,B1)
405      CALL VGTTLT(TIME,D(17),X1,X2,X3,D(9),D(10),A2,B2)
406      CALL VGTTLT(TIME,D(17),X1,X2,X3,D(13),D(14),A3,B3)

```

```

407 CALL VGTTLT(TIME,D(17),X1,X2,X3,D(15),D(16),A4,B4)
408 7 IF(U3,FQ.0.)GOTO8
409 CALL VGTCPS(TIME,D(18),X1,X2,X3,D(3),D(4),CA1,DB1)
410 CALL VGTCPS(TIME,D(18),X1,X2,X3,D(1),D(2),CA2,DB2)
411 CALL VGTCPS(TIME,D(18),X1,X2,X3,D(5),D(6),CA3,DB3)
412 CALL VGTCPS(TIME,D(18),X1,X2,X3,D(7),D(8),DA4,DB4)
413 8 THETA1(I,K)=A1-A2-A3+A4+DA1-DA2-DA3+DA4
414 6 THETA2(I,K)=B1-B2-B3+B4+DB1-DB2-DB3+DB4
415 5 RETURN
416 END
417 SUBROUTINE VGTTLT(TIME,U1,X1,X2,X3,P1,P3,T1,T2)
418 PI=3.14159 $CHI=X2/SQRT(2.)
419 R=SQRT(((X1-P1)**2)+(X2**2)+(P3**2))
420 F1=(R**2)*((2.*(CHI**2))-(R*(R+P3)))
421 F5=(2.*P3)*(((X2**2)*((2.*R)+P3))-((R**2)*(R+P3)))
422 T1=(-U1/(4.*PI))*(((1.-5.*EXP(-6.*TIME))*(R**2))+((2.*P3)
423 1 *((2.*R)+P3))*(X2*(P1-X1)/((R**3)*((R+P3)**2))))
424 T2=(U1/(4.*PI))*(((1.-5.*EXP(-6.*TIME))*F1)+F5)*(1./((R**3)*((
425 1 R+P3)**2))))
426 RETURN
427 END
428 SUBROUTINE MAXTLT(TIME,U1,X1,X2,X3,P1,P3,T1,T2)
429 PI=3.14159 $CHI=X2/SQRT(2.)
430 R=SQRT(((X1-P1)**2)+(X2**2)+(P3**2))
431 F1=(R**2)*((2.*(CHI**2))-(R*(R+P3)))
432 F5=(2.*P3)*(((X2**2)*((2.*R)+P3))-((R**2)*(R+P3)))
433 T1=(-U1/(4.*PI))*(((2.-EXP((-5./6.)*TIME))*(R**2))+((2.*P3)*
434 1 ((2.*R)+P3)))*((X2*(P1-X1))/((R**3)*((R+P3)**2))))
435 T2=(U1/(4.*PI))*(((2.-EXP((-5./6.)*TIME))*F1)+F5)*(1./((R**3)
436 1 *((R+P3)**2))))
437 RETURN
438 END
439 SUBROUTINE WRITE1
440 WRITE(7,1)
441 1 FORMAT(*SPECIFY INITIAL AND FINAL SLIP VALUES*,/,
442 1 *RIGHT-LATERAL STRIKE-SLIP IS <0.*,/,
443 1 *DIP-SLIP >0. FOR *X2>0. SIDE* DOWN*,/,
444 1 *FORMAT=U1IN,U1FN,U3IN,U3FN*)
445 RETURN
446 END
447 SUBROUTINE WRITE2(C)
448 DIMENSION C(40),A(30)
449 WRITE(7,3)
450 3 FORMAT(*SPECIFY INITIAL AND FINAL COORDINATES*,/,
451 1 *OF DIP-SLIP ZONE CORNERS AS INDICATED.*,/,
452 1 *D1X1IN,D1X3IN,D2X3IN,D3X1IN,D1X1FN,D1X3FN,D2X3FN,D3X1FN*)
453 CALL GETNUM(A) $C(1)=A(1) $C(2)=A(2) $C(4)=A(3) $C(5)=A(4)
454 C(9)=A(5) $C(10)=A(6) $C(12)=A(7) $C(13)=A(8)
455 C(3)=C(1) $C(7)=C(5) $C(6)=C(4) $C(8)=C(2)
456 C(11)=C(9) $C(16)=C(10) $C(14)=C(12) $C(15)=C(13)
457 RETURN
458 END
459 SUBROUTINE WRITE3(C)
460 REAL A(30)
461 DIMENSION C(40)
462 WRITE(7,8)
463 8 FORMAT(*SPECIFY INITIAL AND FINAL VALUES *,/,
464 1 *OF STRIKE-SLIP ZONE COORDINATES AS INDICATED*,/,

```

```

465     1 *C1X1IN,C1X3IN,C2X3IN,C3X1IN,C1X1FN,C1X3FN,C2X3FN,C3X1FN*)
466     CALL GETNUM(A) $C(17)=A(1) $C(18)=A(2) $C(20)=A(3) $C(21)=A(4)
467     C(25)=A(5) $C(26)=A(6) $C(28)=A(7) $C(29)=A(8)
468     C(19)=C(17) $C(22)=C(20) $C(23)=C(21) $C(24)=C(18)
469     C(27)=C(25) $C(30)=C(28) $C(31)=C(29) $C(32)=C(26)
470     RETURN
471     END
472     SUBROUTINE WRITE4(TRIGGR,M1,M2,M3,KCHECK)
473     REAL A(30)
474     M1=100 $M2=100 $M3=100
475     WRITE(7,150)
476     150 FORMAT(*DO YOU WISH THE *TRIGGER* OPTION?*,/,
477     1 **YES* OR *NO ***)
478     READ(7,151)TRIGGR
479     151 FORMAT(A3)
480     KCHECK=0
481     IF(TRIGGR.EQ.3HNO)GO TO 160
482     WRITE(7,157)
483     157 FORMAT(*SPECIFY *TRIGGER* PARAMETERS IN THE FORM*,/,
484     1 *D(M1).GF.C(M2)*)
485     CALL GETNUM(A) $M1=A(1) $M2=A(2)
486     WRITE(7,158)
487     158 FORMAT(*SPECIFY WHICH SLIP COMPONENT IS TO BE =0.*/,
488     1 *WHEN D(M1).LT.C(M2)*,/,
489     1 *STRIKE-SLIP=S, DIP-SLIP=C*)
490     READ(7,159)SLIP
491     159 FORMAT(A1)
492     IF(SLIP.EQ.1HS)M3=17
493     IF(SLIP.EQ.1HD)M3=18
494     160 CONTINUE
495     RETURN
496     FNC
497     SUBROUTINE GETNUM(P)
498     DIMENSION F(1),L(80)
499     READ(7,9)L $ I=J=0
500     6 J=J+1 $ N=P=S=0 $ N=F=1
501     5 I=I+1 $ IF(I.GT.80)RETURN $ D=L(I) $ K=4
502     IF(D.FQ.38)K=2 $ IF(D.GE.27.A.D.LE.30)K=1
503     IF(D.EQ.47)K=3 $ K=K+S $ GOTO(1,2,3,5,1,4,3,4)K
504     1 N=N*10+D-27 $ S=4 $ GOTO 5
505     2 M=-1 $ S=4 $ GOTO 5
506     3 P=I $ S=4 $ GOTO 5
507     4 IF(P.NE.0)F=10.** (I-P-1) $ F(J)=N/F*M $ GOTO 6
508     9 FORMAT(80R1)
509     END
510     SUBROUTINE MAXDPS(TIME,R1,X1,X2,X3,R2,R3,T1,T2)
511     ALPHA=TIME*R1*X1*X2*X3*R2*R3
512     T1=ALPHA
513     T1=T2=0.
514     RETURN
515     END
516     SUBROUTINE VGTDPS(TIME,R1,X1,X2,X3,R2,R3,T1,T2)
517     ALPHA=TIME*R1*X1*X2*X3*R2*R3
518     T1=T2=0.
519     RETURN
520     FND

```

ST3PHS, ST3PSZ

STNRML, SHRSTN

DBLNRM

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 $D < A < 0$.) 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 ($D > 0$.)

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 $y > 0$.)

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 at the free surface and/or at 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:

$$\frac{D(\text{final}) - D(\text{initial})}{2.7} + D(\text{initial})$$

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

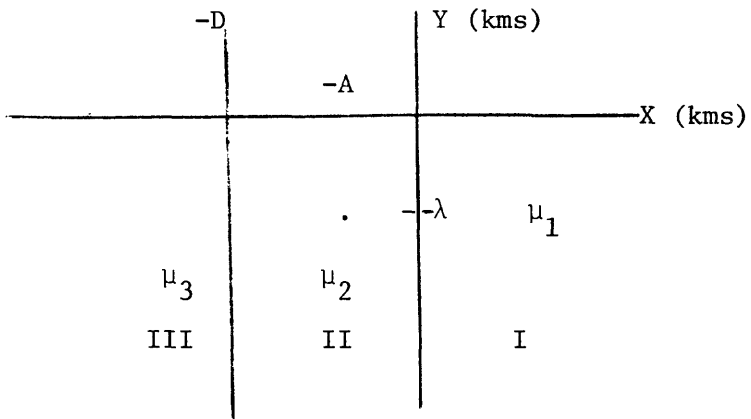


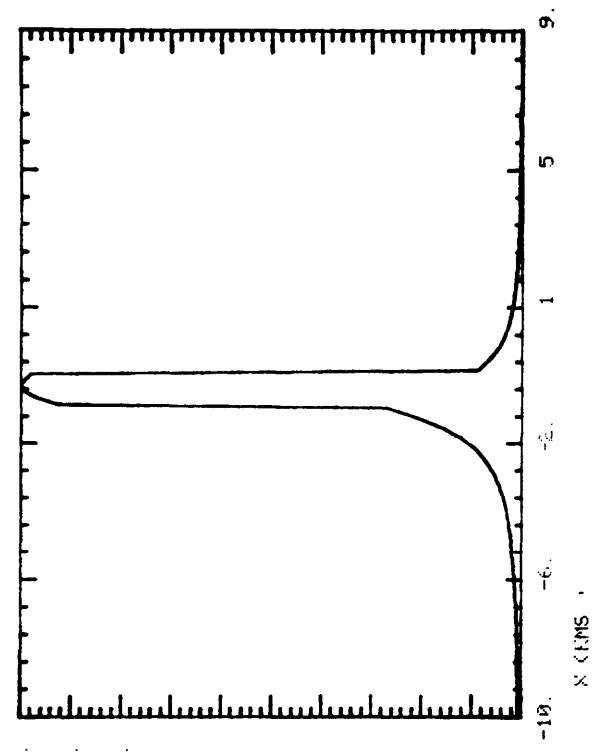
FIGURE 1

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

EXAMPLE FROM ST3PHS

C12

ST391P: EXAMPLE
XZ STRAIN



0 1 2 3 4 5 6 7 8 9 10

1 0=RE-START, 1=NEW PLOT
21
MIN:MAX VALUES OF XZ MICROSTRAIN
5.6515E-04 3.1000E-01
SET VERTICAL SCALE: 7 OR 10=BLANK)
1
SET HORIZONTAL SCALE: Y OR N=BLANK)
1
CLIP=

C15

C16

```

10=PE--START, 1=NEW PLOT
2,
MIN: MAX VALUES OF (KX) US: X 3 5560E-01
-5 5602E-01
SET VERTICAL SCALE? Y (OR N=BLANK)
1
SET HORIZONTAL SCALE? Y (OR N=BLANK)
1
SKIP?

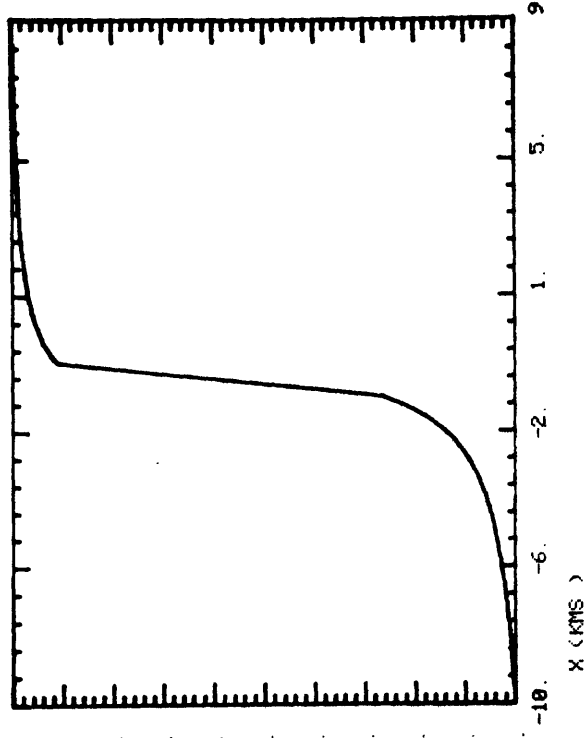
```

```

STEPHS EXAMPLE
(MIN) US: X

```

DISPLACEMENT - INCHES

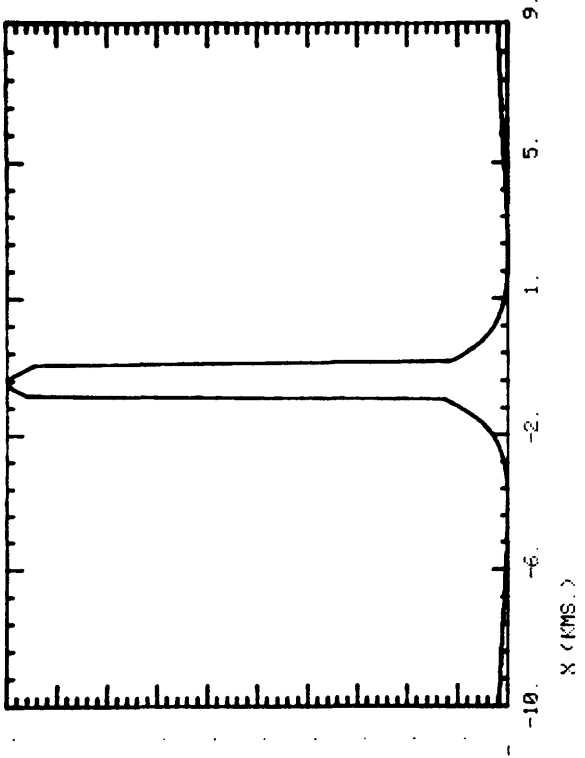


1
0=RE-START, 1=NEW PLOT
2!
MU1 = 3.0000E+11 MU2 = 3.0000E+10
MU3 = 1.0000E+11 M = 5.3846E-01
K = 8.1818E-01
DISLOCATION POSITION = - 500, - 1.000 KMS.
SLAB THICKNESS = 1.000 KMS.
BURGERS VECTOR = -1.000 MMS.
NO STRESS/STRAIN US. X
0=RE-START
1=STOP
2=RE-START WITH SAME VALUES, DIFFERENT PROFILE
!!
OK - ^EDIT

EXAMPLE FROM ST3PSZ

c19a

ST3P5Z (SAMPLE WIDTH=5
 XZ (INCH)



MULTIPLIER

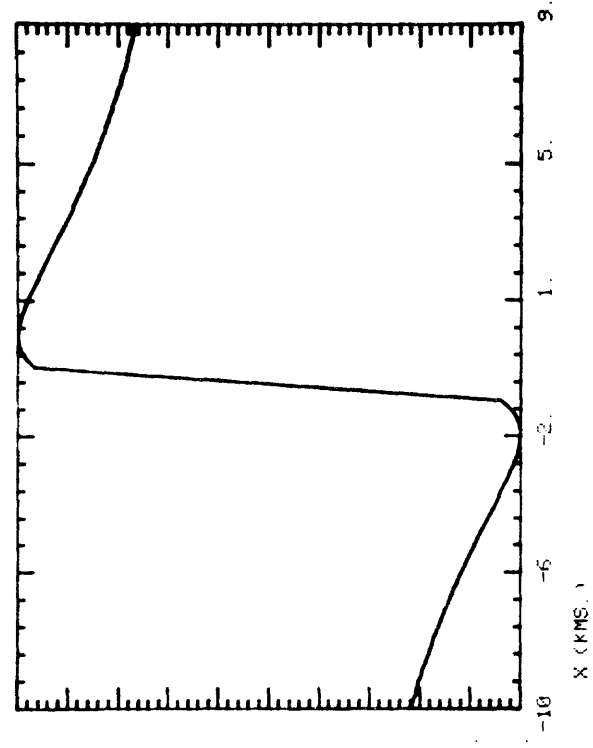
- 1. START WHEN PLOT
- 2. MULT VALUES OF MICROPHONIC
- 3. VOLTAGE=33
- 4. VERTICAL SCALE BY 1000000
- 5. HORIZONTAL SCALE BY 1000000
- 6. END

```

1  NAME=START, I=NEW PLOT
2  TITLE MAX VALUES OF W(X) US
   -2  DAPPE=01
3  VERTICAL SCALE? Y OR N=BLANK
1  HORIZONTAL SCALE? Y OR N=BLANK
1  SLIP?

```

STARTING EXAMPLE PLOTS
W(X)



```

1
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
MULTIPLICATION POSITION = .500 = 1.000 KMS.
WIDTH OF SLIP ZONE = 5.000
SLIP THICKNESS = 1.000 KMS.
BURGER VECTOR = -1.000 KMS.
% STRESS-STRAIN US. %
0=RE-START
1=STOP
2=RE-START WITH SAME VALUES, DIFFERENT PROFILE
11
UK = ^EDIT

```

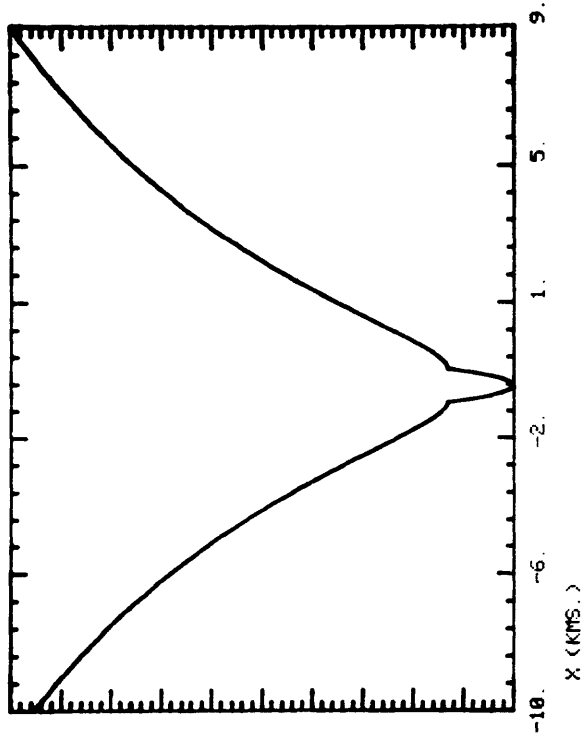
EXAMPLE FROM STNRML

```

SHEAR MODULI--IN CGS UNITS--MU1,MU2,MU3
3E11 3.1E11
SHEAR MODULUS OF HOMOGENEOUS HALF-SPACE
3E11
POSITION OF DISLOCATION--IN KMS , (X,Z) (M,OA)
NOTE --H=9 , L=0E+0
.5
1
THICKNESS OF SLAB, IN KMS --0
1
BURGER'S VECTOR, IN KMS --B
BCO FOR PITCH LATERAL SLIP
1
1=AMPLITUDE VS. X, Z=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
2.321E-01 6.0766E-01
PLOT TITLE, 80 CHARACTERS MAXIMUM
STNRM/EXAMPLE1
GET VERTICAL SCALE? Y OR N(=BLANK)
1
SET HORIZONTAL SCALE? Y OR N(=BLANK)
1
SKIP?

```

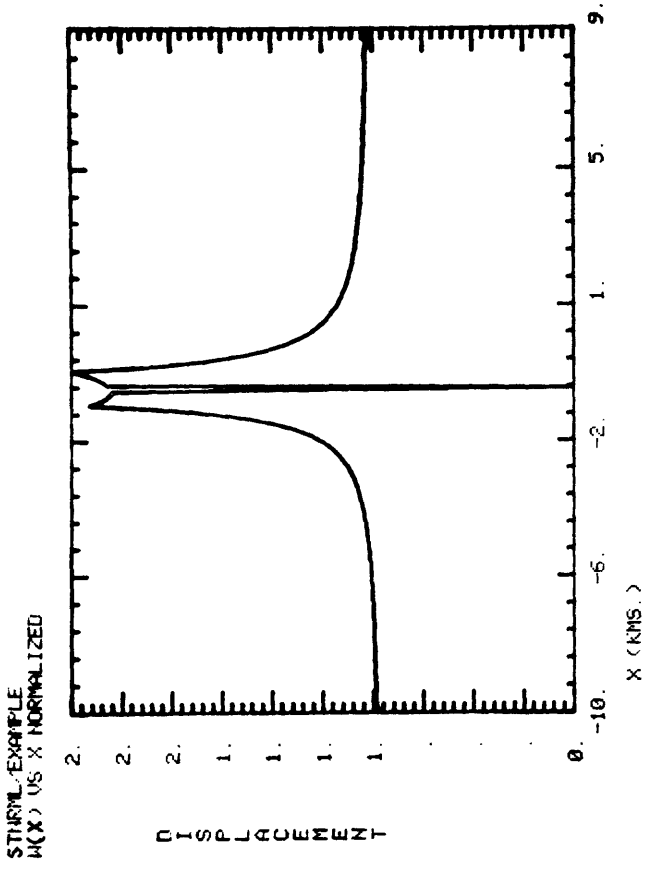
STNRM/EXAMPLE
XZ STRESS-NORMALIZED




```

1 0=KE-START, 1=NEW PLOT
2 3,
MIN MAX VALUES OF N(X) IS 2.5227E+01
SET VERTICAL SCALE: Y OR N(=BLANK)
SET HORIZONTAL SCALE: Y OR N(=BLANK)
SKIP?

```




```

1
0=RE-START, 1=NEW PLOT
2)
MU1 =          3.0000E+11      MU2 =          3.0000E+10
MU3 =          3.0000E+11      MUH =          3.0000E+11
M =          8.1815E-01        K =          8.1815E-01
DISLOCATION POSITION =          .500. -          1.000 KMS.
SLAB THICKNESS =          1.000 KMS.
BURGER'S VECTOR =          -1.000 NMS.
NO STRESS/STRAIN US. X
0=RE-START
1=STOP
2=RE-START WITH SAME VALUES, DIFFERENT PROFILE
1)
UP = ^EDIT

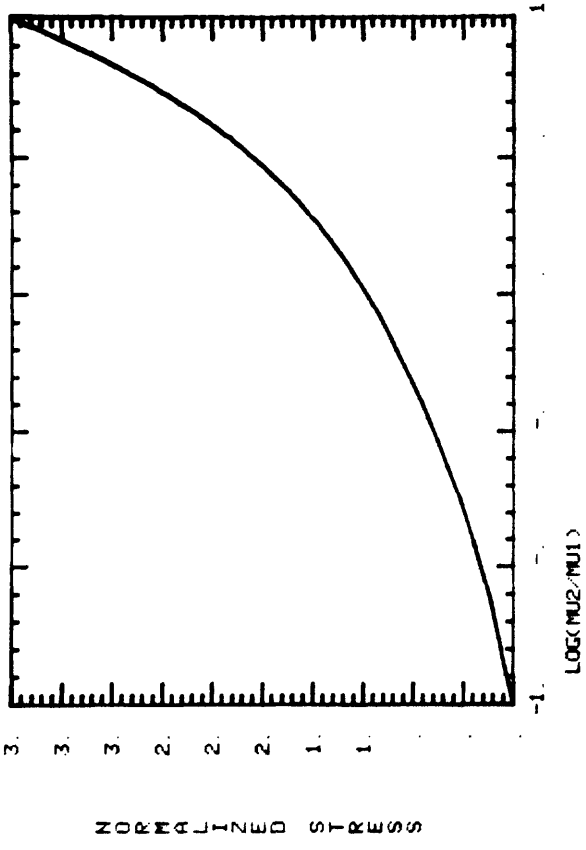
```

EXAMPLE FROM SHRSTN

NOTE NUMBER OF DATA POINTS = 101
 UPPER LIMIT ON SERIES = 20
 MIN MAX VALUES OF NORMALIZED STRESS
 VER MOD SHEAR MODULUS CONTRAST 3.1196E+00
 2.3214E-01
 PLOT TITLE: 80 CHARACTERS MAXIMUM
 STRS/MODULUS RATIO VARIES FROM 10 TO 0.101
 SET VERTICAL SCALE: Y OR N=(BLANK)
 SET HORIZONTAL SCALE: Y OR N=(BLANK)
 1 1 1 MAX X VALUES
 1 1 1
 5 1 1 0

SHEAR PROFILE--IN US UNIT --M1,M2,LOD,M3,M4,M5,M6,M7,M8,M9,M10,M11
 3E11 3E11 3E11
 SKEW MODULUS OF HOMOGENEOUS HALF--PHASE
 3E11
 POSITION OF DISLOCATION--IN RMS --M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,M11
 NOTE --H 0 --LAMEDA
 5 1 1
 THICKNESS OF SLAB: IN KMS --D
 1 1
 NUMBERS VECTOR: IN RMS --B
 BCU FOR RIGHT LATERAL SLIP
 1 1
 1=AMPLITUDE VS X, 2=AMPLITUDE VS Y
 1 1
 1=XZ STRESS, 2=YZ STRESS
 1 1
 1=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
 31
 ENTER X COORDINATE OF POINT
 -- 51
 DEPTH OF PROFILE, IN KMS.
 0

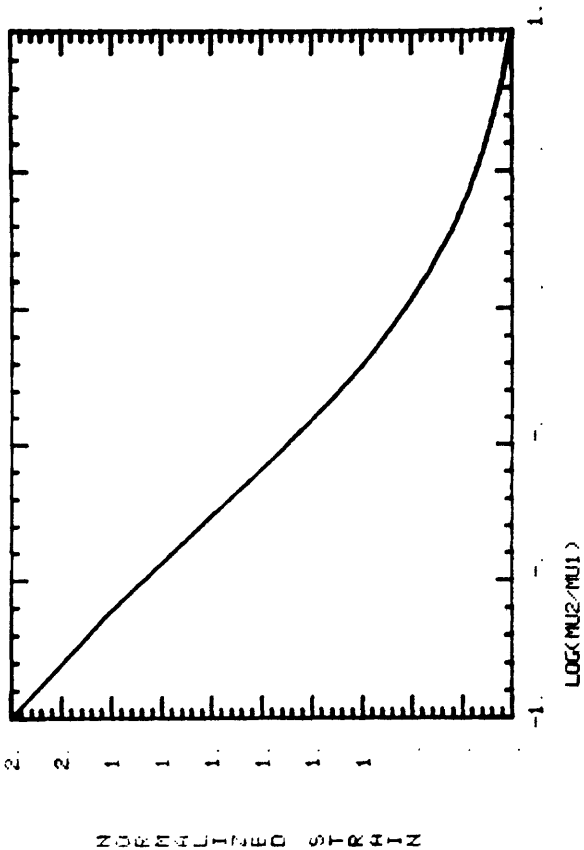
SHIFT IN MODULUS RATIO VARIES FROM 10 TO 0.10
 XZ STRESS-NORMALIZED



```

0= 1.000000E+01 1 NEW PLOT
2
3.000000E+01 100 VALUES OF NORMALIZED STRESS AT
3.000000E+01 100 VALUES OF MODULUS CONTRAST
SET 3.75200E-01 2.3214E+08
SET 0.000000E+00 0.000000E+00 (OF H=BLANK)
SET 0.000000E+00 0.000000E+00 (OF H=BLANK)
3.000000E+01 100 VALUES
  
```

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



0=PLC START, 1=NEW PLOT
 Z1
 MU1 = 3.0000E+11 MU2 LOW = 3.0000E+10
 MU HIGH = 3.0000E+12 MU M = 3.0000E+11
 MUH = 3.0000E+11 M = -8.1653E-01
 K = -8.1653E-01
 STRESS/STRAIN US X 500. = 1.000 KMS.
 INITIATION POSITION = 0 500. = 1.000 KMS.
 GAGE THICKNESS = 0 1.000 KMS.
 GAGE'S VECTOR = 0 -1.000 MMS.
 PLC START
 UP
 PLC-START WITH SAME VALUES. DIFFERENT PROFILE
 0 = EDIT

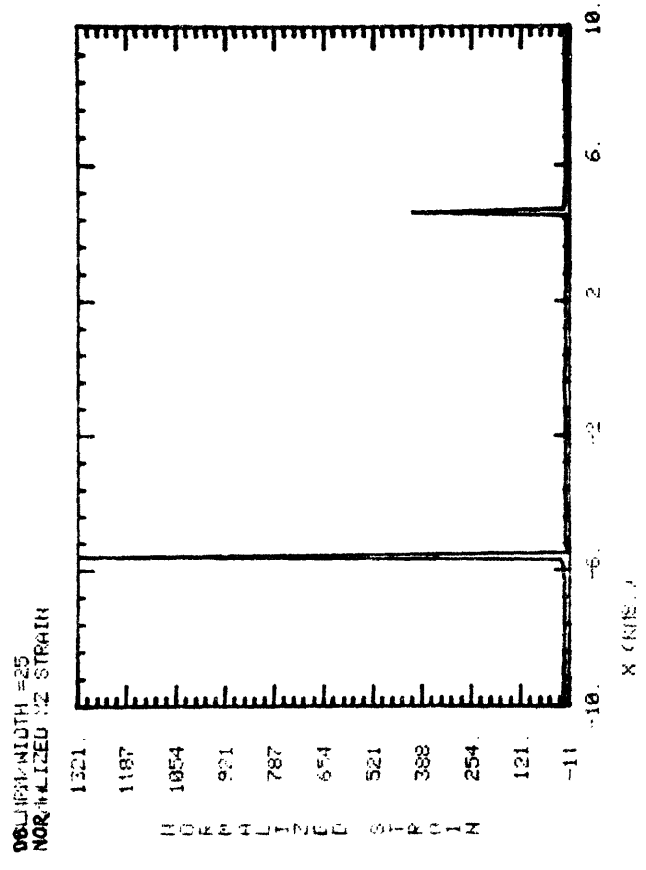
EXAMPLE FROM DBLNRM

C39a


```

1 0801 START, 1=NEW PLOT
2  WITH MAX VALUES OF NORMALIZED X2 STRAIN.
   -1 1889E+01 1.351E+01
3  SET VERTICAL SCALE Y OF PLOT=0.01
4  SET HORIZONTAL SCALE X OF PLOT=0.01
5  SKIP

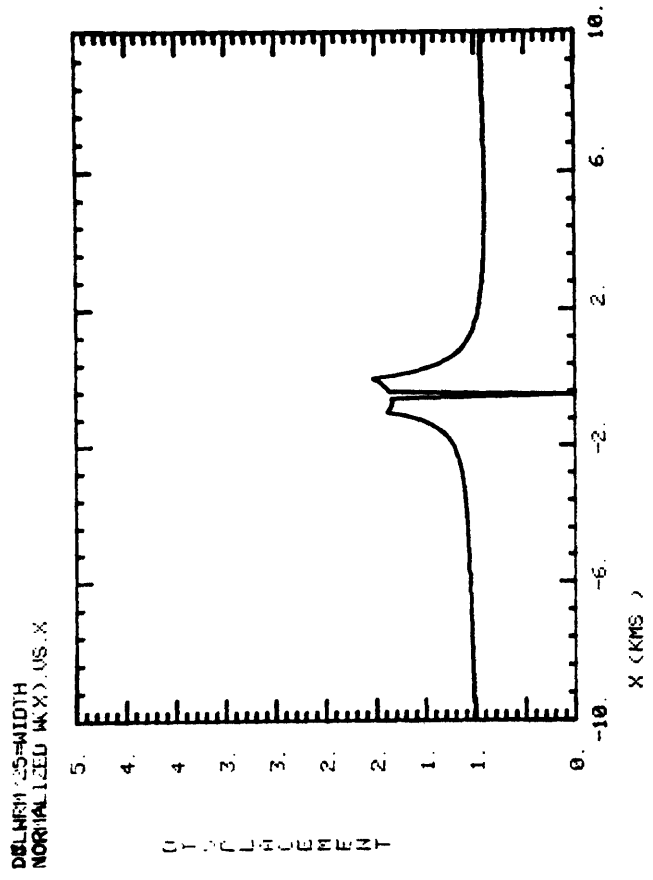
```




```

1 0=RE-START, 1=NEW PLOT
2 1
.....
WARNING--CAUTION--WARNING
NORMALIZED DISPLACEMENTS MAY BE IN ORBLCT
CHECK DELNRM DOCUMENTATION FOR EXPLANATION
.....
MIN/MAX VALUES OF NORMALIZED WIDTHS %
0 2 DELNRM
SET VERTICAL SCALE? Y OR N=BLANK)
Y1
MIN/MAX Y VALUES
0 51
SET HORIZONTAL SCALE? Y OR N=BLANK)
Y1
MIN/MAX X VALUES
-10 101
SKIP?

```



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.

WIDTH OF SLIP ZONE = 25.000

SLAB THICKNESS = 1.000 KMS.

BURGERS VECTOR = -1.000 MMS.

Y2 STRESS/STRAIN VS. X

0=RE-START

1=STOP

2=RE-START WITH SAME VALUES, DIFFERENT PROFILE

1'

LISTING OF ST3PHS

```

1  DELETE(LG0,OUTPUT,ST3PHS)
2  ST3PHS.
3  CXIT.
4  LIBCOPY (GRAPHIC, TXLG0/RR, TXLGC)
5  LIBCOPY (JDRAT, NPLG0/RR, NPLG0)
6  DELETE(LG0,CUTFUT,ST3PHS)
7  RUN76(S)
8  LINK(F=LG0,F=TXLG0,F=NPLG0,B=ST3PHS)
9  ST3PHS.
10 FIN.
11 EOR
12     PROGRAM ST3PHS(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(500),Y(500),SXZ(500),SYZ(500),AB(30)
19     DIMENSION PLOTY(500),PLCTX(500)
20     DATA YMIN,YMAX,XCUT,CUTYMAX/-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     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,LAMEDA)*,/,
32     1 *NOTE---A>0., LAMEDA>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(AE) $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(AE)
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

```

```

59      WRITE(7,25) NUM,IUPPER
60      25 FORMAT(*NOTE--NUMBER OF DATA POINTS = *,I5,/,
61      1      *      UPPER LIMIT CN SERIES = *,I5)
62      AM=(AMU3-AMU2)/(AMU3+AMU2) $IF(AM.EQ.0.)AM=1.E-20
63      AK=(AMU1-AMU2)/(AMU1+AMU2) $IF(AK.EQ.0.)AK=1.E-20
64      BOUND=-D $AL=-ALAMBD $AR=-A
65      IF(KFLAG.EQ.1 .AND.AL.GT.SET)GOTO74
66      GOTO75
67      74 WRITE(7,7E)
68      76 FORMAT(*WARNING---PROFILE BELOW DISLOCATION LINE*,/,
69      1      *SOLUTION INVALID, TRY AGAIN*)
70      GOTO63
71      75 IF(KFLAG.EQ.1)GOTO90
72      PCIST=ABS(SET-(-A))
73      IF(RDIST.LT.YINC)WRITE(7,9I)
74      IF(RDIST.LT.YINC)GOTO63
75      GOTO92
76      90 RDIST=ABS(SET-(-ALAMBD))
77      IF(RDIST.LT.XINC)WRITE(7,9I)
78      IF(RDIST.LT.XINC)GOTO63
79      91 FORMAT(*WARNING---PROFILE TOO CLOSE TO DISLOCATION CORE*,/,
80      1      *CHOOSE NEW PROFILE OR INCREASE GRID POINT DENSITY*)
81      92 CONTINUE
82      IF(KFLAG.EQ.1.AND.SET.GT.0.)WRITE(7,93)
83      IF(KFLAG.EQ.1.AND.SET.GT.0.)GOTO63
84      93 FORMAT(*WARNING---PROFILE ABOVE SURFACE OF HALF-SPACE*,/,
85      1      *SOLUTION INVALID, TRY AGAIN*)
86      DO 200 J=1,NUM
87      IF(KFLAG.EQ.1)X(J)=XMIN+(XINC*(J-1))
88      IF(KFLAG.EQ.1) Y(J)=SET
89      IF(KFLAG.EQ.2)X(J)=SET
90      IF(KFLAG.EQ.2)Y(J)=YMIN+(YINC*(J-1))
91      IF(X(J).EQ.AR.AND.Y(J).EQ.AL)GOTO32
92      GO TO 33
93      32 X(J)=X(J)+1.E-10 $Y(J)=Y(J)+1.E-10
94      33 IF(X(J) .LT.BOUND)GOTO103
95      IF(X(J).GE.BOUND.AND.X(J).LE.0.)GOTO102
96      101 IF(IFLAG .EQ. 2)GOTO12
97      SUM1=SUM2=0.
98      DO 1 I=1,IUPPER
99      NU=I-1
100     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
101     TERM1=((AM*AK)**NU)*F1
102     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,I,D,ALAMBD,F2)
103     TERM2=(AM**I)*(AK**NU) *F2
104     SUM1=SUM1+TERM1      $SUM2=SUM2+TERM2
105     1 CONTINUE
106     SXZ(J)=(((-AMU1*9)/(2.*PI))*(1.-AK))*(SUM1+SUM2) $GOTO200
107     12 SUM1=SUM2=SUM3=SUM4=0.
108     DO 5 I=1,IUPPER
109     NU=I-1
110     CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
111     CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMEL,F2)
112     CALL FUNC1(X(J),-A,I,D,Y(J),ALAMBD,F3)
113     CALL FUNC1(X(J),-A,I,D,Y(J),-ALAMEC,F4)
114     TERM1=((AM*AK)**NU)*F1 $TERM2=((AM*AK)**NU)*F2
115     TERM3=(AM**I)*(AK**NU)*F3 $TERM4=(AM**I)*(AK**NU)*F4
116     SUM1=SUM1+TERM1 $SUM2=SUM2+TERM2 $SUM3=SUM3+TERM3

```

```

117     SUM4=SUM4+TERM4
118     5  CCNTINUE
119     SYZ(J)=((AMU1*B)*(1.-AK)/(2.*PI))*(SUM1+SUF3-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  CCNTINUE
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),ALAMED,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  CCNTINUE
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,ALAMED,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  CCNTINUE
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  CCNTINUE

```



```

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 CCNTINUE
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 CYNES/(CM2)*)
189      29 FORMAT(*MIN/MAX VALUES OF YZ STRESS--UNITS ARE CYNES/(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 CCNTINUE
216      417 LU=7 $NLNGX=1 $NLNGY=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)=10+XZ STRESS
221      IF(IFLAG.EQ.2) LTITL2(1)=10+YZ 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 $YLC=SDN $YUP=SUP
228      IF(KFLAG.EQ.2.AND. YMIN.LT.AL) XRT=XCUT
229      IF(KFLAG.EQ.2.AND. YMAX.GT.C.) XLT=CUTMAX
230      IF(KFLAG.EQ.2.AND. YMIN.GE.AL) XRT=PLOTX(NUM)
231      AA=YUP$IF(YLO.EQ.AA) YUP=YUP+1.$IF(YLO.EQ.AA) YLC=YLC-1.
232      WRITE(7,37)

```

```

233 37 FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
234 READ(7,38)CHARAC
235 38 FCRMAT(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 )GOTC42
244 WRITE(7,43)
245 43 FCRMAT(*MIN/MAX X VALUES*)
246 CALL GETNUM(AB) $IF(KFLAG.EQ.1)XLT=AB(1)
247 IF(KFLAG.EQ.1)XRT=AB(2)
248 IF(KFLAG.EQ.2)XLT=-AB(2)
249 IF(KFLAG.EQ.2)XRT=-AB(1)
250 42 WRITE(7,44)
251 44 FCRMAT(*SKIP?*)
252 READ(7,38)CHARAC
253 IF(CHARAC.EQ.1HN .OR. CHARAC.EQ.1H )CALL PLOTS(PLCTY,PLOTX,1,NUM)
254 WRITE(7,46)
255 46 FCRMAT(*0=RE-START, 1=NEW PLOT*)
256 READ(7,71)LCHECK
257 71 FORMAT(I1)
258 IF(LCHECK.EQ.0)GOTO63$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)=(PLCTY(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(PLCTY(I).LT.SDN)SDN=PLCTY(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)=10+XZ STRAIN
274 IF(IFLAG.EQ.2)LTITL2(1)=10+YZ STRAIN
275 418 KY(1)=10+MICROSTRAI $KY(2)=10HN
276 YLC=SDN $YUP=SUP
277 AA=YUP $IF(YLO.EQ.AA)YUP=YUP+1. $IF(YLO.EQ.AA)YLO=YLO-1.
278 WRITE(7,37) $PEAD(7,38)CHARAC
279 IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC55
280 WRITE(7,40) $CALL GETNUM(AE) $YLO=AB(1) $YUP=AB(2)
281 55 WRITE(7,41)$READ(7,38)CHARAC
282 IF(CHAPAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC301
283 WRITE(7,43) $CALL GETNUM(AE)
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) $PEAD(7,38)CHARAC
287 IF(CHARAC.EQ.1HN.OF.CHARAC.EQ.1H )CALL PLCTS(PLCTY,PLOTX,1,NUM)
288 56 WRITE(7,46) $CALL GETNUM(AE)$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 $SUP=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=-ALAMBC-(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 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 CF W(X) VS. X*)
323 61 FORMAT(*MIN/MAX VALUES CF 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 )GOTC62
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 )GOTC303
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) $READ(7,38)CHARAC
339 IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H ) CALL PLCTS(PLCTY,PLOTX,1,NUM)
340 WRITE(7,46)$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,ALAMBDA,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,/,

```

```

349      1      * K = *,E20.4)
350  47  FORMAT(*DISLOCATION POSITION = -,F10.3,*, -,F10.3,* KMS.*,/,
351      1      *SLAB THICKNESS      = *,F10.3,* KMS.*,/,
352      1      *BURGERS VECTOR      = *,F10.3,* MMS.*)
353  49  FORMAT(A2,* STRESS/STRAIN \S. *,A2)
354      WRITE(7,64)
355  64  FORMAT(*U=RE-START*,/,*1=STOP*,/,
356      1      *2=RE-START WITH SAME VALUES, DIFFERENT PROFILE*)
357      CALL GETNUM(AB) $LJK=AB(1)
358      IF(LJK.EQ.0)GOTO48 $IF(LJK.EQ.2)GO TO 70
359      STOP $END
360      SUBROUTINE FUNC1(P,A,NU,D,G,ALAMB,D,F)
361      R=(((P+A+(2.*NU*D))**2)+((G+ALAMB)**2))
362      IF(R.EQ.0.)R=1.E-10$F=(P+A+(2.*NU*D))/R
363      RETURN $END
364      SUBROUTINE FUNC2(P,Q,R,S,A,NU,D,ALAMB,D,F)
365      T=(((R+A+(2.*NU*D))**2)+((S+ALAMB)**2))
366      IF(T.EQ.0.)T=1.E-10$F=(P+G)/T
367      RETURN $END
368      SUBROUTINE FUNC3(P,Q,R,S,F)
369      T=(((R+S)**2)+((P+Q)**2))
370      IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
371      RETURN $END
372      SUBROUTINE GETNUM(R)
373      DIMENSION R(1),L(80)
374      READ(7,9)L $I=J=0
375      6  J=J+1 $N=P=S=0 $M=F=1
376      5  I=I+1 $IF(I.GT.80)RETURN $D=L(I) $K=4
377      IF(D.EQ.38)K=2 $IF(D.GE.27.A.D.LE.36)K=1
378      IF(D.EQ.47)K=3 $K=K+S $GCTC(1,2,3,5,1,4,3,4)K
379      1  N=N*10+D-27 $S=4 $GOTO5
380      2  M=-1 $S=4 $GOTO5
381      3  P=I $S=4 $GOTO5
382      4  IF(P.NE.0)F=10.** (I-P-1) $R(J)=N/F*M $GOTO6
383      9  FORMAT(80F1)
384      END

```

LISTING OF ST3PSZ

```

1  DELETE(LGO,OUTPUT,ST3PSZ)
2  ST3PSZ.
3  CXIT.
4  LIBCOPY(GRAPHIC, TXLGO/RR, TXLGC)
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/TAPCOL/TVPUL(8)
14   COMMON/TVTUNE/ITUNE(30)
15   COMMON/JPLCT/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), SXZ(200), SYZ(200), AB(30)
19   DIMENSION PLOTY(200), PLCTX(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,LAMEDA)*,/,
33   1 *NOTE---A>0., LAMEDA>0.*)
34   CALL GETNUM(AB) $A=AB(1) $ALAMBDA=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(AE) $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(AE)
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

```

```

59     IF(KFLAG.EQ.1)XINC=(XMAX-XMIN)/NUM
60     IF(KFLAG.EQ.2)YINC=(YMAX-YMIN)/NUM
61     WRITE(7,25) NUM,IUPPER
62     25 FORMAT(*NOTE--NUMBER OF DATA POINTS = *,I5,/,
63     1      *      UPPER LIMIT (N SERIES = *,I5)
64     AM=(AMU3-AMU2)/(AMU3+AMU2)$IF(AM.EQ.0.)AM=1.E-20
65     AK=(AMU1-AMU2)/(AMU1+AMU2) $IF(AK.EQ.0.)AK=1.E-20
66     BOUND=-D $AL=-ALAMED $AR=-A
67     IF(KFLAG.EQ.1 .AND.AL.GT.SET)GOTO74
68     GCTO75
69     74 WRITE(7,76)
70     76 FORMAT(*WARNING---PROFILE BELOW DISLOCATION LINE*,/,
71     1      *SOLUTION INVALID, TRY AGAIN*)
72     GOTO63
73     75 IF(KFLAG.EQ.1)GOTO90
74     RDIST=ABS(SET-(-A))
75     IF(RDIST.LT.YINC)WRITE(7,91)
76     IF(RDIST.LT.YINC)GCTO63
77     GCTO92
78     90 RDIST=ABS(SET-(-ALAMBD))
79     IF(RDIST.LT.XINC)WRITE(7,91)
80     IF(RDIST.LT.XINC)GOTO63
81     91 FORMAT(*WARNING---PROFILE TOO CLOSE TO DISLOCATION CORE*,/,
82     1      *CHOOSE NEW PROFILE OR INCREASE GRIC PCINT DENSITY*)
83     92 CONTINUE
84     IF(KFLAG.EQ.1.AND.SET.GT.0.)WRITE(7,93)
85     IF(KFLAG.EQ.1.AND.SET.GT.0.)GOTO63
86     93 FORMAT(*WARNING---PROFILE ABOVE SURFACE OF HALF-SPACE*,/,
87     1      *SOLUTION INVALID, TRY AGAIN*)
88     DO 910 MK=1,2
89     IF(MK.EQ.2)ALAMBD=ALAMBD+WIDTH
90     IF(MK.EQ.2)B=-B
91     DO 200 J=1,NUM
92     IF(KFLAG.EQ.1)X(J)=XMIN+(XINC*(J-1))
93     IF(KFLAG.EQ.1) Y(J)=SET
94     IF(KFLAG.EQ.2)X(J)=SET
95     IF(KFLAG.EQ.2)Y(J)=YMIN+(YINC*(J-1))
96     IF(X(J).EQ.AR.AND.Y(J).EQ.AL)GOTO32
97     GO TO 33
98     32 X(J)=X(J)+1.E-10 $Y(J)=Y(J)+1.E-10
99     33 IF(X(J) .LT.BOUND)GOTO103
100    IF(X(J) .GE.BOUND.AND.X(J) .LE.0.)GOTO102
101    101 IF(IFLAG .EQ. 2)GOTO12
102    SUM1=SUM2=0.
103    DO 1 I=1,IUPPER
104    NU=I-1
105    CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
106    TERM1=((AM*AK)**NU)*F1
107    CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,I,D,ALAMBD,F2)
108    TERM2=(AM**I)*(AK**NU) *F2
109    SUM1=SUM1+TERM1      $SUM2=SUM2+TERM2
110    1 CONTINUE
111    SXZ(J)=(((AMU1*B)/(2.*PI))*(1.-AK))*(SUM1+SUM2) $GOTO200
112    12 SUM1=SUM2=SUM3=SUM4=0.
113    DO 5 I=1,IUPPER
114    NU=I-1
115    CALL FUNC1(X(J),A,NU,D,Y(J),ALAMED,F1)
116    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),-ALAMEC,F4)
119     TERM1=((AM*AK)**NU)*F1 ‡TERM2=((AM*AK)**NU)*F2
120     TERM3=(AM**I)*(AK**NU)*F3 ‡TERM4=(AM**I)*(AK**NL)*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,ALAMBC,F1)
130     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMEC,F2)
131     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,NL,D,ALAMEC,F3)
132     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NL,D,ALAMEC,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) ‡GCTO200
139 13 SUM1=SUM2=0.
140     DO 3 NU=1,IUPPER
141     CALL FUNC1(X(J),A,NU,D,Y(J),ALAMED,F1)
142     CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMED,F2)
143     CALL FUNC1(X(J),-A,NU,D,Y(J),ALAMEC,F3)
144     CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMEC,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),-ALAMBC,F5)
148     CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F6)
149     CALL FUNC1(X(J),-A,NU,D,Y(J),-ALAMED,F7)
150     CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMED,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,ALAMEC,F1)
165     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMED,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 ‡GCTC200
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),ALAMEC,F1)
173     CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMED,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)=SXZSLF(I,1)+SXZSLP(I,2)
188      930 IF(IFLAG.EQ.2)SYZ(I)=SYZSLF(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 PLCTX(L)=SYZ(I)
220      DO 82 I=1,NUM
221      IF(KFLAG.EQ.2.AND.PLOTX(I).LT.AL)LLP=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 $NLGX=1 $NLGY=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)

```

```

233 IF(IFLAG.EQ.1)LTITL2(1)=10FXZ STRESS
234 IF(IFLAG.EQ.2)LTITL2(1)=10FYZ STRESS
235 DC 36 I=2,8
236 36 LTITL2(I)=10H
237 IF(KFLAG.EQ.1)KX(1)=10HX (KMS.)
238 IF(KFLAG.EQ.2)KX(1)=10HY (KMS.)
239 KX(2)=10H $KY(1)=10HSTRESS $KY(2)=10H
240 XRT=PLOTX(NUM) $XLT=PLOTX(1) $MAJX=5 $MAJY=10 $YLC=SDN $YUP=SUP
241 IF(KFLAG.EQ.2.AND.YMIN.LT.AL)XRT=XCUR
242 IF(KFLAG.EQ.2.AND.YMAX.GT.C.)XLT=CUTMAX
243 IF(KFLAG.EQ.2.AND.YMIN.GE.AL)XRT=PLOTX(NUM)
244 AA=YUP$IF(YLO.EQ.AA)YUP=YUP+1.$IF(YLO.EQ.AA)YLC=YLC-1.
245 WRITE(7,37)
246 37 FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
247 READ(7,38)CHARAC
248 38 FORMAT(A1)
249 IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO39
250 WRITE(7,40)
251 40 FORMAT(*MIN/MAX Y VALUES*)
252 CALL GETNUM(AB) $YLO=AB(1) $YUP=AB(2)
253 39 WRITE(7,41)
254 41 FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
255 READ(7,38)CHARAC
256 IF(CHARAC.EQ.1HN.OF.CHARAC.EQ.1H )GOTC42
257 WRITE(7,43)
258 43 FORMAT(*MIN/MAX X VALUES*)
259 CALL GETNUM(AB) $IF(KFLAG.EQ.1)XLT=AB(1)
260 IF(KFLAG.EQ.1)XRT=AB(2)
261 IF(KFLAG.EQ.2)XLT=-AB(2)
262 IF(KFLAG.EQ.2)XRT=-AB(1)
263 42 WRITE(7,44)
264 44 FORMAT(*SKIP?*)
265 READ(7,38)CHARAC
266 IF(CHARAC.EQ.1HN .OR. CHARAC.EQ.1H )CALL PLOTS(PLCTY,PLCTX,1,NUM)
267 WRITE(7,46)
268 46 FORMAT(*0=RE-START, 1=NEW PLOT*)
269 READ(7,71)LCHECK
270 71 FORMAT(I1)
271 IF(LCHECK.EQ.0)GOTO63$IF(LCHECK.EQ.1)GOTO417
272 DO 50 I=1,NUM
273 IF(X(I).LT.BOUND)AMU=AMU3
274 IF(X(I).GE.BOUND.AND.X(I).LE.0.)AMU=AMU2
275 IF(X(I).GT.0.)AMU=AMU1
276 IF(AMU.EQ.0.)AMU=1.E-10
277 50 PLCTY(I)=(PLOTY(I)/(AMU))*1.E+06
278 STSMID=(PLOTY(NUM)-PLOTY(1))/2.
279 SUP=SDN=STSMID
280 DC 51 I=1,NUM
281 IF(PLOTY(I).GT.SUP)SUP=PLCTY(I)
282 51 IF(PLCTY(I).LT.SDN)SDN=PLCTY(I)
283 IF(IFLAG.EQ.1)WRITE(7,52) $IF(IFLAG.EQ.2)WRITE(7,53)
284 52 FORMAT(*MIN/MAX VALUES CF XZ MICROSTRAIN*)
285 53 FORMAT(*MIN/MAX VALUES CF YZ MICROSTRAIN*)
286 WRITE(7,30)SDN,SUP
287 IF(IFLAG.EQ.1)LTITL2(1)=10FXZ STFAIN
288 IF(IFLAG.EQ.2)LTITL2(1)=10FYZ STRAIN
289 418 KY(1)=10HMICROSTRAI $KY(2)=10HN
290 YLO=SDN $YUP=SUP

```

```

291 AA=YUP $IF(YLO.EQ.AA)YUP=YUP+1. $IF(YLC.EQ.AA)YLO=YLO-1.
292 WRITE(7,37) $READ(7,38)CHARAC
293 IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC55
294 WRITE(7,40) $CALL GETNUM(AE) $YLO=AB(1) $YUP=AE(2)
295 55 WRITE(7,41) $READ(7,38) CHARAC
296 IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTC301
297 WRITE(7,43) $CALL GETNUM(AE)
298 IF(KFLAG.EQ.1)XLT=AB(1) $IF(KFLAG.EQ.1)XRT=AB(2)
299 IF(KFLAG.EQ.2)XLT=-AB(2) $IF(KFLAG.EQ.2)XRT=-AB(1)
300 301 WRITE(7,44) $READ(7,38)CHARAC
301 IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )CALL PLCTS(PLCTY,PLOTX,1,NUM)
302 56 WRITE(7,46) $CALL GETNUM(AE)$LCHECK=AB(1)
303 IF(LCHECK.EQ.0)GOTO63$IF(LCHECK.EQ.1)GCTO418
304 IF(KFLAG.NE.IFLAG)WRITE(7,57)
305 57 FORMAT(*NOTE--ONLY W(X) VS. X OR W(Y) VS. Y IS COMPUTED*)
306 IF(KFLAG.NE.IFLAG)GOTO63
307 IF(KFLAG.EQ.1)DINC=XINC*1.E+06
308 IF(KFLAG.EQ.2)DINC=YINC*1.E+06
309 SUM=0.
310 DO 58 I=1,NUM
311 TERM=2.*PLOTY(I)*DINC*1.E-06 $SUM=SUM+TERM
312 58 PLCTY(I)=SUM
313 MCHECK=0
314 DO 789 M=1,NUM
315 IF(KFLAG.EQ.1)GRD=ABS((XMA)-XMIN)/NUM)
316 IF(KFLAG.EQ.2)GRD=ABS((YMA)-YMIN)/NUM)
317 IF(KFLAG.EQ.1)GOTO790
318 GRD1=-ALAMED+(GRD/100.) $GRD2=-ALAMB0-(GRD/100.)
319 IF(Y(M).GT.GRD2.AND.Y(M).LT.GRD1)KMID=M
320 IF(Y(M).GT.GRD2.AND.Y(M).LT.GRD1)MCHECK=1
321 GOTO789
322 790 GRD1=-A+(GRD/100.) $GRD2=-A-(GRD/100.)
323 IF(X(M).GT.GRD2.AND.X(M).LT.GRD1)KMID=M
324 IF(X(M).GT.GRD2.AND.X(M).LT.GRD1)MCHECK=1
325 789 CONTINUE
326 IF(MCHECK.EQ.0)STSMID=((PLCTY(NUM)-PLCTY(1))/2.)+PLOTY(1)
327 IF(MCHECK.EQ.1)STSMID=PLOTY(KMID)
328 DO 73 I=1,NUM
329 73 PLCTY(I)=PLOTY(I)-STSMID
330 STSMID=(PLCTY(NUM)-PLOTY(1))/2.
331 SUP=SDN=STSMID
332 DO 59 I=1,NUM
333 IF(PLOTY(I).GT.SUP)SUP=PLOTY(I)
334 59 IF(PLOTY(I).LT.SDN)SDN=PLOTY(I)
335 IF(IFLAG.EQ.1) WRITE(7,60) $IF(IFLAG.EQ.2)WRITE(7,61)
336 IF(KFLAG.EQ.1.AND.SET.NE.C.)WRITE(7,87)
337 87 FORMAT(*CAUTION---THIS IS NOT THE TOTAL DISPLACEMENT*)
338 60 FORMAT(*MIN/MAX VALUES CF W(X) VS. X*)
339 61 FORMAT(*MIN/MAX VALUES CF W(Y) VS. Y*)
340 WRITE(7,30) SDN,SUP
341 419 IF(IFLAG.EQ.1)LTITL2(1)=10HW(X) VS. X
342 IF(IFLAG.EQ.2)LTITL2(1)=10HW(Y) VS. Y
343 KY(1)=10HDISPLACEME $KY(2)=10HNT-MMS.
344 YLO=SDN $YUP=SUP$AA=YUP$IF(YLO.EQ.AA)YUP=YUP+1.
345 IF(YLO.EQ.AA)YLO=YLO-1.
346 WRITE(7,37) $READ(7,38)CHARAC
347 IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO62
348 WRITE(7,40) $CALL GETNUM(AE) $YLO=AB(1) $YUP=AE(2)

```

```

349 62 WRITE(7,41) $READ(7,38)CHARAC
350 IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO303
351 WRITE(7,43) $CALL GETNUM(AE)
352 IF(KFLAG.EQ.1)XLT=AB(1) $IF(KFLAG.EQ.1)XRT=AB(2)
353 IF(KFLAG.EQ.2)XLT=-AB(2) $IF(KFLAG.EQ.2)XRT=-AB(1)
354 303 WRITE(7,44) $READ(7,38)CHARAC
355 IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H ) CALL PLCTS(PLOT,Y,PLOTX,1,NUM)
356 WRITE(7,46)$CALL GETNUM(AB)$LCHECK=AB(1)$IF(LCHECK.EQ.1)GC TO419
357 63 ALAMB0=ALAMBC-WIDTH $B=-B
358 WRITE(7,45)AMU1,AMU2,AMU3,AM,AK
359 BB=B*1.E+06
360 WRITE(7,47)A,ALAMBC,WIDTH,C,BB
361 IF(IFLAG.EQ.1)CD1=2HXZ$IF(KFLAG.EQ.1)CD2=1HX
362 IF(IFLAG.EQ.2)CD1=2HYZ$IF(KFLAG.EQ.2)CD2=1HY
363 WRITE(7,49)CD1,CD2
364 45 FORMAT(*MU1 = *,E20.4,5X,*MU2 = *,E20.4,/,
365 1 *MU3 = *,E20.4,5X,* M = *,E20.4,/,
366 1 * K = *,E20.4)
367 47 FCRMAT(*DISLOCATION POSITION = *,F10.3,*, *,F10.3,* KMS.*,/,
368 1 *WIDTH OF SLIP ZONE = *,F10.3,/,
369 1 *SLAB THICKNESS = *,F10.3,* KMS.*,/,
370 1 *BURGERS VECTOR = *,F10.3,* MMS.*)
371 49 FORMAT(A2,* STRESS/STRAIN 1S. *,A2)
372 WRITE(7,64)
373 64 FORMAT(*0=RE-START*,/,*1=STOP*,/,
374 1 *2=RE-START WITH SAME VALUES, DIFFERENT PROFILE*)
375 CALL GETNUM(AB) $LJK=AB(1)
376 IF(LJK.EQ.0)GOTO48 $IF(LJK.EQ.2)GC TO 70
377 STOP $END
378 SUBROUTINE FUNC1(P,A,NU,D,G,ALAMED,F)
379 R=(((P+A+(2.*NU*D))**2)+((G+ALAMED)**2))
380 IF(R.EQ.0.)R=1.E-10$F=(P+A+(2.*NU*D))/R
381 RETURN $END
382 SUBROUTINE FUNC2(P,Q,R,S,A,NU,D,ALAMBC,F)
383 T=(((R+A+(2.*NU*D))**2)+((S+ALAMBC)**2))
384 IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
385 RETURN $END
386 SUBROUTINE FUNC3(P,Q,R,S,F)
387 T=(((P+S)**2)+((P+Q)**2))
388 IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
389 RETURN $END
390 SUBROUTINE GETNUM(R)
391 DIMENSION F(1),L(80)
392 READ(7,9)L $I=J=0
393 6 J=J+1 $N=P=S=0 $M=F=1
394 5 I=I+1 $IF(I.GT.80)RETURN $D=L(I) $K=4
395 IF(D.EQ.38)K=2 $IF(D.GE.27.A.D.LE.36)K=1
396 IF(D.EQ.47)K=3 $K=K+S $GOTO(1,2,3,5,1,4,3,4)K
397 1 N=N*10+D-27 $S=4 $GOTO5
398 2 M=-1 $S=4 $GOTO5
399 3 P=I $S=4 $GOTO5
400 4 IF(P.NE.0)F=10.** (I-P-1) $R(J)=N/F*M $GOTO6
401 9 FORMAT(80R1)
402 END

```

LISTING OF STNRML

C64a

```

1  DELETE(LGO,OUTPUT,STNRML)
2  STNRML.
3  CXIT.
4  LIBCOPY(GRAPHIC,TXLGO/RR,TXLGC)
5  LIBCOPY(JDRAT,NPLGO/RR,NPLGC)
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/JPLCT/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 PLOTX(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=AE(1)*(10.**IAB2) $SETMU2=AE(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) $LAMED=AE(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(*BUFFERS 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.EG.2)GOTO201
49  WRITE(7,21) $CALL GETNUM(AE) $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(AE)
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.*)

```

```

59 202 NUM=200 $IUPPER=10 $EPS=1.E-10 $PI=3.14159
60 IF(KFLAG.EQ.1)XINC=(XMAX-XMIN)/NUM
61 IF(KFLAG.EQ.2)YINC=(YMAX-YMIN)/NUM
62 WRITE(7,25) NUM,IUPPER
63 25 FORMAT(*NOTE--NUMBER OF DATA POINTS = *,I5,/,
64 1 * UPPER LIMIT (N SERIES = *,I5)
65 AMU1=SETMU1$AMU2=SETMU2$AMU3=SETMU3
66 AM=(AMU3-AMU2)/(AMU3+AMU2)$IF(AM.EQ.0.)AM=1.E-20
67 AK=(AMU1-AMU2)/(AMU1+AMU2)$IF(AK.EQ.0.)AK=1.E-20
68 BOUND=-D $AL=-ALAMBDA $AR=-A
69 IF(KFLAG.EQ.1 .AND.AL.GT.SET)GOTO74
70 GOTO75
71 74 WRITE(7,76)
72 76 FORMAT(*WARNING---PROFILE BELOW DISLOCATION LINE*,/,
73 1 *SOLUTION INVALID, TRY AGAIN*)
74 GOTO63
75 75 IF(KFLAG.EQ.1)GOTO90
76 RDIST=ABS(SET-(-A))
77 IF(RDIST.LT.YINC)WRITE(7,91)
78 IF(RDIST.LT.YINC)GOTO63
79 GOTO92
80 90 RDIST=ABS(SET-(-ALAMBDA))
81 IF(RDIST.LT.XINC)WRITE(7,91)
82 IF(RDIST.LT.XINC)GOTO63
83 91 FORMAT(*WARNING---PROFILE TOO CLOSE TO DISLOCATION CORE*,/,
84 1 *CHOOSE NEW PROFILE OR INCREASE GRID POINT DENSITY*)
85 92 CCNTINUE
86 IF(KFLAG.EQ.1.AND.SET.GT.0.)WRITE(7,93)
87 IF(KFLAG.EQ.1.AND.SET.GT.0.)GOTO63
88 93 FORMAT(*WARNING---PROFILE ABOVE SURFACE OF HALF-SPACE*,/,
89 1 *SOLUTION INVALID, TRY AGAIN*)
90 DO 230 K=1,2
91 IF(K.EQ.2)GOTO231
92 AMU1=AMU2=AMU3=AMUHOM
93 GOTO221
94 231 AMU1=SETMU1 $AMU2=SETMU2 $AMU3=SETMU3
95 221 AM=(AMU3-AMU2)/(AMU3+AMU2)$IF(AM.EQ.0.)AM=1.E-20
96 AK=(AMU1-AMU2)/(AMU1+AMU2)$IF(AK.EQ.0.)AK=1.E-20
97 DO 200 J=1,NUM
98 IF(KFLAG.EQ.1)X(J)=XMIN+(XINC*(J-1))
99 IF(KFLAG.EQ.1)Y(J)=SET
100 IF(KFLAG.EQ.2)X(J)=SET
101 IF(KFLAG.EQ.2)Y(J)=YMIN+(YINC*(J-1))
102 IF(X(J).EQ.AR.AND.Y(J).EQ.AL)GOTO32
103 GO TO 33
104 32 X(J)=X(J)+1.E-10 $Y(J)=Y(J)+1.E-10
105 33 IF(X(J) .LT.BOUND)GOTO103
106 IF(X(J).GE.BOUND.AND.X(J).LE.0.)GOTO102
107 101 IF(IFLAG .EQ. 2)GOTO12
108 SUM1=SUM2=0.
109 DO 1 I=1,IUPPER
110 NU=I-1
111 CALL FUNC2(Y(J),ALAMBDA,X(J),Y(J),A,NU,D,ALAMBDA,F1)
112 TERM1=((AM*AK)**NU)*F1
113 CALL FUNC2(Y(J),ALAMBDA,X(J),Y(J),-A,I,D,ALAMBDA,F2)
114 TERM2=(AM**I)*(AK**NU) *F2
115 SUM1=SUM1+TERM1 $SUM2=SUM2+TERM2
116 1 CCNTINUE

```

```

117     AMATRX(J,1,K)=((( -AMU1*B)/(2.*PI))*(1.-AK))*(SUM1+SUM2) $GOTO200
118     12 SUM1=SUM2=SUM3=SUM4=0.
119     DO 5 I=1,IUPPER
120     NU=I-1
121     CALL FUNC1(X(J),A,NU,D,Y(J),ALAMED,F1)
122     CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMED,F2)
123     CALL FUNC1(X(J),-A,I,D,Y(J),ALAMED,F3)
124     CALL FUNC1(X(J),-A,I,D,Y(J),-ALAMED,F4)
125     TERM1=((AM*AK)**NU)*F1 $TERM2=((AM*AK)**NU)*F2
126     TERM3=(AM**I)*(AK**NU)*F3 $TERM4=(AM**I)*(AK**NU)*F4
127     SUM1=SUM1+TERM1 $SUM2=SUM2+TERM2 $SUM3=SUM3+TERM3
128     SUM4=SUM4+TERM4
129     5 CONTINUE
130     AMATRX(J,1,K)=((AMU1*B)*(1.-AK)/(2.*PI))*(SUM1+SUM3-SUM2-SUM4)
131     GOTO200
132     102 IF(IFLAG.EQ.2)GOTO13
133     SUM1=0.
134     DO 7 NU=1,IUPPER
135     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
136     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMED,F2)
137     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,NU,D,ALAMBD,F3)
138     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMBD,F4)
139     TERM=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4) $SUM1=SUM1+TERM
140     7 CONTINUE
141     CALL FUNC3(Y(J),ALAMBD,X(J),A,F1)
142     CALL FUNC3(Y(J),ALAMBD,X(J),-A,F2)
143     TERM2=F1+(AK*F2)
144     AMATRX(J,1,K)=((-AMU2*B)/(2.*PI))*(TERM2+SUM1) $GOTO200
145     13 SUM1=SUM2=0.
146     DO 3 NU=1,IUPPER
147     CALL FUNC1(X(J),A,NU,D,Y(J),ALAMED,F1)
148     CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMED,F2)
149     CALL FUNC1(X(J),-A,NU,D,Y(J),ALAMED,F3)
150     CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMED,F4)
151     TERM1=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4)
152     SUM1=SUM1+TERM1
153     CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMED,F5)
154     CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMED,F6)
155     CALL FUNC1(X(J),-A,NU,D,Y(J),-ALAMED,F7)
156     CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMED,F8)
157     TERM2=((AM*AK)**NU)*(F5+(AK*F6)+((1./AK)*F7)+F8)
158     SUM2=SUM2+TERM2
159     3 CONTINUE
160     CALL FUNC3(X(J),A,Y(J),ALAMBD,F1)
161     CALL FUNC3(X(J),-A,Y(J),ALAMBD,F2)
162     CALL FUNC3(X(J),A,Y(J),-ALAMBD,F3)
163     CALL FUNC3(X(J),-A,Y(J),-ALAMBD,F4)
164     AMATRX(J,1,K)=((AMU2*B)/(2.*PI))*((F1+(AK*F2)+SUM1)-
165     1 (F3+(AK*F4)+SUM2))
166     GO TO 200
167     103 IF(IFLAG.EQ.2)GOTO14
168     SUM1=0.
169     DO 9 I=1,IUPPER
170     NU=I-1
171     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMED,F1)
172     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMED,F2)
173     TERM=(F1+(AK*F2))*((AK*AM)**NU) $SUM1=SUM1+TERM
174     9 CONTINUE

```



```

175     AMATRX(J,1,K)=((-AMU3*B)/(2.*PI))*(1.-AM)*SUM1 $GOTO200
176     14 SUM1=SUM2=0.
177     DO 11 I=1,IUPPER
178     NU=I-1
179     CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMED,F1)
180     CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMED,F2)
181     TERM1=((AM*AK)**NU)*(F1+(AK*F2))
182     SUM1=SUM1+TERM1
183     CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMED,F1)
184     CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMED,F2)
185     TERM2=((AM*AK)**NU)*(F1+(AK*F2)) $SUM2=SUM2+TERM2
186     11 CCNTINUE
187     AMATRX(J,1,K)=((AMU3*B)/(2.*PI))*(1.-AM)*(SUM1-SUM2)
188     200 CONTINUE
189     230 CONTINUE
190     DO 204 I=1,NUM
191     IF(AMATRX(I,1,1).EQ.0.)AMATRX(I,1,1)=1.E-10
192     204 PLCTY(I)=AMATRX(I,1,2)/AMATRX(I,1,1)
193     L12=NUM/2 $STSMID=FLOTY(L12)
194     SUP=SDN=STSMID
195     DO 26 I=1,NUM
196     IF(PLOTY(I).GT.SUP)SUP=PLCTY(I)
197     26 IF(PLOTY(I).LT.SDN)SDN=PLCTY(I)
198     IF(IFLAG.EQ.1)WRITE(7,28) $IF(IFLAG.EQ.2)WRITE(7,29)
199     28 FORMAT(*MIN/MAX VALUES CF >Z NORMALIZED STRESS*)
200     29 FORMAT(*MIN/MAX VALUES CF YZ NORMALIZED STRESS*)
201     WRITE(7,30)SDN,SUP
202     30 FORMAT(E15.4,5X,E15.4)
203     LUP=0
204     DO 31 I=1,NUM
205     IF(KFLAG.EQ.1)PLOTX(I)=X(I)
206     31 IF(KFLAG.EQ.2)PLOTX(I)=Y(I)
207     IF(KFLAG.EQ.2)GOTO83
208     GOTO85
209     83 DO 81 I=1,NUM
210     AMATRX(I,2,1)=PLOTY(I)
211     81 AMATRX(I,2,2)=PLOTX(I)
212     DO 84 I=1,NUM
213     L=NUM-I+1
214     PLOTY(L)=AMATRX(I,2,1)
215     84 PLCTX(L)=AMATRX(I,2,2)
216     DO 82 I=1,NUM
217     IF(KFLAG.EQ.2.AND.PLOTX(I).LT.AL)LUP=LUP+1
218     82 IF(LUP.EQ.1)XCUT=PLOTX(I-1)
219     LDN=0
220     DO 86 I=1,NUM
221     L=NUM-I+1
222     IF(KFLAG.EQ.2.AND.PLOTX(L).GT.0.)LDN=LDN+1
223     86 IF(LDN.EQ.1)CUTMAX=PLOTX(L-1)
224     85 CONTINUE
225     417 LU=7 $LNLGX=1 $LNLY=1 $NCLX=2 $NCLY=2 $WRITE(7,34)
226     34 FORMAT(*PLOT TITLE, 80 CHARACTERS MAXIMUM*)
227     READ(7,35)(LTITL(JM),JM=1,8)
228     35 FORMAT(8A10)
229     IF(IFLAG.EQ.1)LTITL2(1)=10#XZ STRESS-
230     IF(IFLAG.EQ.1)LTITL2(2)=10#NCRMALIZED
231     IF(IFLAG.EQ.2)LTITL2(1)=10#YZ STRESS-
232     IF(IFLAG.EQ.2)LTITL2(2)=10#NCRMALIZED

```

```

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 $YUP=SUP
239          IF(KFLAG.EQ.2.AND.YMIN.LT.AL)XRT=XCUT
240          IF(KFLAG.EQ.2.AND.YMAX.GT.C.)XLT=CUTMAX
241          IF(KFLAG.EQ.2.AND.YMIN.GE.AL)XRT=PLOTX(NUM)
242          AA=YUP$IF(YLO.EQ.AA)YUP=YUP+1.$IF(YLO.EQ.AA)YLC=YLC-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)XFT=-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(PLCTY,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)GOTC417
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          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  CCNTINUE
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 CF )Z NORMALIZED STRAIN*)
292 53 FORMAT(*MIN/MAX VALUES CF 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)=10FYZ 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=-AE(1)
307 301 WRITE(7,44) $READ(7,38)CHARAC
308 IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )CALL PLCTS(PLCTY,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 CCMPUTED*)
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 PLCTY(I)=AMATRX(I,3,2)/AMATRX(I,3,1)
344 L12=NUM/2 $STSMID=PLCTY(L12)
345 SUP=SCN=STSMID
346 DO 59 I=1,NUM
347 IF(PLCTY(I).GT.SUP)SUP=PLCTY(I)
348 59 IF(PLCTY(I).LT.SDN)SDN=PLCTY(I)

```

```

349     IF(IFLAG.EQ.1) WRITE(7,60) $IF(IFLAG.EQ.2)WRITE(7,61)
350     IF(KFLAG.EQ.1.AND.SET.NE.0.)WRITE(7,87)
351     87  FORMAT(*CAUTION---THIS IS NOT THE TOTAL DISPLACEMENT*)
352     60  FORMAT(*MIN/MAX VALUES OF W(X) VS. X*)
353     61  FORMAT(*MIN/MAX VALUES OF W(Y) VS. Y*)
354     WRITE(7,30) SDN,SUP
355     419 IF(IFLAG.EQ.1)LTITL2(1)=10HW(X) VS X
356     IF(IFLAG.EQ.2)LTITL2(1)=10HW(Y) VS Y
357     KY(1)=10*DISPLACEME      $KY(2)=10HNT
358     YLO=SDN $YUP=SUP $AA=YUP $IF(YLO.EQ.AA)YUP=YUP+1.
359     IF(YLO.EQ.AA)YLO=YLO-1.
360     WRITE(7,37) $READ(7,38)CHARAC
361     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO62
362     WRITE(7,40) $CALL GETNUM(AE) $YLC=AB(1) $YUP=AB(2)
363     62  WRITE(7,41) $READ(7,38)CHARAC
364     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H ) GOTC303
365     WRITE(7,43) $CALL GETNUM(AB)
366     IF(KFLAG.EQ.1)XLT=AB(1) $IF(KFLAG.EQ.1)XRT=AB(2)
367     IF(KFLAG.EQ.2)XLT=-AB(2) $IF(KFLAG.EQ.2)XRT=-AB(1)
368     303 WRITE(7,44) $READ(7,38)CHARAC
369     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H ) CALL PLCTS(PLOTX,PLOTY,1,NUM)
370     WRITE(7,46)$CALL GETNUM(AE)$LCHECK=AB(1)$IF(LCHECK.EQ.1)GOTO419
371     63  WRITE(7,45)SETMU1,SETMU2,SETMU3,AMUHOM,AM,AK
372     BB=B*1.E+06
373     WRITE(7,47)A,ALAMBD,D,BE
374     IF(IFLAG.EQ.1)CD1=2HXZ$IF(KFLAG.EQ.1)CD2=1HX
375     IF(IFLAG.EQ.2)CD1=2HYZ$IF(KFLAG.EQ.2)CD2=1FY
376     WRITE(7,49)CD1,CD2
377     45  FORMAT(*MU1 = *,E20.4,5X,*MU2 = *,E20.4,/,
378     1      *MU3 = *,E20.4,5X,*MUH = *,E20.4,/,
379     1      * M = *,E20.4,5X,* K = *,E20.4)
380     47  FORMAT(*DISLOCATION POSITION = *,F10.3,*, *,F10.3,* KMS.*,/,
381     1      *SLAB THICKNESS      = *,F10.3,* KMS.*,/,
382     1      *BURGERS VECTOR      = *,F10.3,* MMS.*)
383     49  FORMAT(A2,* STRESS/STRAIN IS. *,A2)
384     WRITE(7,64)
385     64  FORMAT(*0=RE-START*,/,*1=STOP*,/,
386     1      *2=RE-START WITH SAME VALUES, DIFFERENT PROFILE*)
387     CALL GETNUM(AB) $LJK=AB(1)
388     IF(LJK.EQ.0)GOTO48 $IF(LJK.EQ.2)GO TO 70
389     STOP $END
390     SUBROUTINE FUNC1(P,A,NU,D,G,ALAMBD,F)
391     R=(((P+A+(2.*NU*D))**2)+((G+ALAMBD)**2))
392     IF(R.EQ.0.)F=1.E-10$F=(P+A+(2.*NU*D))/R
393     RETURN $END
394     SUBROUTINE FUNC2(P,Q,F,S,A,NU,D,ALAMBD,F)
395     T=(((R+A+(2.*NU*D))**2)+((S+ALAMBD)**2))
396     IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
397     RETURN $END
398     SUBROUTINE FUNC3(P,Q,R,S,F)
399     T=(((R+S)**2)+((P+Q)**2))
400     IF(T.EQ.0.)T=1.E-10$F=(P+Q)/T
401     RETURN $END
402     SUBROUTINE GETNUM(K)
403     DIMENSION R(1),L(80)
404     READ(7,9)L $I=J=0
405     6   J=J+1 $N=P=S=0 $M=F=1
406     5   I=I+1 $IF(I.GT.80)RETURN $D=L(I) $K=4

```


LISTING OF SHRSTN

```

1  DELETE(LGO,OUTPUT,SHRSTN)
2  SHRSTN.
3  CXIT.
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 SHFSTN(TAPETTY=201, FILM=TAPETTY, TAPE7=TAPETTY)
13     COMMON/TVPCOL/TVPUL(8)
14     COMMON/TVTUNE/ITUNE(30)
15     COMMON/JPLCT/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), AMATRIX(200,3,2), AB(30)
19     DIMENSION PLOTY(200), PLCTX(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) $BGMU2=AB(3)*(10.**IAB4)
31     FNDMU2=AB(5)*(10.**IAB6)
32     WRITE(7,95)
33     95 FORMAT(*SHEAR MODULUS OF HOMOGENEOUS HALF-SPACE*)
34     CALL GETNUM(AB) $AMUHCM=AB(1)*(10.**AB(2))
35     WRITE(7,1F)
36     16 FORMAT(*POSITION OF DISLOCATION--IN KMS., (A, LAMBDA)*,/,
37     1      *NOTE---A>0., LAMBDA>0.*)
38     CALL GETNUM(AB) $A=AB(1) $ALAMBDA=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)

```

```

59     IF(KFLAG.EQ.1)WRITE(7,22)
60     IF(KFLAG.EQ.1)CALL GETNUM(AB)
61     IF(KFLAG.EQ.1)SET=-AB(1)
62     IF(LNUM.EQ.3)GOTO253 $IF(KFLAG.EQ.2)GOTO201
63     WRITE(7,21) $CALL GETNUM(AE) $XMIN=AB(1)
64     XMAX=AB(2) $GOTO202
65     201 WRITE(7,23) $CALL GETNUM(AE)
66     YMAX=-AB(1) $YMIN=-AB(2) $WRITE(7,24)
67     CALL GETNUM(AB) $SET=AB(1)
68     21  FORMAT(*ENTER MINIMUM AND MAXIMUM X VALUES, IN KMS.*)
69     22  FORMAT(*DEPTH OF PROFILE, IN KMS.*)
70     23  FORMAT(*ENTER MINIMUM AND MAXIMUM Y VALUES, IN KMS.*)
71     24  FORMAT(*X COORDINATE OF PR(FILE, IN KMS.*)
72     250  FORMAT(*1=COMPUTE VARIATION OF GLOBAL MINIMUM OF PROFILE*,/,
73     1      *2=COMPUTE VARIATION OF GLOBAL MAXIMUM OF PROFILE*,/,
74     1      *3=COMPUTE VARIATION OF STRESS AND STRAIN ONLY AT A *,/,
75     1      * POINT ON THE PROFILE*,/,
76     1      *4=COMPUTE VARIATION OF STRESS, STRAIN, AND DISPLACEMENT*
77     1      ,/,* AT A POINT ON THE PROFILE*)
78     202 CONTINUE
79     253  NUM=100 $IUPPER=20 $PI=3.14159
80     IF(KFLAG.EQ.1.AND.LNUM.NE.3)XINC=(XMAX-XMIN)/NUM
81     IF(KFLAG.EQ.2.AND.LNUM.NE.3)YINC=(YMAX-YMIN)/NUM
82     WRITE(7,25)NUM,IUPPER
83     25  FORMAT(*NOTE--NUMBER OF DATA POINTS = *,I5,/,
84     1      * UPPER LIMIT ON SERIES = *,I5)
85     AMU1=SETMU1$AMU2=ENDMU2$AMU3=SETMU3=SETMU1
86     AM=(AMU3-AMU2)/(AMU3+AMU2)$IF(AM.EQ.0.)AM=1.E-20
87     AK=(AMU1-AMU2)/(AMU1+AMU2)$IF(AK.EQ.0.)AK=1.E-20
88     BOUND=-D $AL=-ALAMBDA $AR=-A
89     IF(KFLAG.EQ.1 .AND.AL.GT.SET)GOTO74
90     GOTO75
91     74  WRITE(7,76)
92     76  FORMAT(*WARNING---PROFILE BELOW DISLOCATION LINE*,/,
93     1      * SOLUTION INVALID, TRY AGAIN*)
94     GOTO63
95     75  IF(LNUM.EQ.3)GOTO254
96     IF(KFLAG.EQ.1)GOTO90
97     RDIST=ABS(SET-(-A))
98     IF(RDIST.LT.YINC)WRITE(7,91)
99     IF(RDIST.LT.YINC)GOTO63
100    GOTO92
101    90  RDIST=ABS(SET-(-ALAMBDA))
102    IF(RDIST.LT.XINC)WRITE(7,92)
103    IF(RDIST.LT.XINC)GOTO63
104    91  FORMAT(*WARNING---PROFILE TOO CLOSE TO DISLOCATION CORE*,/,
105    1      * CHOOSE NEW PROFILE OR INCREASE GRID POINT DENSITY*)
106    92  CONTINUE
107    254  IF(KFLAG.EQ.1.AND.SET.GT.0.)WRITE(7,93)
108    IF(KFLAG.EQ.1.AND.SET.GT.0.)GOTO63
109    93  FORMAT(*WARNING---PROFILE ABOVE SURFACE OF HALF-SPACE*,/,
110    1      * SOLUTION INVALID, TRY AGAIN*)
111    DO 400 LJR=1,NUM
112    SETMU2=BEGINMU2+(((ENDMU2-BEGINMU2)/NUM)*(LJR-1))
113    SETMU3=SETMU1
114    IF(SETMU2.EQ.0.)SETMU2=1.E-10 $PLOT X(LJR)=ALOG10(SETMU2/SETMU1)
115    DO 230 K=1,2
116    IF(K.EQ.2)GOTO231

```



```

117      AMU1=AMU2=AMU3=AMUHOH
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), -ALAMEC, F2)
153      CALL FUNC1 (X(J), -A, I, D, Y(J), ALAMBD, F3)
154      CALL FUNC1 (X(J), -A, I, D, Y(J), -ALAMEC, 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, ALAMEC, F2)
167      CALL FUNC2 (Y(J), ALAMBD, X(J), Y(J), -A, NU, D, ALAMEC, F3)
168      CALL FUNC2 (Y(J), ALAMBD, X(J), Y(J), A, -NU, D, ALAMEC, 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

```

```

175      13 SUM1=SUM2=0.
176      DO 3 NU=1,IUPPER
177      CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
178      CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMED,F2)
179      CALL FUNC1(X(J),-A,NU,D,Y(J),ALAMEC,F3)
180      CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMEC,F4)
181      TERM1=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4)
182      SUM1=SUM1+TERM1
183      CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMEC,F5)
184      CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F6)
185      CALL FUNC1(X(J),-A,NU,D,Y(J),-ALAMED,F7)
186      CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMED,F8)
187      TERM2=((AM*AK)**NU)*(F5+(AK*F6)+((1./AK)*F7)+F8)
188      SUM2=SUM2+TERM2
189      3 CONTINUE
190      CALL FUNC3(X(J),A,Y(J),ALAMBD,F1)
191      CALL FUNC3(X(J),-A,Y(J),ALAMBD,F2)
192      CALL FUNC3(X(J),A,Y(J),-ALAMBD,F3)
193      CALL FUNC3(X(J),-A,Y(J),-ALAMBD,F4)
194      AMATRX(J,1,K)=((AMU2*B)/(2.*PI))*((F1+(AK*F2)+SUM1)-
195      1 (F3+(AK*F4)+SUM2))
196      GO TO 200
197      103 IF(IFLAG.EQ.2)GOTO14
198      SUM1=0.
199      DO 9 I=1,IUPPER
200      NU=I-1
201      CALL FUNC2(Y(J),ALAMEC,X(J),Y(J),A,-NU,D,ALAMEC,F1)
202      CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMEC,F2)
203      TERM=(F1+(AK*F2))*((AK*AM)**NU) §SUM1=SUM1+TERM
204      9 CCNTINUE
205      AMATRX(J,1,K)=((-AMU3*B)/(2.*PI))*(1.-AM)*SUM1 §GOTO200
206      14 SUM1=SUM2=0.
207      DO 11 I=1,IUPPER
208      NU=I-1
209      CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMEC,F1)
210      CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMED,F2)
211      TERM1=((AM*AK)**NU)*(F1+(AK*F2))
212      SUM1=SUM1+TERM1
213      CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMED,F1)
214      CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F2)
215      TERM2=((AM*AK)**NU)*(F1+(AK*F2)) §SUM2=SUM2+TERM2
216      11 CONTINUE
217      AMATRX(J,1,K)=((AMU3*B)/(2.*PI))*(1.-AM)*(SUM1-SUM2)
218      200 CONTINUE
219      230 CONTINUE
220      DO 204 I=1,INUM
221      IF(AMATRX(I,1,1).EQ.0.)AMATRX(I,1,1)=1.E-10
222      204 PLOTY(I)=AMATRX(I,1,2)/AMATRX(I,1,1)
223      STSMID=PLOTY(1) §SUP=SDN=STSMID
224      IF(LNUM.EQ.3)GOTO255 §IF(LNUM.EQ.4)GOTC257
225      GOTO258
226      257 DO 259 I=1,NUM
227      IF(KFLAG.EQ.1)RD=ABS(X(I)-COORD)
228      IF(KFLAG.EQ.1.AND.RD.LE.XINC)LI2=I
229      IF(KFLAG.EQ.2)RD=ABS(Y(I)-COORD)
230      259 IF(KFLAG.EQ.2.AND.RD.LE.YINC)LI2=I
231      STRSMX(LJR)=PLOTY(LI2) §GOTO255
232      258 DO 26 I=1,NUM

```

```

233     IF(PLOTY(I).GT.SUP)SUP=PLCTY(I)
234     26  IF(PLOTY(I).LT.SDN)SDN=FLCTY(I)
235     IF(LNUM.EQ.1)STRSMX(LJR)=SDN
236     IF(LNUM.EQ.2)STRSMX(LJR)=SLP
237     255  IF(LNUM.EQ.3)STRSMX(LJR)=SLP
238     DO 210 K=1,2
239     IF(K.EQ.2)GOTO211
240     AMU1=AMU2=AMU3=AMUHOM
241     GCTO212
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=PLCTY(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)=SLP
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=-ALAMB0+(GRD/100.) $GRD2=-ALAMB0-(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  CCNTINUE
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=AMATFX(KMID,3,K)
288     DO 73 *I=1,NUM
289     73  AMATRX(I,3,K)=AMATRX(I,3,K)-STSMID
290     58  CCNTINUE

```

```

291 DO 215 I=1,NUM
292 IF (AMATRX(I,3,1).EQ.0.)AMATRX(I,3,1)=1.E-10
293 215 PLOTY(I)=AMATRX(I,3,2)/AMATRX(I,3,1)
294 STSMID=PLOTY(1) $SUP=SDN=STSMID
295 IF (LNUM.EQ.4)DISPMX(LJR)=PLOTY(LI2)
296 IF (LNUM.EQ.4)GOTO260
297 DO 59 I=1,NUM
298 IF (PLOTY(I).GT.SUP)SUP=PLCTY(I)
299 59 IF (PLCTY(I).LT.SDN)SDN=PLCTY(I)
300 IF (LNUM.EQ.1)DISPMX(LJR)=SDN
301 260 IF (LNUM.EQ.2)DISPMX(LJR)=SLP
302 400 CONTINUE
303 STSMID=STRSMX(1)
304 SUP=SDN=STSMID
305 DO 28 I=1,NUM
306 IF (STRSMX(I).GT.SUP)SUP=STRSMX(I)
307 28 IF (STRSMX(I).LT.SDN)SDN=STRSMX(I)
308 WRITE(7,29)SDN,SUP
309 29 FORMAT(*MIN/MAX VALUES OF NORMALIZED STRESS*,/,
310 1 *VERSUS SHEAR MODLLS CONTRAST*,/,
311 1 E20.4, 5X, E20.4)
312 417 LU=7 $LNLGX=1$LNLGY=1$NCLX=2$NCLY=2$WRITE(7,34)
313 34 FORMAT(*PLCT TITLE, 80 CHARACTERS MAXIMUM*)
314 READ(7,35)(LTITL(J),JM=1,8)
315 35 FORMAT(8A10)
316 IF (IFLAG.EQ.1)LTITL2(1)=10+XZ STRESS-
317 IF (IFLAG.EQ.2)LTITL2(2)=10+YZ STRESS-
318 LTITL2(2)=10HNORMALIZED
319 DO 36 I=3,8
320 36 LTITL2(I)=10H
321 KX(1)=10HLCG(MU2/MU $KY(1)=10HNORMALIZED
322 KX(2)=10H1) $KY(2)=10H STRESS
323 XRT=PLOTX(NUM) $XLT=PLOTX(1)
324 MAJX=5$MAJY=10$YLO=SDN$YUP=SUP$AA=YUP
325 IF (YLO.EQ.AA)YUP=YUP+1$IF (YLO.EQ.AA)YLC=YLC-1
326 WRITE(7,37)
327 37 FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
328 READ(7,38)CHARAC
329 38 FORMAT(A1)
330 IF (CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO39
331 WRITE(7,40)
332 40 FORMAT(*MIN/MAX Y VALUES*)
333 CALL GETNUM(AB) $YLO=AB(1)$YUP=AB(2)
334 39 WRITE(7,41)
335 41 FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
336 READ(7,38)CHARAC
337 IF (CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO42
338 WRITE(7,43)
339 43 FORMAT(*MIN/MAX X VALUES*)
340 CALL GETNUM(AB)$XLT=AB(1)$XRT=AB(2)
341 42 WRITE(7,44)
342 44 FORMAT(*SKIP?*)
343 READ(7,38)CHARAC
344 IF (CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )CALL PLOTS(STRSMX,PLOTX,1,NUM)
345 WRITE(7,46)
346 46 FORMAT(*0=RE-START, 1=NEW FLOT*)
347 READ(7,71)LCHECK
348 71 FORMAT(I1)

```

```

349     IF(LCHECK.EQ.0)GOTO63 $IF(LCHECK.EQ.1)GOTO417
350     STSMID=PLOTY(1) $SUP=SDN=STSMID
351     DC 52 I=1,NUM
352     IF(STRNMX(I).GT.SUP)SUP=STRNMX(I)
353     52 IF(STRNMX(I).LT.SDN)SDN=STRNMX(I)
354     IF(IFLAG.EQ.1)CD=2HXZ $IF(IFLAC.EQ.2)CD=2HYZ
355     WRITE(7,53)CD,SDN,SUP
356     53 FORMAT(*MIN/MAX VALUES OF NORMALIZED *,A2,* STRAIN*,/,
357     1      *VERSUS SHEAR MOCLLUS CONTRAST*,/,E20.4,5X,E20.4)
358     418 KY(2)=10H STRAIN $YLO=SDN $YUP=SUP$AA=YUP
359     IF(YLO.EQ.AA)YUP=YUP+1 $IF(YLO.EQ.AA)YLC=YLO-1
360     IF(IFLAG.EQ.1)LTITL2(1)=10FXZ STRAIN-
361     IF(IFLAG.EQ.2)LTITL2(1)=10HYZ STRAIN-
362     WRITE(7,37)$READ(7,38)CHARAC
363     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO550
364     WRITE(7,40)$CALL GETNUM(AB)$YLO=AB(1)$YUP=AB(2)
365     550 WRITE(7,41)$READ(7,38)CHARAC
366     IF(CHARAC.EQ.1HN .OR.CHARAC.EQ.1H )GOTO55
367     WRITE(7,43)$CALL GETNUM(AB)$XLT=AB(1)$XRT=AB(2)
368     55 WRITE(7,44)$READ(7,38)CHARAC
369     IF(CHARAC.EQ.1HN .OR.CHARAC.EQ.1H )CALL PLCTS(STRNMX,PLOTX,1,NUM)
370     56 WRITE(7,46)$CALL GETNUM(AB)$LCHECK=AB(1)
371     IF(LCHECK.EQ.0)GOTO63$IF(LCHECK.EQ.1)GOTO418
372     IF(KFLAG.NE.IFLAG)WRITE(7,57)
373     57 FORMAT(*NOTE--ONLY W(X) VS. X OR W(X) VS. Y IS COMPUTED*)
374     IF(KFLAG.NE.IFLAG)GOTO63
375     STSMID=D,ISPMX(1) $IF(LNUM.EQ.3)GOTO63
376     SUP=SDN=STSMID
377     DC 45 I=1,NUM
378     IF(DISPMX(I).GT.SUP)SUP=DISPMX(I)
379     45 IF(DISPMX(I).LT.SDN)SDN=DISPMX(I)
380     WRITE(7,60)CD,SDN,SUP
381     60 FORMAT(*MIN/MAX VALUES OF NORMALIZED DISPLACEMENT*,/,
382     1      *VERSUS SHEAR MOCLLUS CONTRAST, COMPUTED*,/,
383     1      *FROM *,A2,* STRAIN*,/,E20.4,5X,E20.4)
384     419 KY(1)=10HDISPLACEME $KY(2)=10HNT
385     YLC=SDN$YUP=SUP$AA=YUP$IF(YLO.EQ.AA)YUP=YUP+1
386     IF(YLO.EQ.AA)YLC=YLO-1$WRITE(7,37)$READ(7,38)CHARAC
387     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO620
388     WRITE(7,40)$CALL GETNUM(AB) $YLO=AB(1)$YUP=AB(2)
389     620 WRITE(7,41)$READ(7,38)CHARAC
390     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )GOTO62
391     WRITE(7,43)$CALL GETNUM(AB)$XLT=AB(1)$XRT=AB(2)
392     62 WRITE(7,44)$READ(7,38)CHARAC
393     IF(CHARAC.EQ.1HN.OR.CHARAC.EQ.1H )CALL PLOTS(DISPMX,PLCTX,1,NUM)
394     WRITE(7,46)$CALL GETNUM(AB)$LCHECK=AB(1)
395     IF(LCHECK.EQ.1)GOTO419
396     63 WRITE(7,450)SETMU1,BGNMU2,ENDMU2,SETMU3
397     WRITE(7,451)AMUHOM,AM,AK
398     IF(IFLAG.EQ.1)CD1=2HXZ $IF(KFLAG.EQ.1)CD2=1HX
399     IF(IFLAG.EQ.2)CD1=2HYZ $IF(KFLAG.EQ.2)CD2=1HY
400     WRITE(7,49)CD1,CD2
401     450 FORMAT(*MU1 = *,E20.4,5X,*MU2LOW = *,E20.4,/,
402     1      *MU2HIGH = *,E20.4,5X,*MU3 = *,E20.4)
403     451 FORMAT(*MUH = *,E20.4,5X,*M = *,E20.4,/,
404     1      *K = *,E20.4)
405     BB=B*1.E+06
406     WRITE(7,47)A,ALAMBDA,D,RR

```


LISTING OF 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 EOP
12     PROGRAM DBLNRM(TAPETTY=201,FILM=TAPETTY,TAPE7=TAPETTY)
13     COMMON/TVPOCL/TVPUL(8)
14     COMMON/TVTUNE/ITUNF(30)
15     COMMON/JPLOT/XLT,XFT,YLO,YOP,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,IFFT,-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 FOPMAT(*POSITION OF DISLOCATION--IN KMS., (A,LAMBDA)*,/,
36     1     *NOTE---A>0., LAMEDA>0.*)
37     CALL GETNUM(AB) $A=AB(1) $LAMBD=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=-ALAMBDA $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-(-ALAMBDA))
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)ALAMBDA=ALAMBDA-WIDTH
99 IF(K.EQ.2)B=-B
100 DO 910 MK=1,2
101 IF(MK.EQ.2)ALAMBDA=ALAMBDA+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

```

```

117 CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
118 TERM1=((AM*AK)**NU)*F1
119 CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,I,D,ALAMBD,F2)
120 TERM2=(AM**I)*(AK**NU)*F2
121 SUM1=SUM1+TERM1 $SUM2=SUM2+TERM2
122 1 CONTINUE
123 SYZ(J)=(((AMU1*B)/(2.*PI))*(1.-AK))*(SUM1+SUM2) $GOTO200
124 12 SUM1=SUM2=SUM3=SUM4=0.
125 DO 5 I=1,IUPPF.
126 NU=I-1
127 CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
128 CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMBD,F2)
129 CALL FUNC1(X(J),-A,I,D,Y(J),ALAMBD,F3)
130 CALL FUNC1(X(J),-A,I,D,Y(J),-ALAMBD,F4)
131 TERM1=((AM*AK)**NU)*F1 $TERM2=((AM*AK)**NU)*F2
132 TERM3=(AM**I)*(AK**NU)*F3 $TERM4=(AM**I)*(AK**NU)*F4
133 SUM1=SUM1+TERM1 $SUM2=SUM2+TERM2 $SUM3=SUM3+TERM3
134 SUM4=SUM4+TERM4
135 5 CONTINUE
136 SYZ(J)=((AMU1*B)*(1.-AK)/(2.*PI))*(SUM1+SUM3-SUM2-SUM4)
137 GOTO200
138 102 IF(IFLAG.EQ.2)GOTO13
139 SUM1=0.
140 DO 7 NU=1,IUPPF.
141 CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,NU,D,ALAMBD,F1)
142 CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMBD,F2)
143 CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,NU,D,ALAMBD,F3)
144 CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMBD,F4)
145 TERM=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4) $SUM1=SUM1+TERM
146 7 CONTINUE
147 CALL FUNC3(Y(J),ALAMBD,X(J),A,F1)
148 CALL FUNC3(Y(J),ALAMBD,X(J),-A,F2)
149 TERM2=F1+(AK*F2)
150 SYZ(J)=((-AMU2*B)/(2.*PI))*(TERM2+SUM1) $GOTO200
151 13 SUM1=SUM2=0.
152 DO 3 NU=1,IUPPF.
153 CALL FUNC1(X(J),A,NU,D,Y(J),ALAMBD,F1)
154 CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMBD,F2)
155 CALL FUNC1(X(J),-A,NU,D,Y(J),ALAMBD,F3)
156 CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMBD,F4)
157 TERM1=((AM*AK)**NU)*(F1+(AK*F2)+((1./AK)*F3)+F4)
158 SUM1=SUM1+TERM1
159 CALL FUNC1(X(J),A,NU,D,Y(J),-ALAMBD,F5)
160 CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F6)
161 CALL FUNC1(X(J),-A,NU,D,Y(J),-ALAMBD,F7)
162 CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMBD,F8)
163 TERM2=((AM*AK)**NU)*(F5+(AK*F6)+((1./AK)*F7)+F8)
164 SUM2=SUM2+TERM2
165 3 CONTINUE
166 CALL FUNC3(X(J),A,Y(J),ALAMBD,F1)
167 CALL FUNC3(X(J),-A,Y(J),ALAMBD,F2)
168 CALL FUNC3(X(J),A,Y(J),-ALAMBD,F3)
169 CALL FUNC3(X(J),-A,Y(J),-ALAMBD,F4)
170 SYZ(J)=((AMU2*B)/(2.*PI))*((F1+(AK*F2)+SUM1)-(F3+(AK*F4)+SUM2))
171 GO TO 200
172 103 IF(IFLAG.EQ.2)GOTO14
173 SUM1=0.
174 DO 9 I=1,IUPPF.

```

```

175     NU=I-1
176     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),A,-NU,D,ALAMBD,F1)
177     CALL FUNC2(Y(J),ALAMBD,X(J),Y(J),-A,-NU,D,ALAMBD,F2)
178     TERM=(F1+(AK*F2))*((AK*AM)**NU) $SUM1=SUM1+TERM
179     9 CONTINUE
180     SXZ(J)=((-AMU3*B)/(2.*PI))*(1.-AM)*SUM1 $GCTC200
181     14 SUM1=SUM2=0.
182     DO 11 I=1,IUPPER
183     NU=I-1
184     CALL FUNC1(X(J),A,-NU,D,Y(J),ALAMBD,F1)
185     CALL FUNC1(X(J),-A,-NU,D,Y(J),ALAMBD,F2)
186     TERM1=((AM*AK)**NU)*(F1+(AK*F2))
187     SUM1=SUM1+TERM1
188     CALL FUNC1(X(J),A,-NU,D,Y(J),-ALAMBD,F1)
189     CALL FUNC1(X(J),-A,-NU,D,Y(J),-ALAMBD,F2)
190     TERM2=((AM*AK)**NU)*(F1+(AK*F2)) $SUM2=SUM2+TERM2
191     11 CONTINUE
192     SYZ(J)=(-AMU3*B)/(2.*PI)*(1.-AM)*(SUM1-SUM2)
193     200 CONTINUE
194     DO 920 IMK=1,NUM
195     IF(IFLAG.EQ.1)CM(IMK,MK,K)=SXZ(IMK)
196     920 IF(IFLAG.EQ.2)CM(IMK,MK,K)=SYZ(IMK)
197     910 CONTINUE
198     230 CONTINUE
199     DO 930 I=1,NUM
200     IF(IFLAG.EQ.1)SXZ(I)=(CM(I,1,2)+CM(I,2,2))/(CM(I,1,1)+CM(I,2,1))
201     930 IF(IFLAG.EQ.2)SYZ(I)=(CM(I,1,2)+CM(I,2,2))/(CM(I,1,1)+CM(I,2,1))
202     IF(IFLAG.EQ.1)STSMID=SXZ(1)
203     IF(IFLAG.EQ.2)STSMID=SYZ(1)
204     SUP=SDN=STSMID
205     DO 26 I=1,NUM
206     IF(IFLAG.EQ.1)GOTO27
207     IF(SYZ(I).GT.SUP)SUP=SYZ(I)
208     IF(SYZ(I).LT.SDN)SDN=SYZ(I)
209     GO TO 26
210     27 IF(SXZ(I).GT.SUP)SUP=SXZ(I)
211     IF(SXZ(I).LT.SDN)SDN=SXZ(I)
212     26 CONTINUE
213     IF(IFLAG.EQ.1)WRITE(7,28) $IF(IFLAG.EQ.2)WRITE(7,29)
214     28 FORMAT(*MIN/MAX VALUES OF NORMALIZED XZ STRESS*)
215     29 FORMAT(*MIN/MAX VALUES OF NORMALIZED YZ STRESS*)
216     WRITE(7,30)SDN,SUP
217     30 FORMAT(E15.4,5X,E15.4)
218     LUP=0
219     DO 31 I=1,NUM
220     IF(IFLAG.EQ.1)PLOTY(I)=SXZ(I)
221     IF(IFLAG.EQ.2)PLOTY(I)=SYZ(I)
222     IF(KFLAG.EQ.1)PLOTX(I)=X(I)
223     31 IF(KFLAG.EQ.2)PLOTX(I)=Y(I)
224     IF(KFLAG.EQ.2)GOTO83
225     GOTO85
226     83 DO 81 I=1,NUM
227     SXZ(I)=PLOTY(I)
228     81 SYZ(I)=PLOTX(I)
229     DO 84 I=1,NUM
230     L=NUM-I+1
231     PLOTY(L)=SXZ(I)
232     84 PLOTX(L)=SYZ(I)

```

```

233      DO 82 I=1,NUM
234      IF (KFLAG.EQ.2.AND.PLOTX(I).LT.AL) LUP=LUP+1
235      82 IF (LUP.EQ.1) XCLT=PLOTX(I-1)
236      LDN=0
237      DO 86 L=1,NUM
238      L=NUM-I+1
239      IF (KFLAG.EQ.2.AND.PLOTX(L).GT.0.) LDN=LDN+1
240      86 IF (LDN.EQ.1) CUTMAX=PLOTX(L-1)
241      85 CONTINUE
242      417 LU=7  $LNLCX=1  $LNLCY=1  $INCLX=2  $INCLY=2  $WRITE(7,34)
243      34  FORMAT(*PLOT TITLE, 80 CHARACTERS MAXIMUM*)
244      READ(7,35) (LTITL(JM),JM=1,8)
245      35  FORMAT(8A10)
246      IF (IFLAG.EQ.1) LTITL2(2)=10H XZ STRESS
247      IF (IFLAG.EQ.2) LTITL2(2)=10H YZ STRESS
248      LTITL2(1)=10HNORMALIZED
249      DO 36 I=1,8
250      36  LTITL2(I)=10H
251      IF (KFLAG.EQ.1) KX(1)=10HX (KMS.)
252      IF (KFLAG.EQ.2) KY(1)=10HY (KMS.)
253      KX(2)=10H          $KY(1)=10HNORMALIZED  $KY(2)=10H STRESS
254      XRT=PLOTX(NUM)  $XLT=PLOTX(1)  $MAJX=5  $MAJY=10  $YLC=SDN  $YUP=SUP
255      IF (KFLAG.EQ.2.AND.YMIN.LT.AL) XRT=XCUT
256      IF (KFLAG.EQ.2.AND.YMAX.GT.0.) XLT=CUTMAX
257      IF (KFLAG.EQ.2.AND.YMIN.GT.AL) XRT=PLOTX(NUM)
258      AA=YUP  $IF (YLO.EQ.AA) YUP=YUP+1.  $IF (YLO.EQ.AA) YLC=YLC-1.
259      WRITE(7,37)
260      37  FORMAT(*SET VERTICAL SCALE? Y OR N(=BLANK)*)
261      READ(7,38) CHARAC
262      38  FORMAT(A1)
263      IF (CHARAC.EQ.1HN.OF.CHARAC.EQ.1H ) GOTO39
264      WRITE(7,40)
265      40  FORMAT(*MIN/MAX Y VALUES*)
266      CALL GETNUM(AB)  $YLO=AB(1)  $YUP=AB(2)
267      39  WRITE(7,41)
268      41  FORMAT(*SET HORIZONTAL SCALE? Y OR N(=BLANK)*)
269      READ(7,38) CHARAC
270      IF (CHARAC.EQ.1HN.OF.CHARAC.EQ.1H ) GOTO42
271      WRITE(7,43)
272      43  FORMAT(*MIN/MAX X VALUES*)
273      CALL GETNUM(AB)  $IF (KFLAG.EQ.1) XLT=AB(1)
274      IF (KFLAG.EQ.1) XRT=AB(2)
275      IF (KFLAG.EQ.2) XLT=-AB(2)
276      IF (KFLAG.EQ.2) XRT=-AB(1)
277      42  WRITE(7,44)
278      44  FORMAT(*SKIP?*)
279      READ(7,38) CHARAC
280      IF (CHARAC.EQ.1HN.OF.CHARAC.EQ.1H ) CALL PLOTS(PLOTX,PLOTY,1,NUM)
281      WRITE(7,46)
282      46  FORMAT(*0=RE-START, 1=NEW PLOT*)
283      READ(7,71) LCHECK
284      71  FORMAT(I1)
285      IF (LCHECK.EQ.0) GOTO63  $IF (LCHECK.EQ.1) GOTO417
286      DO 210 K=1,2
287      IF (K.EQ.2) GOTO211
288      AMU1=AMU2=AMU3=AMUFORM
289      GOTO212
290      211  AMU1=SETAMU1  $AMU2=SETAMU2  $AMU3=SETAMU3

```

```

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(AE) $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(AE)
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(AE) $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.*SYZ(I)*DINC*1.E-06 $TERM2=2.*SXZ(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=-ALAMB0+(GRD/100.) $GRD2=-ALAMB0-(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

```