Jour. Fac. Sci., Hokkaido Univ., Ser. VII (Geophysics), Vol. 10, No. 2, 295-310, 1997.

A Computer Program for the Calculation of Geomagnetic Field Changes due to an Inclined Rectangular Fault in a Half Space

Mitsuru Utsugi

Division of Earth and Planetary Sciences, Graduate School of Science, Hokkaido University, Sapporo 060, Japan

(Received December 9, 1996)

Abstract

It has been reported that the geomagnetic changes were accompanied with seismic activities. To estimate the seismomagnetic effect, Sasai (1991) presented the analytical solution of piezomagnetic potential in case of a vertical faulting. We expanded Sasai's method to an inclined faulting with arbitrary dip angle and developed a FORTRAN program to calculate the geomagnetic field changes.

1. Introduction

We established an analytic form of seismomagnetic effect due to an inclined rectangular fault with uniform slip (Utsugi et al., in preparation). As the mathematical expression is somewhat complicated, we present the FORTRAN program MGCAL for the sake of user's convenience. The program allows us to calculate the geomagnetic field changes in the northward, eastward and downward components, as well as the total force.

2. Formulation of piezomagnetic potential

2.1. Piezomagnetic potential due to an inclined fault

In the present section, we show an expression of the piezomagnetic potential due to an inclined fault briefly. Sasai (1991) formulated the elementary piezomagnetic potential ω_{lm}^{k}, i.e. piezomagnetic potential due to a point dislocation source. ω_{lm}^{k} is given by the integral over the surface of magnetized region of the medium ;

$$\omega_{lm}^{k}(\boldsymbol{r}, \boldsymbol{\xi})=C_k\iint_S\left[\boldsymbol{S}_{lm}^{k}(\boldsymbol{r}', \boldsymbol{\xi})\cdot\boldsymbol{n}\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}+\tau_{lm}^{k}(\boldsymbol{r}', \boldsymbol{\xi})\frac{\partial}{\partial n}\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}\right]dS_{r'} \quad (1)$$

$$C_k=\frac{1}{2}\beta J_k\mu\frac{3\lambda+2\mu}{\lambda+\mu}, \quad (2)$$

where $\boldsymbol{\xi}$ and $\boldsymbol{r}$ denote source point and observation point, respectively. τ_{lm}^{k} is the k-th component of displacement vector produced by point dislocation source of type (l, m). S_{lm}^{k} is the stress-induced magnetization vector. λ and μ are the Lamé constants, β is the stress sensitivity, J_k is the k-th component of initial magnetization of crustal rock, and C_k is the k-th component of seismomagnetic moment produced by the piezomagnetic effect.

For the finite fault, we can obtain the piezomagnetic potential as the following integral over the fault plain Σ ;

$$W^k(\boldsymbol{r})=\iint_\Sigma \Delta u_l(\boldsymbol{\xi})\cdot\omega_{lm}^{k}(\boldsymbol{r}, \boldsymbol{\xi})\cdot\nu_m(\boldsymbol{\xi})d\Sigma_\xi, \quad (3)$$

where W^k indicates the piezomagnetic potential produced by k-th component of initial magnetization of crustal rock, Δu_l is the dislocation vector (its l-th component) and ν_m is a unit vector normal to the fault plain (its m-th component).

For an inclined rectangular fault (length L, width W, depth d and dip-angle δ) with uniform slip (U_1, U_2, U_3) (Fig. 1), eq. (3) can be written in the following form ;

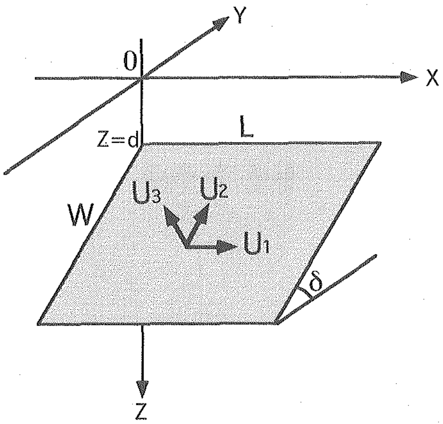

Fig. 1. Geometry of the fault model. Positive U_1, U_2 and U_3 with $0<\delta<\pi/2$ indicate left-lateral slip, thrusting slip and tensile opening, respectively.

$$W^k(r) = U_1 \cdot S^k(r) + U_2 \cdot D^k(r) + U_3 \cdot T^k(r) \tag{4}$$

where S^k, D^k and T^k indicate the piezomagnetic potentials due to strike-slip, dip-slip and tensile opening faulting, respectively. These potentials can be written by the integral of ω_{lm}^k over the fault plain (Utsugi et al., in preparation).

2.2. *Earth model*

We assume the earth as a homogeneous and isotropic elastic half-space having a uniformly magnetized upper layer with a constant piezomagnetic stress sensitivity. We also assume the observation point is outside the medium.

We introduce a Cartesian coordinate as shown in Fig. 2, where X, Y and Z axes are directed east, north and downward, respectively. A point above the left edge of the fault is taken to be the origin of the coordinate as shown in Fig. 2. X' and Y' are parallel and perpendicular axes to the strike direction of the fault.

A semi-infinite elastic medium occupies $Z \geq 0$ and magnetized region is limited within a layer $0 \leq Z \leq H$, where H is the depth of the Curie point isotherm. Then we have to calculate ω_{lm}^k of eq. (1) over the free-surface (S_0), the fault plain (Σ) and the plain $Z=H(S_H)$. Therefore, W^k of eq. (3) can be written by the summation of the contributions from S_0, Σ and S_H. As W^k

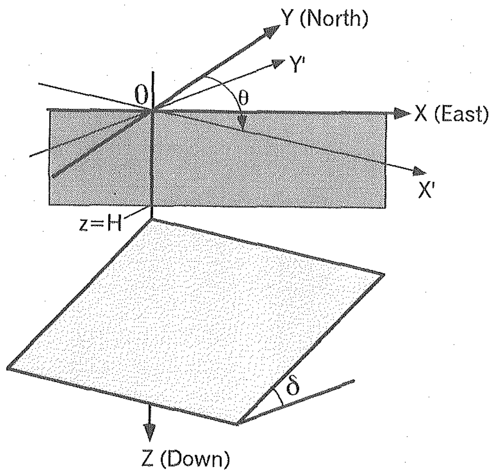

Fig. 2. Coordinate system. X, Y and Z axis direct eastward, northward and downward, respectively. θ and H mean strike-angle and depth of Curie point isotherm, rspectively.

depends on the fault depth, we write W^k as the following form ;

$$W^k = W^k_{S_0+\Sigma} + W^k_{H0} + \begin{cases} W^k_{HI} & (d + W \sin\delta > H) \\ W^k_{HII} & (d < H) \end{cases}$$

For this earth model, we can obtain the analytical solution of W^k as well as S^k, D^k and T^k. The geomagnetic changes can be obtained by differentiating W^k with respect to X, Y and Z. The analytical solution of geomagnetic changes are listed in Utsugi et al. (in preparation). As these solutions become singular when dip angle $\delta = \pi/2$ (i.e. vertical fault), we cannot calculate geomagnetic changes. In this case, we should use the solution for vertical faulting presented by Sasai (1991).

3. Program

3.1. *Design of the program*

The program MGCAL is designed for the analytical calculation of geomagnetic field changes due to an inclined rectangular fault. This program consists of a main program and 14 subroutines. The complete source list is given in Appendix.

The main program MGCAL calls subroutine CALCON1 and PIEZ. CALCON1 calculates seismomagnetic moment and constants with respect to medium included in S^k, D^k and T^k. The geomagnetic changes are calculated by subroutine PIEZ. This subroutine calls subroutine DIFCOEF which calculate the geomagnetic changes in the coordinate system (X', Y', Z) (Fig. 2). These geomagnetic changes are transformed in the coordinate system (X, Y, Z) in subroutine PIEZ. The subroutine DIFCOEF calls following subroutines ;

DERIV Calculate constants which are needed to calculate geomagnetic changes.

STRIKE_0, DIP_0, TENSILE_0 Calculate geomagnetic changes produced by free-surface and fault plain due to strike, dip and tensile-opening faulting ($S_0 + \Sigma$).

STRIKE_H0, DIP_H0, TENSILE_H0 Calculate geomagnetic changes produced by Curie surface due to strike, dip and tensile-opening faulting (H0).

STRIKE_H, DIP_H, TENSILE_H Calculate geomagnetic changes produced by Curie surface due to strike, dip and tensile-opening faulting (HI+HII).

3.2. Process

In our program, we can chose the calculation point: the grid point or arbitrary point. Prior to execution of this program, we have to make a parameter file. The format of the parameter file is as follows;

Calculation at grid points;	Calculation at arbitrary points;
IFLG	IFLG
NDX NDY XX0 YY0	NUMB
DX DY	U1 U2 U3 FL FW DPT STR DIP
Z	POI AMU CMZ GMINC GMDEC HH
U1 U2 U3 FL FW DPT STR DIP	BETA
POI AMU CMZ GMINC GMDEC HH	X Y Z
BETA	

The meaning of these parameters are described in Table 1 and 2. When our program is executed, firstly we have to chose the calculation point:
"CALCULATE AT GRID POINTS OR ARBITRARY POINTS?"
"TYPE GRID POINTS=1 OR ARBITRARY POINTS=2"

Next, the parameter file name and output file name are requested:

"PARAMETER FILE NAME IS=?"
"OUTPUT FILE NAME IS=?"

Then the output file is made: geomagnetic field changes are output with the coordinate of calculation point. The parameter "IFLG" is prepared for choice of geomagnetic field components (eastward-, northward-, downward-component or total force) (see Table 1).

Table 1. The roles of input parameter IFLG.

IFLG	Designates output values.
=1	Output eastward component of geomagnetic field change.
=2	Output northward component of geomagnetic field change.
=3	Output downward component of geomagnetic field change.
=4	Output total force of geomagnetic field change.

Table 2. The meaning of input parameters.

In the case of the calculation at grid points:

NDX, NDY		Grid number.
DX, DY		Sample interval of X and Y axis.
XX0, YY0		Coordinate of start point of grid.
Z	(km)	Z-coordinate of observation points ($Z \leq 0$).

In the case of the calculation at arbitrary points:

NUMB		Number of arbitrary points.
X, Y, Z	(km)	Coordinates of observation points.

Fault parameters and medium parameters:

U1, U2, U3	(m)	Components of the displacement vector (U_1, U_2, U_3).
FL, FW	(km)	Fault length L and width W.
STR	(degree)	Strike angle of the fault θ.
DIP	(degree)	Dip angle of the fault δ ($0 \leq \delta < \pi/2$).
DPT	(km)	Depth of the upper boundary of the fault d.

POI		Poisson ratio.
AMU	(cgs)	Lamé constant μ.
CMZ	(A/m)	Initial magnetization of crustal rock $\|\boldsymbol{J}\|$.
GMDEC	(degree)	Magnetic declination.
GMINC	(degree)	Magnetic inclination.
HH	(km)	Depth of the Curie point isotherm H.
BETA	(bar^{-1})	Stress sensitivity β.

Acknowledgement

I would like to express my thank to Prof. Y. Nishida, Prof. Y. Sasai and Mr. H. Oshima for their valuable comments.

References

Sasai Y., 1991. Tectonomagnetic modeling on the basis of the linear piezomagnetic effect. Bull. Earth. Res. Inst., Univ. Tokyo, **66**, 585-722.

Appendix

```
C
      PROGRAM MGCAL
C**********************************************************
C
C      GEOMAGNETIC FIELD CHANGE AT OUTSIDE THE MEDIUM
C          DUE TO AN INCLINED RECTANGULAR FAULT
C               IN A SEMIINFINITE MEDIUM
C             CODED BY M. UTSUGI....DEC 1996
C
C************************************************************
C**** INPUT PARAMETERS
C     NDX,NDY      : GRID NUMBER (X,Y)
C     DX,DY        : SAMPLE INTERVAL (X,Y)
C     NUMB         : NUMBER OF ARBITRARY POINTS
C     Z            : Z-COORDINATE OF OBS. POINT
C
C     IFLG =1..OUTPUT EASTWARD 2..NORTHWARD 3..DOWNWARD COMPONENT
C           4..TOTAL FORCE OF GEOMAGNETIC FIELD CHANGE
C     U1,U2,U3     : DISLOCATION VECTOR(M)
C                          U1..STRIKE-SLIP
C                          U2..DIP-SLIP
C                          U3..TENSILE-OPENING
C     DIP,STR      : FAULT DIP AND STRIKE ANGLE (DEGREE)
C     FL,FW,DPT    : FAULT LENGTH, WIDTH AND DEPTH(KM)
C     POI,AMU      : POISSON RATIO AND LAME CONSTANT MU(CGS)
C     BETA         : STRESS SENSITIVITY
C     HH           : CURIE DEPTH(KM)
C     CMZ          : CRUSTAL MAGNETIZATION(A/M)
C     GMINC,GMDEC : GEOMAGNETIC INCLINATION AND DECRINATION(DEG.)
C***** OUTPUT VALUES
C     DMX,DMY,DMZ,DMF : EAST, NORTH, DOWNWARD COMPONENTS
C                       AND TOTAL FORCE OF GEOMAGNETIC
C                       FIELD CHANGE (NT)
C
      INTEGER I,J,K,INDX,NDX,NDY,NUMB,IFLG
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SITX(50),SITY(50),SITZ(50)
      CHARACTER*20 INFILE1,INFILE2
C
      WRITE(6,*) 'CALCULATE AT GRID POINTS OR ARBITRARY POINTS'
      WRITE(6,*) 'TYPE GRID POINTS=1 OR ARBITRARY POINTS=2'
      READ(5,*) INDX
      WRITE(6,*) 'PARAMETER FILE NAME=?'
      READ(5,*) INFILE1
      WRITE(6,*) 'OUTPUT FILE NAME=?'
      READ(5,*) INFILE2
C
      OPEN(10,FILE=INFILE1,STATUS='OLD')
      OPEN(20,FILE=INFILE2,STATUS='NEW')
      IF(INDX.EQ.1) GOTO 100
      IF(INDX.EQ.2) GOTO 200
C
 100  READ(10,*) IFLG
      READ(10,*) NDX,NDY
      READ(10,*) Z
      READ(10,*) DX,DY,XX0,YY0
      READ(10,*) U1,U2,U3,FL,FW,DPT,STR,DIP
      READ(10,*) POI,AMU,CMZ,GMINC,GMDEC,HH,BETA
      CLOSE(10)
      IF(Z.GT.0.D0) THEN
         WRITE(6,*) 'POSITIV Z CANNOT BE ACCEPTED IN THIS PROGRAM'
         GOTO 400
      ENDIF
      IF(DIP.EQ.90.D0) THEN
         WRITE(6,*) 'DIP=90 CANNOT BE ACCEPTED IN THIS PROGRAM'
         GOTO 400
      ENDIF
      STR=90.D0-STR
      CALL CALCON1(POI,AMU,BETA,CMZ,DIP,STR,GMINC,GMDEC)
C
      X=XX0
      DO 120 I=1,NDX+1
         Y=YY0
         DO 110 J=1,NDY+1
            CALL PIEZ(X,Y,Z,U1,U2,U3,DIP,STR,FL,FW,DPT,POI,
     &           AMU,BETA,HH,CMZ,GMINC,GMDEC,DMX,DMY,DMZ,DMF)
            IF(IFLG.EQ.1) WRITE(20,310) X,Y,DMX
            IF(IFLG.EQ.2) WRITE(20,310) X,Y,DMY
            IF(IFLG.EQ.3) WRITE(20,310) X,Y,DMZ
            IF(IFLG.EQ.4) WRITE(20,310) X,Y,DMF
            Y=Y+DY
 110     CONTINUE
         X=X+DX
 120  CONTINUE
      GOTO 300
C
 200  READ(10,*) IFLG
      READ(10,*) NUMB
      READ(10,*) U1,U2,U3,FL,FW,DPT,STR,DIP
      READ(10,*) POI,AMU,CMZ,GMINC,GMDEC,HH,BETA
      DO 210 K=1,NUMB
```

```
         READ(10,*) SITX(K),SITY(K),SITZ(K)
         IF(SITZ(K).GT.0.D0) THEN
            WRITE(6,*) 'POSITIV Z CANNOT BE ACCEPTED IN THIS PROGRAM'
            GOTO 400
         ENDIF
  210 CONTINUE
      CLOSE(10)
      IF(DIP.EQ.90.D0) THEN
         WRITE(6,*) 'DIP=90 CANNOT BE ACCEPTED IN THIS PROGRAM'
         GOTO 400
      ENDIF
      STR=90.D0-STR
      CALL CALCON1(POI,AMU,BETA,CMZ,DIP,STR,GMINC,GMDEC)
C
      DO 220 K=1,NUMB
         X=SITX(K)
         Y=SITY(K)
         Z=SITZ(K)
         CALL PIEZ(X,Y,Z,U1,U2,U3,DIP,STR,FL,FW,DPT,POI,
     &         AMU,BETA,HH,CMZ,GMINC,GMDEC,DMX,DMY,DMZ,DMF)
         IF(IFLG.EQ.1) WRITE(20,310) X,Y,DMX
         IF(IFLG.EQ.2) WRITE(20,310) X,Y,DMY
         IF(IFLG.EQ.3) WRITE(20,310) X,Y,DMZ
         IF(IFLG.EQ.4) WRITE(20,310) X,Y,DMF
  220 CONTINUE
  300 CLOSE(20)
  310 FORMAT(' ',F10.3,F10.3,F15.5)
  400 STOP
      END
C
C
C
      SUBROUTINE PIEZ(X,Y,Z,U1,U2,U3,DIP,STR,FL,FW,DPT,POI,
     &            AMU,BETA,HH,CMZ,GMINC,GMDEC,DMX,DMY,DMZ,DMF)
      INTEGER I
      IMPLICIT REAL*8 (A-H,O-Z)
C
C*************************************************************
C     CALCULATE EACH COMPONENT OF GEOMAGNETIC FIELD CHANGE
C*************************************************************
C****** INPUT ******
C     X,Y,Z        : COORDINATE OF OBS. POINT
C     U1,U2,U3     : DISLOCATION VECTOR
C     DIP,STR      : FAULT DIP AND STRIKE ANGLE
C     FL,FW,DPT    : FAULT LENGTH, WIDTH AND DEPTH OF BURIAL
C     POI,AMU      : POISSON RATIO AND LAME CONSTANT MU
C     BETA         : STRESS SENSITIVITY
C     HH           : CURIE DEPTH
C     CMZ          : CRUSTAL MAGNETIZATION
C     GMINC,GMDEC  : GEOMAGNETIC INCLINATION AND DECRINATION
C****** OUTPUT ******
C     DMX,DMY,DMZ,DMF : EAST, NORTH, DOWNWARD COMPONENTS
C                       AND TOTAL FORCE OF GEOMAGNETIC
C                       FIELD CHANGE
C
      COMMON /C0/ SD,CD,TD,SED
      DIMENSION DM0(3)
      DATA PI/3.141592653589793D0/
      P18=PI/180.D0
C-------------------------------------------
C          COORDINATE TRANSFORM
C-------------------------------------------
      XT=X*DCOS(STR*P18)+Y*DSIN(STR*P18)
      YT=X*DSIN(STR*P18)-Y*DCOS(STR*P18)
      ZT=Z
C-------
      DD1=DPT-ZT
      DD2=DPT-2.D0*HH+ZT
      DD3=DPT+2.D0*HH-ZT
C
      P1=YT*CD-DD1*SD
      P2=YT*CD-DD2*SD
      P3=YT*CD-DD3*SD
      Q1=YT*SD+DD1*CD
      Q2=YT*SD+DD2*CD
      Q3=YT*SD+DD3*CD
      QD=YT*SD+(DPT-HH)*CD
C
      FW2=(HH-DPT)/SD
      BDPT=DPT+FW*SD
      ZH=ZT-HH
C
      DO 100 I=1,3
         JDRV=I
         CALL DIFCOEF(XT,ZT,P1,P2,P3,Q1,Q2,Q3,QD,U1,U2,U3,
     &        FL,FW,DPT,FW2,BDPT,HH,JDRV,DM)
         DM0(I)=DM
  100 CONTINUE
      DMX0=DM0(1)
      DMY0=DM0(2)
      DMZ0=DM0(3)
```

```
C------------------------------------------
C       CALCULATE EACH COMPONENT OF
C         GEOMAGNETIC FIELD CHANGE
C------------------------------------------
      DMX=-DMX0*DCOS(STR*P18)-DMY0*DSIN(STR*P18)
      DMY=-DMX0*DSIN(STR*P18)+DMY0*DCOS(STR*P18)
      DMZ=-DMZ0
      DMF=DMX*DCOS(GMINC*P18)*DSIN(GMDEC*P18)
     &     +DMY*DCOS(GMINC*P18)*DCOS(GMDEC*P18)
     &     +DMZ*DSIN(GMINC*P18)
      RETURN
      END
C
C
C
      SUBROUTINE DIFCOEF(XT,ZT,P1,P2,P3,Q1,Q2,Q3,QD,U1,U2,U3,
     &         FL,FW,DPT,FW2,BDPT,HH,JDRV,DM)
      INTEGER I,J,K,JDRV
      IMPLICIT REAL*8 (A-H,O-Z)
C
C************************************************************
C           CALCULATE DIFFERENTIAL COEFFICIENT
C               OF PIEZOMAGNETIC POTENTIAL
C************************************************************
C****** INPUT ******
C     XT,ZT      : TRANSFORMED COORDINATE OF OBS. POINT
C     P1,P2,P3   : Y-COORDINATE OF OBS. POINT IN FAULT SYSTEM
C     Q1,Q2,Q3   : Z-COORDINATE OF OBS. POINT IN FAULT SYSTEM
C     U1,U2,U3   : DISLOCATION VECTOR
C     FL,FW,DPT  : FAULT LENGTH, WIDTH AND FAULT DEPTH
C     BDPT       : DEPTH OF BOTTOM OF FAULT
C     HH         : CURIE DEPTH
C     JDRV =1..OUTPUT XT-DERIVATION
C          =2..       YT-DERIVATION
C          =3..       ZT-DERIVATION OF PIEZOMAGNETIC POTENTIAL
C****** OUTPUT ******
C     DM : DIFFERENTIAL COEFFICIENT OF PIEZOMAGNETIC POTENTIAL
C
      COMMON /C0/ SD,CD,TD,SED
      DIMENSION DIF0(4),DIFH0(4),DIFH(4)
      DATA F0/0.D0/
C
      K=1
      ZH=ZT-HH
      DO 200 I=1,2
         IF(I.EQ.1) XI=XT
         IF(I.EQ.2) XI=XT-FL
         DO 100 J=1,2
            IF(J.EQ.1) THEN
               ET1=P1
               ET3=P3
            ELSEIF(J.EQ.2) THEN
               ET1=P1-FW
               ET3=P3-FW
            ENDIF
C
            SGN=1.D0
            CALL DERIV(XI,ET3,Q3,QD,ZH,HH,SGN,JDRV)
            IF(U1.NE.F0) CALL STRIKE_H0(HH,SH0)
            IF(U2.NE.F0) CALL DIP_H0(HH,DH0)
            IF(U3.NE.F0) CALL TENSILE_H0(HH,TH0)
            DIFH0(K)=U1*SH0+U2*DH0+U3*TH0
            CALL DERIV(XI,ET1,Q1,QD,ZH,HH,SGN,JDRV)
            IF(U1.NE.F0) CALL STRIKE_0(HH,S0)
            IF(U2.NE.F0) CALL DIP_0(HH,D0)
            IF(U3.NE.F0) CALL TENSILE_0(HH,T0)
            DIF0(K)=U1*S0+U2*D0+U3*T0
C
            IF(DPT.GT.HH) THEN
               IF(U1.NE.F0) CALL STRIKE_H(SGN,HH,SH)
               IF(U2.NE.F0) CALL DIP_H(SGN,HH,DH)
               IF(U3.NE.F0) CALL TENSILE_H(SGN,HH,TH)
               DIFH(K)=U1*SH+U2*DH+U3*TH
               GOTO 30
            ENDIF
            IF(DPT.LT.HH.AND.BDPT.LT.HH) GOTO 10
            IF(DPT.LT.HH.AND.BDPT.GT.HH) GOTO 20
C
 10         IF(J.EQ.1) ET2=P2
            IF(J.EQ.2) ET2=P2-FW
            SGN=-1.D0
            CALL DERIV(XI,ET2,Q2,QD,ZH,HH,SGN,JDRV)
            IF(U1.NE.F0) CALL STRIKE_H(SGN,HH,SH)
            IF(U2.NE.F0) CALL DIP_H(SGN,HH,DH)
            IF(U3.NE.F0) CALL TENSILE_H(SGN,HH,TH)
            DIFH(K)=U1*SH+U2*DH+U3*TH
            GOTO 30
C
 20         IF(J.EQ.1) THEN
               ET1=P1-FW2
               ET2=P2
```

```
            ELSEIF(J.EQ.2) THEN
               ET1=P1-FW
               ET2=P2-FW2
            ENDIF
            SGN=1.D0
            CALL DERIV(XI,ET1,Q1,QD,ZH,HH,SGN,JDRV)
            IF(U1.NE.F0) CALL STRIKE_H(SGN,HH,SH1)
            IF(U2.NE.F0) CALL DIP_H(SGN,HH,DH1)
            IF(U3.NE.F0) CALL TENSILE_H(SGN,HH,TH1)
            SGN=-1.D0
            CALL DERIV(XI,ET2,Q2,QD,ZH,HH,SGN,JDRV)
            IF(U1.NE.F0) CALL STRIKE_H(SGN,HH,SH2)
            IF(U2.NE.F0) CALL DIP_H(SGN,HH,DH2)
            IF(U3.NE.F0) CALL TENSILE_H(SGN,HH,TH2)
            DIFH(K)=U1*(SH1+SH2)+U2*(DH1+DH2)+U3*(TH1+TH2)
 30         K=K+1
 100      CONTINUE
 200  CONTINUE
      DM0=DIF0(1)-DIF0(2)-DIF0(3)+DIF0(4)
      DMH0=DIFH0(1)-DIFH0(2)-DIFH0(3)+DIFH0(4)
      DMH=DIFH(1)-DIFH(2)-DIFH(3)+DIFH(4)
      DM=DM0+DMH0+DMH
      RETURN DM
      END
C
C
C
      SUBROUTINE CALCON1(POI,AMU,BETA,CMZ,DIP,STR,GMINC,GMDEC)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C*************************************************************
C        CALCLATE MEDIUM CONSTANTS, FALUT-DIP CONSTANTS
C                 AND SEISMOMAGNETIC MOMENT
C*************************************************************
C****** INPUT ******
C     POI,AMU      : POISSON RATIO AND LAME CONSTANT MU
C     BETA         : STRESS SENSITIVITY
C     CMZ          : CRUSTAL MAGNETIZATION
C     DIP,STR      : FAULT DIP AND STRIKE ANGLE
C     GMINC,GMDEC  : GEOMAGNETIC INCLINATION AND DECRINATION
C
      COMMON /C0/ SD,CD,TD,SED
     &       /C1/ AL,AL1,AL2,AL3,AL4,AL5,AL6,CX,CY,CZ
      DATA PI/3.141592653589793D0/
C
      P18=PI/180.D0
      SD=DSIN(DIP*P18)
      CD=DCOS(DIP*P18)
      TD=DTAN(DIP*P18)
      SED=1.D0/CD
C------  AL : MEDIUM CONSTANT ALPHA        ------
C------        =(LAMBDA+MU)/(LAMBDA+2MU) ------
      AL=1.D0/(2.D0*(1.D0-POI))
      ALL=4.D0*AL-1.D0
      AL1=3.D0*AL/ALL
      AL2=12.D0*AL/ALL
      AL3=AL*(2.D0*AL+1.D0)/ALL
      AL4=AL*(2.D0*AL-5.D0)/ALL
      AL5=2.D0*AL*(1.D0-AL)/ALL
      AL6=6.D0*(AL*AL)/ALL
C------ SMM : SEISMOMAGNETIC MOMENT ------
      SMM=CMZ*BETA*AMU*(1.D0+POI)
      C0=0.5D0*SMM*1.D-7
      CX=C0*DCOS(GMINC*P18)*DSIN((STR+GMDEC)*P18)
      CY=-C0*DCOS(GMINC*P18)*DCOS((STR+GMDEC)*P18)
      CZ=C0*DSIN(GMINC*P18)
      RETURN
      END
C
C
C
      SUBROUTINE CALCON2(XI,ET,QQ,CC)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C*************************************************************
C           CALCULATE STATION GEOMETRY CONSTANTS
C*************************************************************
C****** INPUT ******
C     XI,ET,QQ : COORDINATES OF OBS. POINT IN FAULT SYSTEM
C     CC       : SGN*(QQ*COS(DIP)-ET*SIN(DIP))
C
      COMMON /PRM/ R,R3,RX,RE,RC,RRX,RRE,RRC,RX2,RE2,RC2,
     &      TRX,TRE,TRC,R3X2,R3E2,R3C2
      R=DSQRT(XI*XI+ET*ET+QQ*QQ)
      R3=R*R*R
C
      RX=R+XI
      RE=R+ET
      RC=R+CC
C
      RRX=R*RX
```

```
      RRE=R*RE
      RRC=R*RC
C
      RX2=RX*RX
      RE2=RE*RE
      RC2=RC*RC
C
      TRX=2.D0*R+XI
      TRE=2.D0*R+ET
      TRC=2.D0*R+CC
      R3X2=R3*RX2
      R3E2=R3*RE2
      R3C2=R3*RC2
      RETURN
      END
C
C
C
      SUBROUTINE DERIV(XI,ET,QQ,QD,ZH,HH,SGN,JDRV)
      INTEGER LT,JDRV
      IMPLICIT REAL*8 (A-H,O-Z)
C
C**************************************************************
C         CALCULATE DIFFERENTIAL COEFFICIENT OF
C                ELEMENTARY FUNCTIONS
C**************************************************************
C****** INPUT ******
C     XI,ET,QQ : COORDINATES OF OBS. POINT IN FAULT SYSTEM
C     GD       : YT*SIN(DIP)+(DPT-HH)*COS(DIP)
C     ZH       : ZT-HH
C     HH       : CURIE DEPTH
C     JDRV =1..CALCULATE XT-DERIVATE
C          =2..           YT-DERIVATE
C          =3..           ZT-DERIVATE OF ELEMENTARY FUNCTIONS
C
      COMMON /C0/ SD,CD,TD,SED
     &     /PRM/ R,R3,RX,RE,RC,RRX,RRE,RRC,RX2,RE2,RC2,
     &     TRX,TRE,TRC,R3X2,R3E2,R3C2
     &     /MAIN/ W(9)
     &     /ELM1/ FLX,FLE,FLC,TAN,G(2)
     &     /ELM2/ A(3),B(3),D(3),E(3),F(3)
     &     /ELM3/ SX1,SY1,SZ1,DX1,DY1,DZ1,TX1,TY1,TZ1
      DATA F0,F1,EPS/0.D0,1.D0,1.D-6/
C
      YY=ET*CD+QQ*SD
      CC=SGN*(QQ*CD-ET*SD)
      CALL CALCON2(XI,ET,QQ,CC)
      IF(ABS(RE).GT.EPS) LT=1
      IF(ABS(RE).LE.EPS) LT=0
C
      AI0=F1/RRC
      IF(LT.EQ.1) AI0=AI0+SGN*SD/RRE
C=====================================
C        X-DERIVATIVE
C=====================================
      IF(JDRV.NE.1) GOTO 100
C--------------------------
C   ELEMENTARY FUNCTIONS
C        DIPOLES
C--------------------------
      TAN=F0
      FLE=F0
      FLX=F1/R
      FLC=XI/RRC
      G(1)=SGN*YY/RRC
      IF(LT.EQ.1) THEN
         TAN=TAN-QQ/RRE
         FLE=FLE+XI/RRE
         G(1)=G(1)+QQ/RRE
      ENDIF
      G(2)=FLC+SGN*SD*FLE
C
      W(1)=TD*(F1/RC-(XI*XI)/(R*RC2))+TD*TD*G(1)
      W(2)=-TD*XI*YY/(R*RC2)-SGN*TD*TD*G(2)
      W(3)=TD*G(2)
      W(4)=CD*(W(2)-SD*FLE)
      W(5)=CD*W(1)
      W(6)=-SD*G(1)
      W(7)=-SD*(W(2)-SD*FLE)
      W(8)=-SD*W(1)
      W(9)=SD*TD*G(1)
C--------------------------
C   ELEMENTARY FUNCTIONS
C   QUADRA AND OCTA POLES
C--------------------------
      IF(LT.EQ.1) THEN
         A(1)=F1/RRE-(XI*XI)*TRE/R3E2
         A(2)=-XI*CD/(R*RE2)-XI*YY*TRE/R3E2
         A(3)=SGN*XI*SD/(R*RE2)-XI*CC*TRE/R3E2
      ELSE
         DO 10 I=1,3
```

```
            A(I)=F0
 10      CONTINUE
       ENDIF
C
       B(1)=F1/RRC-(XI*XI)*TRC/R3C2
       B(2)=-XI*YY*TRC/R3C2
C
       D(1)=-XI/R3
       D(2)=-YY/R3
       D(3)=-CC/R3
C
       E(1)=F0
       E(2)=SGN*CC/R3
       E(3)=-SGN*YY/R3
       AI1=-XI*TRC/R3C2
       F(2)=F0
       IF(LT.EQ.1) THEN
          E(1)=E(1)+XI*QQ*TRE/R3E2
          E(2)=E(2)+SD/RRE-(XI*XI)*SD*TRE/R3E2
          E(3)=E(3)+SGN*(CD/RRE-(XI*XI)*CD*TRE/R3E2)
          AI1=AI1-SGN*XI*SD*TRE/R3E2
          F(2)=F(2)-SGN*XI*SD/(R*RE2)
       ENDIF
       F(1)=XI*AI1+AI0
       F(2)=F(2)+YY*SED*AI1
       F(3)=B(1)+D(3)
C
       SX1=QD*A(1)+ZH*F(1)*TD
       SY1=QD*A(2)+ZH*F(2)*SD
       SZ1=QD*A(3)+ZH*A(2)*SD
       DX1=QD*D(1)+ZH*B(2)*SD
       DY1=QD*D(2)-ZH*F(3)*SD
       DZ1=QD*D(3)+ZH*D(2)*SD
       TX1=QD*E(1)-ZH*(A(2)+SD*B(2))*TD
       TY1=QD*(D(3)-SD*A(1))+ZH*(D(2)-TD*F(1))*SD
       TZ1=QD*E(3)+ZH*E(2)*SD
C==============================================
C         Y-DERIVATIVE
C==============================================
 100   IF(JDRV.NE.2) GOTO 200
C--------------------------
C    ELEMENTARY FUNCTIONS
C         DIPOLES
C--------------------------
       TAN=-SGN*CC/RRX
       FLE=F0
       FLX=YY/RRX
       FLC=YY/RRC
       G(1)=-SGN*XI/RRC
       IF(LT.EQ.1) THEN
          TAN=TAN+XI*SD/RRE
          FLE=FLE+CD/RE+YY/RRE
          G(1)=G(1)-XI*SD/RRE
       ENDIF
       G(2)=FLC+SGN*FLE
C
       W(1)=-TD*XI*YY/(R*RC2)+TD*TD*G(1)
       W(2)=TD*(F1/RC-(YY*YY)/(R*RC2))-SGN*TD*TD*G(2)
       W(3)=TD*G(2)
       W(4)=CD*(W(2)-SD*FLE)
       W(5)=CD*W(1)
       W(6)=-SD*G(1)
       W(7)=-SD*(W(2)-SD*FLE)
       W(8)=-SD*W(1)
       W(9)=SD*TD*G(1)
C--------------------------
C    ELEMENTARY FUNCTIONS
C    QUADRA AND OCTA POLES
C--------------------------
       IF(LT.EQ.1) THEN
          A(1)=-XI*CD/(R*RE2)-XI*YY*TRE/R3E2
          A(2)=(SD*SD)/RE2-(YY*CD+SGN*CC*SD)/(R*RE2)
     &         -YY*YY*TRE/R3E2
          A(3)=SGN*SD*CD/RE2-(CC*CD-SGN*YY*SD)/(R*RE2)
     &         -YY*CC*TRE/R3E2
       ELSE
          DO 110 I=1,3
            A(I)=F0
 110      CONTINUE
       ENDIF
C
       B(1)=-XI*YY*TRC/R3C2
       B(2)=F1/RRC-(YY*YY)*TRC/R3C2
C
       D(1)=-YY/R3
       D(2)=F1/RRX-(YY*YY)*TRX/R3X2
       D(3)=-YY*CC*TRX/R3X2
C
       E(1)=F0
       E(2)=SGN*YY*CC*TRX/R3X2
       E(3)=SGN*(F1/RRX-(YY*YY)*TRX/R3X2)
```

```
      AI1=-YY*TRC/R3C2
      F(2)=SED*AI0
      IF(LT.EQ.1) THEN
         E(1)=E(1)-SD/RE2+SGN*CC/(R*RE2)+YY*QQ*TRE/R3E2
         E(2)=E(2)    -SD*(XI*CD/(R*RE2)+XI*YY*TRE/R3E2)
         E(3)=E(3)-SGN*CD*(XI*CD/(R*RE2)+XI*YY*TRE/R3E2)
         AI1=AI1-SGN*SD*(CD/(R*RE2)+YY*TRE/R3E2)
         F(2)=F(2)-SGN*SD*(CD/RE2+YY/(R*RE2))
      ENDIF
      F(1)=XI*AI1
      F(2)=F(2)+YY*SED*AI1
      F(3)=B(1)+D(3)
C
      SX1=QD*A(1)+ZH*F(1)*TD
      SY1=QD*A(2)+ZH*F(2)*SD
      SZ1=QD*A(3)+ZH*A(2)*SD
      DX1=QD*D(1)+ZH*B(2)*SD+SD/R
      DY1=QD*D(2)-ZH*F(3)*SD+SD*YY/RRX
      DZ1=QD*D(3)+ZH*D(2)*SD+SD*CC/RRX
      TX1=QD*E(1)-ZH*(A(2)+SD*B(2))*TD
      TY1=QD*(D(3)-SD*A(1))+ZH*(D(2)-TD*F(1))*SD
     &      +SD*CC/RRX
      TZ1=QD*E(3)+ZH*E(2)*SD+SD*SGN*YY/RRX
      IF(LT.EQ.1) THEN
         SX1=SX1+SD*XI/RRE
         SY1=SY1+SD*(CD/RE+YY/RRE)
         SZ1=SZ1-SD*(SGN*SD/RE-CC/RRE)
         TX1=TX1-SD*QQ/RRE
         TY1=TY1-SD*SD*XI/RRE
         TZ1=TZ1+SD*SGN*CD*XI/RRE
      ENDIF
C==========================================
C        Z-DERIVATIVE
C==========================================
 200  IF(JDRV.NE.3) GOTO 300
C--------------------------
C   ELEMENTARY FUNCTIONS
C         DIPOLES
C--------------------------
      TAN=-SGN*YY/RRX
      FLE=F0
      FLX=-CC/RRX
      FLC=-F1/R
      G(1)=F0
      G(2)=F0
      IF(LT.EQ.1) THEN
         TAN=TAN-SGN*XI*CD/RRE
         FLE=FLE+SGN*SD/RE-CC/RRE
         G(1)=G(1)+SGN*XI*CD/RRE
      ENDIF
      G(2)=FLC+SGN*FLE
C
      W(1)=TD*XI/RRC+TD*TD*G(1)
      W(2)=TD*YY/RRC-SGN*TD*TD*G(2)
      W(3)=TD*G(2)
      W(4)=CD*(W(2)-SD*FLE)
      W(5)=CD*W(1)
      W(6)=-SD*G(1)
      W(7)=-SD*(W(2)-SD*FLE)
      W(8)=-SD*W(1)
      W(9)=SD*TD*G(1)
C--------------------------
C   ELEMENTARY FUNCTIONS
C   QUADRA AND OCTA POLES
C--------------------------
      IF(LT.EQ.1) THEN
         A(1)=-XI*CD/(R*RE2)-XI*YY*TRE/R3E2
         A(2)=SD*SD/RE2-(YY*CD+SGN*CC*SD)/(R*RE2)
     &         -YY*YY*TRE/R3E2
         A(3)=SGN*SD*CD/RE2-(CC*CD-SGN*YY*SD)/(R*RE2)
     &         -YY*CC*TRE/R3E2
      ELSE
         DO 210 I=1,3
            A(I)=F0
 210     CONTINUE
      ENDIF
C
      B(1)=XI/R3
      B(2)=YY/R3
C
      D(1)=CC/R3
      D(2)=YY*CC*TRX/R3X2
      D(3)=-F1/RRX+(CC*CC)*TRX/R3X2
C
      E(1)=F0
      E(2)=SGN*(F1/RRX-(CC*CC)*TRX/R3X2)
      E(3)=SGN*YY*CC*TRX/R3X2
      AI1=F1/R3
      F(2)=F0
      IF(LT.EQ.1) THEN
         E(1)=E(1)+SGN*(CD/RE2+YY/(R*RE2))-QQ*CC*TRE/R3E2
```

```
         E(2)=E(2)+SD*(-SGN*XI*SD/(R*RE2)+XI*CC*TRE/R3E2)
         E(3)=E(3)+CD*(-XI*SD/(R*RE2)+SGN*XI*CC*TRE/R3E2)
         AI1=AI1-SD*(SD/(R*RE2)-SGN*CC*TRE/R3E2)
         F(2)=F(2)+SD*(-SD/RE2+SGN*CC/(R*RE2))
      ENDIF
      F(1)=XI*AI1
      F(2)=F(2)+SED*YY*AI1
      F(3)=B(1)+D(3)
C
      SX1=QD*A(1)+ZH*F(1)*TD+XI*AI0*TD
      SY1=QD*A(2)+ZH*F(2)*SD+YY*AI0*TD
      SZ1=QD*A(3)+ZH*A(2)*SD
      DX1=QD*D(1)+ZH*B(2)*SD+SD*YY/RRC
      DY1=QD*D(2)-ZH*F(3)*SD-SD*(XI/RRC+CC/RRX)
      DZ1=QD*D(3)+ZH*D(2)*SD+SD*YY/RRX
      TX1=QD*E(1)-ZH*(A(2)+SD*B(2))*TD
     &     -SD*TD*YY/RRC
      TY1=QD*(D(3)-SD*A(1))+ZH*(D(2)-TD*F(1))*SD
     &     +SD*(YY/RRX-TD*XI*AI0)
      TZ1=QD*E(3)+ZH*E(2)*SD-SD*SGN*CC/RRX
      IF(LT.EQ.1) THEN
         SY1=SY1+SGN*SD*SD/RE
         SZ1=SZ1+SD*(CD/RE+YY/RRE)
         TX1=TX1-TD*(CD/RE+YY/RRE)
         TZ1=TZ1+SD*SD*XI/RRE
      ENDIF
C
  300 RETURN
      END
C
C
C
      SUBROUTINE STRIKE_0(HH,S0)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /MAIN/ W(9)
     &     /C1/ AL,AL1,AL2,AL3,AL4,AL5,AL6,CX,CY,CZ
C
      SX=2.D0*W(1)
      SY=2.D0*W(2)
      SZ=2.D0*W(3)
      S0=CX*SX+CY*SY+CZ*SZ
      RETURN S0
      END
C
C
C
      SUBROUTINE STRIKE_H0(HH,SH0)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /C0/ SD,CD,TD,SED
     &     /C1/ AL,AL1,AL2,AL3,AL4,AL5,AL6,CX,CY,CZ
     &     /MAIN/ W(9)
     &     /ELM1/ FLX,FLE,FLC,TAN,G(2)
     &     /ELM2/ A(3),B(3),D(3),E(3),F(3)
     &     /ELM3/ SX1,SY1,SZ1,DX1,DY1,DZ1,TX1,TY1,TZ1
C
      SH0X=(-2.D0+AL3)*W(3)-AL1*TAN-AL5*G(1)
     &     -2.D0*AL3*HH*(A(1)*CD-F(1)*TD)
     &     -AL2*HH*F(1)*TD-AL5*SX1
C
      SH0Y=(-2.D0+AL3)*W(2)+AL3*FLE*SD+AL5*G(2)
     &     -2.D0*AL3*HH*(A(2)*CD-F(2)*SD)
     &     -AL2*HH*F(2)*SD-AL5*SY1
C
      SH0Z=-(2.D0+AL4)*W(3)-AL*FLE*CD
     &     -2.D0*AL*HH*(A(3)*CD-A(2)*SD)
     &     -AL2*HH*A(2)*SD+AL5*SZ1
C
      SH0=CX*SH0X+CY*SH0Y+CZ*SH0Z
      RETURN SH0
      END
C
C
C
      SUBROUTINE STRIKE_H(SGN,HH,SH)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /C0/ SD,CD,TD,SED
     &     /C1/ AL,AL1,AL2,AL3,AL4,AL5,AL6,CX,CY,CZ
     &     /MAIN/ W(9)
     &     /ELM1/ FLX,FLE,FLC,TAN,G(2)
     &     /ELM2/ A(3),B(3),D(3),E(3),F(3)
     &     /ELM3/ SX1,SY1,SZ1,DX1,DY1,DZ1,TX1,TY1,TZ1
C
      SHX1=-SGN*(AL3*W(2)+AL1*TAN+AL5*G(1))
      SHY1=-SGN*AL3*W(2)-AL5*G(2)
      SHZ1=AL4*W(3)+AL*FLE*CD
      IF(SGN.EQ.1.D0) THEN
         SHX2=AL5*SX1
         SHY2=AL5*SY1-AL3*SD*FLE
         SHZ2=-AL5*SZ1
      ELSEIF(SGN.EQ.-1.D0) THEN
         SHX2=-AL6*SX1
```

```
           SHY2=-AL6*SY1+(AL1-AL6)*FLE
           SHZ2=-AL6*SZ1
        ENDIF
        SHX=SHX1+SHX2
        SHY=SHY1+SHY2
        SHZ=SHZ1+SHZ2
C
        SH=CX*SHX+CY*SHY+CZ*SHZ
        RETURN SH
        END
C
C
C
        SUBROUTINE DIP_0(HH,D0)
        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON /MAIN/ W(9)
     &      /C1/ AL,AL1,AL2,AL3,AL4,AL5,AL6,CX,CY,CZ
C
        DX=-2.D0*W(4)
        DY=2.D0*W(5)
        DZ=2.D0*W(6)
        D0=CX*DX+CY*DY+CZ*DZ
        RETURN D0
        END
C
C
C
        SUBROUTINE DIP_H0(HH,DH0)
        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON /C0/ SD,CD,TD,SED
     &      /C1/ AL,AL1,AL2,AL3,AL4,AL5,AL6,CX,CY,CZ
     &      /MAIN/ W(9)
     &      /ELM1/ FLX,FLE,FLC,TAN,G(2)
     &      /ELM2/ A(3),B(3),D(3),E(3),F(3)
     &      /ELM3/ SX1,SY1,SZ1,DX1,DY1,DZ1,TX1,TY1,TZ1
C
        DH0X=(2.D0-AL3)*W(4)
     &      +AL5*(2.D0*SD*CD*FLE-(CD-SD*TD)*G(2))
     &      +2.D0*AL4*HH*(D(1)*CD-B(2)*SD)-AL2*HH*A(2)
     &      +AL5*DX1
C
        DH0Y=(-2.D0+AL3)*W(5)-AL*SD*FLX+AL3*CD*TAN
     &      -AL5*(CD+SD*TD)*G(1)
     &      +2.D0*AL3*HH*(D(2)*CD+F(3)*SD)+AL2*HH*F(1)*SD
     &      +AL5*DY1
C
        DH0Z=-(2.D0+AL4)*W(6)+AL*CD*FLX+AL3*TAN
     &      +2.D0*AL*HH*(D(3)*CD-D(2)*SD)-AL2*HH*A(1)*SD*CD
     &      -AL5*DZ1
C
        DH0=CX*DH0X+CY*DH0Y+CZ*DH0Z
        RETURN DH0
        END
C
C
C
        SUBROUTINE DIP_H(SGN,HH,DH)
        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON /C0/ SD,CD,TD,SED
     &      /C1/ AL,AL1,AL2,AL3,AL4,AL5,AL6,CX,CY,CZ
     &      /MAIN/ W(9)
     &      /ELM1/ FLX,FLE,FLC,TAN,G(2)
     &      /ELM2/ A(3),B(3),D(3),E(3),F(3)
     &      /ELM3/ SX1,SY1,SZ1,DX1,DY1,DZ1,TX1,TY1,TZ1
C
        DHX1=SGN*AL3*W(4)
     &      -AL5*(2.D0*SD*CD*FLE-(CD-SD*TD)*G(2))
        DHY1=-SGN*AL3*W(5)+AL*FLX-SGN*AL3*CD*TAN
     &      +SGN*AL5*(CD-SD*TD)*G(1)
        DHZ1=SGN*AL4*W(6)-AL*CD*FLX-SGN*AL3*SD*TAN
        IF(SGN.EQ.1.D0) THEN
           DHX2=-AL5*DX1
           DHY2=-AL5*DY1
           DHZ2=AL5*DZ1
        ELSEIF(SGN.EQ.-1.D0) THEN
           DHX2=AL6*DX1
           DHY2=AL6*DY1
           DHZ2=AL6*DZ1
        ENDIF
        DHX=DHX1+DHX2
        DHY=DHY1+DHY2
        DHZ=DHZ1+DHZ2
C
        DH=CX*DHX+CY*DHY+CZ*DHZ
        RETURN DH
        END
C
C
C
        SUBROUTINE TENSILE_0(HH,T0)
        IMPLICIT REAL*8 (A-H,O-Z)
```

```
      COMMON /MAIN/ W(9)
     &      /C1/ AL,AL1,AL2,AL3,AL4,AL5,AL6,CX,CY,CZ
C
      TX=-2.D0*W(7)
      TY=2.D0*W(8)
      TZ=2.D0*W(9)
      T0=CX*TX+CY*TY+CZ*TZ
      RETURN T0
      END
C
C
C
      SUBROUTINE TENSILE_H0(HH,TH0)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /C0/ SD,CD,TD,SED
     &      /C1/ AL,AL1,AL2,AL3,AL4,AL5,AL6,CX,CY,CZ
     &      /MAIN/ W(9)
     &      /ELM1/ FLX,FLE,FLC,TAN,G(2)
     &      /ELM2/ A(3),B(3),D(3),E(3),F(3)
     &      /ELM3/ SX1,SY1,SZ1,DX1,DY1,DZ1,TX1,TY1,TZ1
C
      TH0X=(2.D0-AL3)*W(7)
     &      +2.D0*AL5*SD*FLC-AL*FLE
     &      +2.D0*AL4*HH*(E(1)*CD+(A(2)+SD*B(2))*TD)
     &      -(AL2-2.D0*AL6)*HH*A(3)+AL5*TX1
C
      TH0Y=(-2.D0+AL3)*W(8)
     &      -AL3*SD*TAN-AL*CD*FLX+2.D0*AL5*SD*G(1)
     &      -2.D0*AL*HH*((D(3)-A(1)*SD)*CD
     &                   -(D(2)-F(1)*TD)*SD)
     &      +(AL2-2.D0*AL6)*HH*F(1)*SD*TD-AL5*TY1
C
      TH0Z=-(2.D0+AL4)*W(9)
     &      +AL3*CD*TAN-AL*SD*FLX
     &      -AL3*HH*(E(3)*CD-E(2)*SD)+AL2*HH*A(1)*SD*SD
     &      -AL5*TZ1
C
      TH0=CX*TH0X+CY*TH0Y+CZ*TH0Z
      RETURN TH0
      END
C
C
C
      SUBROUTINE TENSILE_H(SGN,HH,TH)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /C0/ SD,CD,TD,SED
     &      /C1/ AL,AL1,AL2,AL3,AL4,AL5,AL6,CX,CY,CZ
     &      /MAIN/ W(9)
     &      /ELM1/ FLX,FLE,FLC,TAN,G(2)
     &      /ELM2/ A(3),B(3),D(3),E(3),F(3)
     &      /ELM3/ SX1,SY1,SZ1,DX1,DY1,DZ1,TX1,TY1,TZ1
C
      THX1=SGN*AL3*W(7)+SGN*AL*FLE-SGN*2.D0*AL5*SD*FLC
      THY1=SGN*AL3*W(8)+AL*CD*FLX-SGN*AL4*TAN
     &      -SGN*2.D0*AL5*G(1)
      THZ1=SGN*AL4*W(9)+AL*SD*FLX-SGN*AL3*CD*TAN
C
      IF(SGN.EQ.1) THEN
         THX2=-AL5*TX1
         THY2=-AL5*TY2
         THZ2=AL5*TZ1
      ELSE
         THX2=AL6*TX1
         THY2=AL6*TY2
         THZ2=AL6*TZ1
      ENDIF
      THZ=THZ1+THZ2
      TH=CZ*THZ
      RETURN TH
      END
```