

https://dspace.org/

Further developments of the boundary element method with applications in mining

Diering, Julian Anthony Cameron

2015-03-05
http://hdl.handle.net/10539/17170

Downloaded from a DSpace repository. https://dspace.org/

```
C
      IF(ISSW(14).EQ.1)TENS=9999.
      NIL=0
      IDISK=NIL
      DO 100 NUM=1,NITER
      NIL=0
      AMAX=ZERO
      IBUF=NIL
      ISTAR=ICHAN
      JOUT=NIL
      CALL DUMPO(IBUF,IDISK,BUF,ISTAR,2)
      CALL LUMPR
C
C     SET VECTOR OF UNKNOWNS.
C
      DO 506 N=1,NELT
      DO 506 I=1,3
      IBB=IBC(N,I)
      IF(IBB.NE.1)GO TO 507
      VECT(N,I)=U(N,I)
      GO TO 506
  507 VECT(N,I)=T(N,I)
  506 CONTINUE
C
C     B.E. SECTION
C
      DO 511 NL=1,NLUMP
      READ BINARY (6) LIND,KIND
C 102 FORMAT(2011)
      MEL=LMOD(NL)
      CALL NULL(SUML,FALL)
C
C     LUMP-LUMP
C
      DO 19 NL=1,NLUMP
      IF(LIND(NL).NE.1)GO TO 19
      CALL TUL(NL,TRAC,DIS)
      CALL SUMOP(SUML,DIS,TRAC)
   19 CONTINUE
      DO 70 K=1,NLSX
      DO 70 L=1,NLSY
      IF(KIND(K,L).NE.1)GO TO 70
      DIS(1)=XLANG(K,L)
      DIS(2)=YLANG(K,L)
      DIS(3)=ZLANG(K,L)
      CALL SUMDI(SUML,DIS)
   70 CONTINUE
C
C     CALCULATE DETAILED SUMS.
C
      MOD1=LMOD(NL)
      DO 1 ME=1,MEL
      M=LUMPO(NL,ME)
      READ BINARY (1) UDIAG
      READ BINARY (7) TEMP
      KODR=IABS(KOD(M))
      DO 33 I=1,3
   33 SUM(I)=SUML(I)/RNUM+TEMP(I)
C
C     LUMP-ELT
C
```

```
C
C     ELT-ELT
C
      CALL SUMELT(SUM,VECT,MOD1,KODR,N)
C
C     ADD CONTRIBUTION FROM DD ELEMENTS.
C
      IF(MOD1.NE.MODDD)GO TO 63
      DO 60 K=1,NLSX
      DO 60 L=1,NLSY
      IF(KIND(K,L).NE.2)GO TO 61
      DIS(1)=XLARG(K,L)/RNUM
      DIS(2)=YLARG(K,L)/RNUM
      DIS(3)=ZLARG(K,L)/RNUM
      CALL SUMDD(SUM,DIS)
      GO TO 60
   61 CONTINUE
      IF(KIND(K,L).EQ.1)GO TO 60
C     ##########################
      KB=(K-1)*ISMAL+IONE
      KE=K*ISMAL
      LB=(L-1)*JSMAL+IONE
      LE=L*JSMAL
      DO 62 I=KB,KE
      DO 62 J=LB,LE
      DIS(1)=DX(I,J)
      DIS(2)=DY(I,J)
      DIS(3)=DZ(I,J)
   62 CALL SUMDD(SUM,DIS)
   60 CONTINUE
   63 CONTINUE
      IFF=IFACE(M)
      DO 14 I=1,3
      IBB=IBC(M,I)
      IF(IBB.NE.1)GO TO 504
      DIFF=OMEGA*SUM(I)/HALF
      VECT(M,I)=VECT(M,I)+DIFF
      U(M,I)=U(M,I)+DIFF
      IF(KODR.NE.7)GO TO 505
      U(IFF,I)=U(M,I)
      GO TO 505
  504 IF(IBB.NE.2)GO TO 505
      DIFF=OMEGA*SUM(I)/UDIAG(I)
      T(M,I)=T(M,I)-DIFF
      VECT(M,I)=VECT(M,I)-DIFF
      IF(KODR.NE.7)GO TO 505
      T(IFF,I)=-T(M,I)
  505 CONTINUE
      ABSUM=ABS(SUM(I))
      IF(ABSUM.GT.AMAX)AMAX=ABSUM
      IF(DEL.GE.ABSUM)GO TO 10
      IF(IBB.GT.2)GO TO 10
      JOUT=JOUT+1
      IF(JOUT.LT.5)WRITE(10,20) U(M,I),T(M,I),ABSUM,M,I
   20 FORMAT(1X,3(F16.5,1X),2I4)
   10 CONTINUE
   14 CONTINUE
      IF(NUM.LT.NSLIP)GO TO 1
      READ BINARY (8) DCD,W
C
```

```
C     KODE  7 = INTACT
C     KODE  8 = TRACTION FREE SURFACE.
C     KODE  9 = MINED WITH COMPLETE CLOSURE RESTRICTION.
C     KODE 11 = SHEAR FAILED INTERFACE.
C     KODE 10 = TENSILE FAILED INTERFACE.
C
      IF(KODE LT. 7 OR. KODE. EQ. 8)GO TO 1
C
C     LOAD TDIAG AND UDIAG
C
      DO 57 I=1,3
      UDIAG(I)=-SUM(I)*OMEGA/UDIAG(I)
   57 TDIAG(I)=SUM(I)*OMEGA/HALF
C
C     CALCULATE TRAC,DIS AND PRS
C
      DO 41 I=1,3
      PRS(I)=ZERO
      TRAC(I)=ZERO
      DIS(I)=ZERO
      SUMT=ZERO
      SUMU=ZERO
      DO 53 J=1,3
      DCI=DCO(I,J)
      SUMU=SUMU+DCI*UDIAG(J)
      SUMT=SUMT+DCI*TDIAG(J)
      PRS(I)=PRS(I)+DCI*U(J)
      TRAC(I)=TRAC(I)+DCI*T(M,J)
   53 DIS(I)=DIS(I)+DCI*(U(M,J)-U(IFF,J))
C
C     CHANGE IBC 3 AND 4 UNKNOWNS.
C
      IBB=IBC(M,1)
      IF(IBB.EQ.3)DIS(I)=DIS(I)+SUMT
      IF(IBB.EQ.4)TRAC(I)=TRAC(I)+SUMU
      RES(I)=TRAC(I)-PRS(I)
   41 CONTINUE
C
C     CHECK FOR FAILURE.
C
      IF(KODE.EQ.9)GO TO 42
      IF(KODE.EQ.10)GO TO 47
      SHSTR=COH-PHI*RES(3)
      SHEAR=SQRT(RES(1)**2+RES(2)**2)
      IF(KODE.EQ.11)GO TO 43
      IF(SHEAR.GT.SHSTR)GO TO 43
      GO TO 45
   43 IF((RES(3)).GT.TENS)GO TO 44
C
C     SHEAR FAILURE
C
      IBC(M,1)=3
      IBC(M,2)=3
      IBC(M,3)=4
      DIS(3)=ZERO
   54 TRAC(1)=(RES(1)*SHSTR/SHEAR)+PRS(1)
      TRAC(2)=(RES(2)*SHSTR/SHEAR)+PRS(2)
      KODE=11
      TYPE "S",M
      GO TO 45
```

```
C
C     TENSILE FAILURE
C
   44 CONTINUE
      KODR=10
      DO 46 I=1,3
      TRAC(I)=PRS(I)
   46 IBC(M,I)=3
      TYPE "T",M
      GO TO 45
C
C     TOTAL CLOSURE ELEMENTS
C
   42 CONTINUE
      IF(DIS(3).LT.THIK)GO TO 45
      IBC(M,1)=3
      IBC(M,2)=3
      DIS(3)=THIK
      IBC(M,3)=4
      GO TO 45
   47 CONTINUE
C
C     CHECK FOR CLOSURE OF AN OPEN FAULT ELEMENT.
C
      IF(DIS(3).LT.ZERO)GO TO 45
      TYPE "CLOS-F",M
      DIS(3)=ZERO
      IBC(M,3)=4
      IBC(IFF,3)=4
      IF(RES(3).GT.ZERO)GO TO 45
      IBC(M,3)=3
      IBC(IFF,3)=3
      TRAC(3)=PRS(3)
   45 CONTINUE
C
C     BACK TRANSFORM TRAC AND DIS
C
      DO 50 J=1,3
      T(M,J)=ZERO
      DNEW=ZERO
      DO 51 I=1,3
      DNEW=DNEW+DCO(I,J)*DIS(I)
   51 T(M,J)=T(M,J)+DCO(1,J)*TRAC(I)
      T(IFF,J)=-T(M,J)
      U(M,J)=U(IFF,J)+DNEW
      IBB=IBC(M,J)
      IF(IBB.GT.2)IBC(IFF,J)=IBB
   50 CONTINUE
      N4=INDEX(M,4)
      IF(N4.NE.NIL)KODR=-KODR
      KOD(M)=KODR
      KOD(IFF)=KODR
    1 CONTINUE
      IBM=IBUF-2
      IF(NUM.EQ.1)WRITE(IRITE,200)ISTAR,IBUF,(BUF(J),J=IBM,IBUF),LIND
  200 FORMAT(216,3F10.6,1X,6011)
  511 CONTINUE
      CALL DDSOL(JOUT,VECT,AMAX)
      REWIND 8
      REWIND 6
```

```
      REWIND 7
      REWIND 1
      REWIND 2
      REWIND 13
      WRITE(IRITE,448)NUM,JOUT,AMAX
      IF(JOUT.EQ.NIL.OR.AMAX.LT.DLL)GO TO 101
  448 FORMAT(1X,I4" ITS",I6" ELTS OUT","  MAX ER =",F10.5)
      IF(NUM.EQ.1)WRITE(IRITE,449)ISTAR
  449 FORMAT(/" LAST BLOCK ACCESSED ="I4,/)
  100 CONTINUE
      NUM=NUM-1
  101 CONTINUE
      IF(JOUT.NE.0)WRITE(IRITE,121)NUM,JOUT,AMAX
  121 FORMAT(" NO CONVERGENCE AFTER"I4" ITERATIONS"I8" ELTS OUT"
     14X"MAX DIFF ="F16.9)
      DO 450 N=1,NELT
  450 WRITE BINARY (3) (T(N,L),L=1,3),(U(N,L),L=1,3)
      DO 451 I=1,NN
  451 WRITE BINARY (3) ((DX(I,J),DY(I,J),DZ(I,J)),J=1,MM)
      WRITE BINARY (3) OZZT
      CALL EXEC(5,1)
      CALL TTT(IRITE)
  447 CONTINUE
      CALL BACK
      STOP
      END
```

```
      END
```

****

```
C
C     SUBROUTINE TO CALCULATE S AND D COEFFICIENTS.
C
      SUBROUTINE SSDD(XR,XE,DCO,W1)
C     JACD
C
      COMMON KOD(220),IBC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),DLUMC(50,3),LIND(50),LUMPO(50,12),LKOD(50),LMOD(50),
     3  MINE(25,25),DZ(25,25),DX(25,25),DY(25,25),OZZT(25,25),
     4  AK(9,9),BK(9,9),DK(9,9),BNA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIM(3,3),PRIM1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),DNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),DC(3),ISUB(3),XNUM(3),RCOM(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIR(8,8),NMULT(8,3),
     8  TRAN42,TRAN21,TRAN7,TRAN5,TRANLL,TRANL,TRANDL,TRANDD,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,MM,NLSX,NLSY,NDIML,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NSLIP,NIL,MODDD,NS,
     *  ZERO,ONE,IONE,II,I2,I3,THIRD,HALF,TWO,NITER,OMEGA,DLL,PI,A,B,C,R,RSQ,EW,HLIL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,G,COND,CPAR,THIK,ESEAM,GSEAM,COH,PHI,
     2  IDFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELEL,NREGO,KODE,ISSW(15),ISW(15)
C
      DIMENSION XR(9),XE(3),DCO(3)
      R3=R*RSQ
      WCK=-W1*CK/RSQ
      WCR=W1*CONS/R
      WS=W1*CD/R3
      WD=-WCK
      DRDN=ZERO
      DO 5 L=1,3
      RCOM(L)=(XE(L)-XR(L))/R
    5 DRDN=DRDN+RCOM(L)*DCO(L)
      D3=DRDN*3.
      DO 6 I=1,3
```

```
      RCI=RCOM(I)
      RIV3=RCI*V3
      RIV2=RCI*V2
      DCI=DCO(I)
      RI3=RCI*3.
      J=3
      RCJ=RCOM(J)
      DCJ=DCO(J)
      DIJ=DELTA(I,J)
      RIRJ=RCI*RCJ
      RNV7=V7*(DCI*RCJ+DCJ*RCI)
      DO 7 K=1,3
      RMULK=RMULT(ISYM,K)
      RCK=RCOM(K)
      DCK=DCO(K)
      DIK=DELTA(I,K)
      DJK=DELTA(J,K)
      RIJK=RIRJ*RCK
      DUM=DIK*RCJ+DJK*RCI
      DIJRK=DIJ*RCK
      UU(I,K)=UU(I,K)+WD*(DUM-DIJRK*V3*RIJK)*RMULK
    7 TT(I,K)=TT(I,K)-WS*(DC*(DIJRK*V9*DUM-V8*RIJK)+RNV7*RCK*3. *RIRJ*DCK
     1+DCJ*DIK+DCI*DJK-V6*DCK*DIJ)*RMULK
    6 CONTINUE
      RETURN
      END
```

```
C
C     SUBROUTINE TO SUM UP DD EFFECTS ONLY
C
      SUBROUTINE SUMDD(SUM,DIS)
C
C     JACD
C
      COMMON KOD(220),IBC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),DLUMC(50,3),LIND(50),LUMPO(50,12),LKOD(50),LMOD(50),
     3  MINE(25,25),DZ(25,25),DX(25,25),DY(25,25),OZZT(25,25),
     4  AK(9,9),DK(9,9),DK(9,9),DMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIM(3,3),PRIM1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),UNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),DC(3),ISUB(3),XNUM(3),RCOM(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIM(8,8),NMULT(8,3),
     8  TRAN42,TRAN21,TRAN7,TRAN5,TRANL,TRANL,TRANDL,TRANDD,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,MM,NLSX,NLSY,NDIML,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NSLIP,NL,MODDD,NS,
     *  ZERO,ONE,IONE,I1,I2,I3,THIRD,HALF,TWO,NITER,OMEGA,DEL,PI,A,B,C,R,RSQ,EW,H,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,G,COND,CFAN,THIK,ESEAM,GSEAM,COH,PHI,
     2  IBFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELEL,NRECO,KODE,ISSW(15),ISW(15)
C
      DIMENSION SUM(3),DIS(3)
C
      IF(IBUF.GT.LIM)CALL DUMPO(IBUF,IDISK,BUF,ISTAR,2)
      DO 1 I=1,3
      DO 1 J=1,3
      IBUF=IBUF+1
    1 SUM(I)=SUM(I)+BUF(IBUF)*DIS(J)
      RETURN
      END
```

```
C
C     SUBROUTINE FOR B.E.  ELT-ELT SUMMATION.
C
      SUBROUTINE SUMELT(SUM,VECT,MOD1,KODE,N)
C
C     JACD
C
      COMMON MOD(230),ISC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),DLUMC(50,3),LIND(50),LUMPD(50,12),LKOD(50),LMOD(50),
     3  MINE(25,25),DZ(25,25),DX(25,25),DY(25,25),DZZT(25,25),
     4  AK(9,9),BK(9,9),DK(9,9),LMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIM(3,3),PRIM1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),DNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),DC(3),ISUD(3),XNUM(3),RCON(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIR(8,8),NMULT(8,8),
     8  TRAN42,TRAN21,TRAN7,TRAN5,TRANLL,TRANL,TRANDL,TRANDD,FAC,WGHT(11),COORD(11,3),
     9  NOUL,NELT,NLUMP,NN,MM,NLSX,NLSY,NDIML,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NSLIP,NIL,MODDD,NS,
     *  ZERO,ONE,IONE,I1,I2,I3,THIRD,HALF,TWO,NITER,OMEGA,DLL,PI,A,B,C,R,RSQ,EW,H,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CN,CTG,G,COND,CFAC,THIK,ESEAM,GSEAM,COIL,PHI,
     2  IUFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELLL,NRECO,KODE,ISSW(15),ISW(15)
C
C
      DIMENSION VECT(220,3),SUM(3),TRAC(3),DIS(3)
C
      DO 3 NL=1,NLUMP
      IF(LIND(NL).EQ.1)GO TO 3
      IF(LMOD(NL).NE.MOD1)GO TO 3
      NEL=LUMPD(NL)
      IF(LIND(NL).NE.2)GO TO 7
      CALL TOL(NL,TRAC,DIS)
      CALL SUMUP(SUM,DIS,TRAC)
      GO TO 3
    7 CONT
      DO      -,NEL
      N=LL.   (NL,NE)
      KODE=IABS(KOD(N))
      IF(IBUF.GT.LIM)CALL DUMPO(IBUF,IDISK,BUF,ISTAR,2)
      IF(KODE.NE.3)GO TO 508
      IF(KODE.NE.KODR)GO TO 500
      IF(M.EQ.N)GO TO 500
      IBUF=IBUF+11
      SUM(11)=SUM(11)-BUF(IBUF)*VECT(N,3)
      IBUF=IBUF+11
      SUM(12)=SUM(12)-BUF(IBUF)*VECT(N,3)
      SUM(13)=SUM(13)+BUF(IBUF-1)*VECT(N,1)+BUF(IBUF)*VECT(N,2)
      GO TO 502
  508 CONTINUE
      IF(KODE.LT.7)GO TO 500
      DO 501 I=1,3
      DO 501 J=1,3
      IBUF=IBUF+2
      BBU=BUF(IBUF-1)
  501 SUM(I)=SUM(I)+BUF(IBUF)*T(N,J)+BBU*U(N,J)
      GO TO 502
  500 DO 505 I=1,3
      DO 503 J=1,3
      IBUF=IBUF+1
  503 SUM(I)=SUM(I)+BUF(IBUF)*VECT(N,J)
  502 CONTINUE
    6 CONTINUE
    3 CONTINUE
```

```
      RETURN
      END
*****************************************************************************************************************************
C
C     SUMUP IS THE EQUIVALENCE OF LOAD
C
      SUBROUTINE SUMUP(SUM,DIS,TRAC)
C
C     JACD
C
      COMMON KOD(220),IBC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),DLUMC(50,3),LIND(50),LUMPO(50,12),LKOD(50),LMOD(50),
     3  MINE(25,25),DZ(25,25),DX(25,25),DY(25,25),OZZT(25,25),
     4  AK(9,9),BK(9,9),DK(9,9),BMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIN(3,3),PRIM1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),DNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),DC(3),ISUB(3),XNUM(3),RCOM(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIR(8,8),NMULT(8,3),
     8  TRAN42,TRAN21,TRAN7,TRANS,TRANLL,TRANL,TRANDL,TRANDU,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,MM,NLSX,NLSY,NDIML,NDIMN,NDIM,ISHAL,JSHAL,NUMOR,NSLIP,NIL,MODDD,NS,
     #  ZERO,ONE,TONE,I1,I2,I3,THIRD,HALF,TWO,NITER,ONEGA,DEL,PI,A,B,C,R,NSQ,EW,H,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,G,COND,CBAR,THIK,ESEAM,GSEAM,COH,PHI,
     2  IBFSZ,ICHAN,LIM,IDISK,IDUF,ISTAR,BUF(1152),IREAD,IRITE,NELEL,NREQD,KODE,ISSW(15),ISW(15)
C
      DIMENSION SUM(3),TRAC(3),DIS(3)
      IF(IDUF.GT.LIM)CALL DUMPO(IDUF,IDISK,BUF,ISTAR,2)
      DO 1 I=1,3
      DO 1 J=1,3
      IBUF=IBUF+2
    1 SUM(I)=SUM(I)+BUF(IBUF)*TRAC(J)+BUF(IBUF-1)*DIS(J)
      RETURN
      END
*****************************************************************************************************************************
C
C     SUBROUTINE TO FIX SYMMETRY OF A VECTOR
C
      SUBROUTINE SYMFIX(SYM,NMULT,VECT,IS)
      DIMENSION SYM(3),NMULT(8,3),VECT(3)
      DO 1 I=1,3
      IF(NMULT(IS,I).NE.1)VECT(I)=SYM(I)-VECT(I)
    1 CONTINUE
      RETURN
      END
*****************************************************************************************************************************
C
C     TRANSFORM A TENSOR FROM GLOBAL TO LOCAL COORDS.
C
      SUBROUTINE GTOL(T,A)
      DIMENSION T(3,3),A(3,3),S(3,3)
      DO 1 K=1,3
      DO 1 L=K,3
      S(K,L)=0.
      DO 1 I=1,3
      DO 1 J=1,3
    1 S(K,L)=S(K,L)+A(K,I)*A(L,J)*T(I,J)
```

```
      EQUIVALENCE (STRES,TT)
      RS=R*RSQ
      WCK=+W1*CK/KSQ
C           +W1 FOR COMP + CONVENTION
      WS=W1*CD/R3
      SUMD=0.
      SUMDR=0
      DO 1 I=1,3
      RCOM(I)=(XC(I)-XX(1))/R
      SUMD=SUMD+DIS(I)
    1 SUMDR=SUMDR+RCOM(I)*DIS(I)
      DRDN=RCOM(3)
      DIS3=DIS(3)
      D3=3.*DRDN
      WDRDN=WCK*DRDN
      V35=WCK*(V3*SUMDR*DRDN-DIS3)
      WV6=WS*(V6*DIS3-D3*SUMDR)
      WV72=WS*V7*SUMDR
      WDV9=WS*D3*V9
      WDV8=WS*(3.*DIS3-D3*SUMDR*V8)
C
      W(3)=W(3)+WCK*SUMDR
      WDRDN=WCK*DRDN
      DO 6 I=1,3
      RCI=RCOM(I)
      W(I)=W(I)+WDRDN*DIS(I)+RCI*V35
      SUM1=WDV9*DIS(I)+WDV8*RCI
      SUMW=WDV9*RCI
      DO 8 J=1,3
      RCJ=RCOM(J)
    8 STRES(I,J)=STRES(I,J)+RCJ*SUM1+SUMW*DIS(J)
      FAC13=WS*DIS(1)+WV72*RCI
      STRES(I,3)=STRES(I,3)+FAC13
      STRES(3,I)=STRES(3,I)+FAC13
    6 STRES(I,I)=STRES(I,I)-WV6
      RETURN
      END

*****************************************************************************************************************************

      SUBROUTINE SISW
C
C     JACD
C
      COMMON KOD(220),IBC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),DLUMC(50,3),LIND(50),LUMPO(50,12),LKOD(50),LMOD(50),
     3  MINE(25,25),DZ(25,25),DX(25,25),DY(25,25),OZZT(25,25),
     4  AK(9,9),BK(9,9),DK(9,9),EMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIN(3,3),PR1M1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),UNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),DC(3),ISUB(3),XNUM(3),RCOM(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIR(8,8),NMULT(8,3),
     8  TRAN42,TRAN21,TRAN7,TRAN5,TRANLL,TRANL,TRANDL,TRANDD,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,MM,NLSX,NLSY,NDIML,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NSLIP,NIL,MODDD,NS,
     *  ZERO,ONE,TONE,I1,I2,I3,THIRD,HALF,TWO,NITER,OMEGA,DEL,PI,A,B,C,R,RSQ,EW,H,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,G,COND,CFAR,THIK,ESEAM,GSEAM,COH,PHI,
     2  IBFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IWRITE,NELEL,NRECO,KODE,ISW(15),ISW(15)
C
      WRITE(IWRITE,100)(ISW(I),I=3,7)
  100 FORMAT(" SWITCH CONTROLS ARE (0=NO  1=YES   >1 IS SPECIFIED)",/
     3" ISW  3   :   TO READ CLOSURE VALUES FROM DISK",T65,I2,/
```

```
     4" ISW  4    :    TO READ RIDE VALUES FROM DISK",T65,I2,/
     5" ISW  5    :    TO SKIP CLOSURE CALCULATIONS",T65,I2,/
     6" ISW  6    :    TO SKIP RIDE CALCULATIONS",T65,I2,/
     7" ISW  7    :    = 2 TO SUPPRESS MINE LAYOUT",T65,I2,/
     *)
      WRITE(IRITE,101)ISW(8),ISW(9),ISW(15)
  101 FORMAT(" ISW  8    :    TO INCLUDE ON REEF CLOSURES AND RIDES",T65,I2,/
     9" ISW  9    :    TO INCLUDE ONREEF STRESSES",T65,I2,/
     5" ISW 15    :    FOR DEBUG MONITORING",T65,I2)
      RETURN
      END
*****************************************************************************************************************
C     JACD
C
      COMMON KOD(220),IBC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),DLUMC(50,3),LIND(50),LUMPO(50,12),LKOD(50),LMOD(50),
     3  MINE(25,25),DZ(25,25),DX(25,25),DY(25,25),OZZT(25,25),
     4  AK(9,9),BK(9,9),DK(9,9),BMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIM(3,3),PRIM1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),DNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),DC(3),ISUB(3),XNUM(3),RCOM(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIR(8,8),NMULT(8,3),
     8  TRAN42,TRAN21,TRAN7,TRAN5,TRANLL,TRANL,TRANDL,TRANDD,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,MM,NLSX,NLSY,NDIML,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NSLIP,NIL,MODDD,NS,
     *  ZERO,ONE,IONE,I1,I2,I3,THIRD,HALF,TWO,NITER,OMEGA,DEL,PI,A,B,C,R,RSQ,EW,H,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,G,COND,CFAR,THIK,ESEAM,GSEAM,COH,PHI,
     2  IBFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELEL,NRECO,KODE,ISSW(15),ISW(15)
C
C
      DIMENSION TDIAG(3),DIS(3),KIND(9,9),TRAC(3),RES(3),PRS(3),DCO(3,3)
      DIMENSION SUM(3),SUML(3),VECT(220,3),TEMP(3),UDIAG(3)
C
      RNUM=ISMAL*ISMAL
      CALL OPEN(3,"UUTT.",2,IER)
      IF(IER.NE.1)TYPE "UUTT ER",IER
      IF(ISSW(2).EQ.0)GO TO 444
      DO 446 N=1,NELT
  446 READ BINARY (3)(T(N,I),I=1,3),(U(N,I),I=1,3)
      DO 445 I=1,NN
  445 READ BINARY (3) ((DX(I,J),DY(I,J),DZ(I,J)),J=1,MM)
      READ BINARY (3) OZZT
      IF(ISSW(2).NE.2)GO TO 447
  444 CONTINUE
      REWIND 3
      CALL TTT(IRITE)
      CALL OPEN(1,"DIAGEL.",2,IER)
      IF(IER.NE.1)TYPE "DIAGEL ER",IER
      CALL OPEN(2,"LINDD.",2,IER)
      IF(IER.NE.1)TYPE "LINDD ER",IER
      CALL OPEN(6,"LINDEX.",2,IER)
      IF(IER.NE.1)TYPE "LINDEX ER",IER
      CALL OPEN(7,"BVECT.",2,IER)
      IF(IER.NE.1)TYPE "BVECT ER",IER
      CALL OPEN(8,"COSINE.",2,IER)
      IF(IER.NE.1)TYPE "COS OP ER",IER
      CALL OPEN(13,"BVECDD",2,IER)
      IF(IER.NE.1)TYPE "BVECDD OP ER",IER
C
      TENS=ZERO
```

```
      IF(ISSW(14).EQ.1)TENS=9999.
      NIL=0
      IDISK=NIL
      DO 100 NUM=1,NITER
      NIL=0
      AMAX=ZERO
      IBUF=NIL
      ISTAR=1CHAN
      JOUT=NIL
      CALL DUMPO(IBUF,IDISK,BUF,ISTAR,2)
      CALL LUMPR
C
C     SET VECTOR OF UNKNOWNS.
C
      DO 506 N=1,NELT
      DO 506 I=1,3
      IBB=IBC(N,I)
      IF(IBB.NE.1)GO TO 507
      VECT(N,I)=U(N,I)
      GO TO 506
  507 VECT(N,I)=T(N,I)
  506 CONTINUE
C
C     B.E.  SECTION
C
      DO 511 NL=1,NLUMP
      READ BINARY (6) LIND,KIND
C 102 FORMAT(80I1)
      MEL=LMOD(NL)
      CALL NULL(SUML,FALL)
C
C     LUMP-LUMP
C
      DO 19 NL=1,NLUMP
      IF(LIND(NL).NE.1)GO TO 19
      CALL TUL(NL,TRAC,DIS)
      CALL SUMUP(SUML,DIS,TRAC)
   19 CONTINUE
      DO 70 K=1,NLSX
      DO 70 L=1,NLSY
      IF(KIND(K,L).NE.1)GO TO 70
      DIS(1)=XLARG(K,L)
      DIS(2)=YLARG(K,L)
      DIS(3)=ZLARG(K,L)
      CALL SUMDD(SUML,DIS)
   70 CONTINUE
C
C     CALCULATE DETAILED SUMS.
C
      MOD1=LMOD(NL)
      DO 1 ME=1,MEL
      M=LUMPO(NL,ME)
      READ BINARY (1) UD1AG
      READ BINARY (7) TEMP
      KODR=IABS(KOD(M))
      DO 33 I=1,3
   33 SUM(I)=SUML(I)/RNUM+TEMP(I)
C
C     LUMP-ELT
C
```

```
C
C       ELT=ELT
C
        CALL SUMELT(SUM,VECT,MOD1,KODR,M)
C
C       ADD CONTRIBUTION FROM DD ELEMENTS.
C
        IF(MOD1.NE.MODDD)GO TO 63
        DO 60 K=1,NLSX
        DO 60 L=1,NLSY
        IF(KIND(K,L).NE.2)GO TO 61
        DIS(1)=XLARG(K,L)/RNUM
        DIS(2)=YLARG(K,L)/RNUM
        DIS(3)=ZLARG(K,L)/RNUM
        CALL SUMDD(SUM,DIS)
        GO TO 60
   61   CONTINUE
        IF(KIND(K,L).EQ.1)GO TO 60
C       ##########################
        KB=(K-1)*ISMAL+IONE
        KE=K*ISMAL
        LB=(L-1)*JSMAL+IONE
        LE=L*JSMAL
        DO 62 I=KB,KE
        DO 62 J=LB,LE
        DIS(1)=DX(I,J)
        DIS(2)=DY(I,J)
        DIS(3)=DZ(I,J)
   62   CALL SUMDD(SUM,DIS)
   60   CONTINUE
   63   CONTINUE
        IFF=IFACE(M)
        DO 14 I=1,3
        IBB=IBC(M,I)
        IF(IBB.NE.1)GO TO 504
        DIFF=OMEGA*SUM(I)/HALF
        VECT(M,I)=VECT(M,I)+DIFF
        U(M,I)=U(M,I)+DIFF
        IF(KODR.NE.7)GO TO 505
        U(IFF,I)=U(M,I)
        GO TO 505
  504   IF(IBB.NE.2)GO TO 505
        DIFF=OMEGA*SUM(I)/UDIAG(I)
        T(M,I)=T(M,I)-DIFF
        VECT(M,I)=VECT(M,I)-DIFF
        IF(KODR.NE.7)GO TO 505
        T(IFF,I)=-T(M,I)
  505   CONTINUE
        ABSUM=ABS(SUM(I))
        IF(ABSUM.GT.AMAX)AMAX=ABSUM
        IF(DEL.GE.ABSUM)GO TO 10
        IF(IBB.GT.2)GO TO 10
        JOUT=JOUT+1
        IF(JOUT.LT.5)WRITE(10,20) U(M,I),T(M,I),ABSUM,M,I
   20   FORMAT(1X,3(F16.5,1X),2I4)
   10   CONTINUE
   14   CONTINUE
        IF(NUM.LT.NSLIP)GO TO 1
        READ BINARY (3) DCO,W
C
```

```
C
C     KODE  7 = INTACT
C     KODE  8 = TRACTION FREE SURFACE.
C     KODE  9 = MINED WITH COMPLETE CLOSURE RESTRICTION.
C     KODE 11 = SHEAR FAILED INTERFACE.
C     KODE 10 = TENSILE FAILED INTERFACE.
C
      IF(KODR LT. 7. OR. KODR. EQ. 8)GO TO 1
C
C     LOAD TDIAG AND UDIAG
C
      DO 57 I=1,3
      UDIAG(I)=-SUM(I)*OMEGA/UDIAG(I)
   57 TDIAG(I)=SUM(I)*OMEGA/DALF
C
C     CALCULATE TRAC, DIS, AND PRS
C
      DO 41 I=1,3
      PRS(I)=ZERO
      TRAC(I)=ZERO
      DIS(I)=ZERO
      SUMT=ZERO
      SUMU=ZERO
      DO 53 J=1,3
      DCI=DCO(I,J)
      SUMU=SUMU+DCI*UDIAG(J)
      SUMT=SUMT+DCI*TDIAG(J)
      PRS(I)=PRS(I)+DCI*W(J)
      TRAC(I)=TRAC(I)+DCI*T(M,J)
   53 DIS(I)=DIS(I)+UCI*(U(M,J)-U(IFF,J))
C
C     CHANGE IBC 3 AND 4 UNKNOWNS.
C
      IBB=IBC(M,I)
      IF(IBB. EQ. 3)DIS(I)=DIS(I)+SUMT
      IF(IBB. EQ. 4)TRAC(I)=TRAC(I)+SUMU
      RES(I)=TRAC(I)-PRS(I)
   41 CONTINUE
C
C     CHECK FOR FAILURE.
C
      IF(KOUR. EQ. 9)GO TO 42
      IF(KOUR. EQ. 10)GO TO 47
      SHSTR=COH-PHI*RES(3)
      SHEAR=SQRT(RES(1)**2+RES(2)**2)
      IF(KOUR. EQ. 11)GO TO 43
      IF(SHEAR. GT. SHSTR)GO TO 43
      GO TO 45
   43 IF((RES(3)). GT. TENS)GO TO 44
C
C     SHEAR FAILURE
C
      IBC(M,1)=3
      IBC(M,2)=3
      IBC(M,3)=4
      DIS(3)=ZERO
   54 TRAC(1)=(RES(1)*SHSTR/SHEAR)+PRS(1)
      TRAC(2)=(RES(2)*SHSTR/SHEAR)+PRS(2)
      KOUR=11
      TYPE "S",M
      GO TO 45
```

```
C
C     TENSILE FAILURE
C
   44 CONTINUE
      KODR=10
      DO 46 I=1,3
      TRAC(I)=PRS(I)
   46 IBC(M,I)=3
      TYPE "I",M
      GO TO 45
C
C     TOTAL CLOSURE ELEMENTS
C
   42 CONTINUE
      IF(DIS(3).LT.THIK)GO TO 45
      IBC(M,1)=3
      IBC(M,2)=3
      DIS(3)=THIK
      IBC(M,3)=4
      GO TO 45
   47 CONTINUE
C
C     CHECK FOR CLOSURE OF AN OPEN FAULT ELEMENT.
C
      IF(DIS(3).LT.ZERO)GO TO 45
      TYPE "CLOS-F",M
      DIS(3)=ZERO
      IBC(M,3)=4
      IBC(IFF,3)=4
      IF(RES(3).GT.ZERO)GO TO 45
      IBC(M,3)=3
      IBC(IFF,3)=3
      TRAC(3)=PRS(3)
   45 CONTINUE
C
C     BACK TRANSFORM TRAC AND DIS
C
      DO 50 J=1,3
      T(M,J)=ZERO
      DNEW=ZERO
      DO 51 I=1,3
      DNEW=DNEW+DCO(I,J)*DIS(I)
   51 T(M,J)=T(M,J)+DCO(I,J)*TRAC(I)
      T(IFF,J)=-T(M,J)
      U(M,J)=U(IFF,J)+DNEW
      IBB=IBC(M,J)
      IF(IBB.GT.2)IBC(IFF,J)=IBB
   50 CONTINUE
      N4=INDEX(M,4)
      IF(N4.NE.NIL)KODR=-KODR
      KOD(M)=KODR
      KOD(IFF)=KODR
    1 CONTINUE
      IBM=IBUF-2
      IF(NUM.EQ.1)WRITE(IRITE,200)ISTAR,IBUF,(BUF(J),J=IBM,IBUF),LIND
  200 FORMAT(216,3F10.6,1X,6011)
  511 CONTINUE
      CALL DDSOL(JOUT,VECT,AMAX)
      REWIND 8
      REWIND 6
```

```
      REWIND 7
      REWIND 1
      REWIND 2
      REWIND 13
      WRITE(IRITE,448)NUM,JOUT,AMAX
      IF(JOUT.EQ.NIL.OR.AMAX.LT.DEL)GO TO 101
  448 FORMAT(1X,I4" ITS",I8" ELTS OUT","  MAX ER =",F10.5)
      IF(NUM.EQ.1)WRITE(IRITE,449)ISTAR
  449 FORMAT(/" LAST BLOCK ACCESSED ="I4,/)
  100 CONTINUE
      NUM=NUM-1
  101 CONTINUE
      IF(JOUT.NE.0)WRITE(IRITE,121)NUM,JOUT,AMAX
  121 FORMAT(" NO CONVERGENCE AFTER"I4" ITERATIONS"I8" ELTS OUT"
     14X"MAX DIFF ="F16.9)
      DO 450 N=1,NELT
  450 WRITE BINARY (3) (T(N,L),L=1,3),(U(N,L),L=1,3)
      DO 451 I=1,NN
  451 WRITE BINARY (3) ((DX(I,J),DY(I,J),DZ(I,J)),J=1,MM)
      WRITE BINARY (3) OZZT
      CALL EXEC(5,1)
      CALL TTT(IRITE)
  447 CONTINUE
      CALL BACK
      STOP
      END
```

****

```
C
C     SUBROUTINE TO CALCULATE S AND D COEFFICIENTS.
C
      SUBROUTINE SSDD(XR,XE,DCO,W1)
C     JACD
C
      COMMON KOD(220),IBC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),DLUMC(50,3),LIND(50),LUMPO(50,12),LKOD(50),LMOD(50),
     3  MINE(25,25),DZ(25,25),DX(25,25),DY(25,25),OZZT(25,25),
     4  AK(9,9),BK(9,9),DK(9,9),EMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIM(3,3),PRIM1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),DNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),DC(3),ISUB(3),XNUM(3),RCOM(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIR(8,8),NMULT(8,3),
     8  TRAN42,TRANZ1,TRAN7,TRAN5,TRANLL,TRANL,TRANDL,TRANDD,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,MM,NLSX,NLSY,NDIML,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NSLIP,NIL,MODDD,NS,
     *  ZERO,ONE,IONE,I1,IZ,I3,THIRD,HALF,TWO,NITER,OMEGA,DELL,PI,A,B,C,R,RSQ,EW,H,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,S,COND,CFAR,TRIK,ESEAM,USEAM,COH,PHI,
     2  IBFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELEL,NRECO,KODE,ISSW(15),ISW(15)
C
      DIMENSION XR(3),XE(3),DCO(3)
      R3=R*RSQ
      WCK=-W1*CK/RSQ
      WCR=W1*CONS/R
      WS=W1*CD/R3
      WD=-WCK
      DRDN=ZERO
      DO 5 L=1,3
      RCOM(L)=(XE(L)-XR(L))/R
    5 DRDN=DRDN+RCOM(L)*DCO(L)
      D3=DRDN*3.
      DO 6 I=1,3
```

```
      RCI=RCOM(I)
      RIV9=RCI*V9
      RIV2=RCI*V2
      DCI=DCO(I)
      RI3=RCI*3.
      J=3
      RCJ=RCOM(J)
      DCJ=DCO(J)
      DIJ=DELTA(I,J)
      RIRJ=RCI*RCJ
      RNV7=V7*(DCI*RCJ+DCJ*RCI)
      DO 7 K=1,3
      RMULK=RMULT(ISYM,K)
      RCK=RCOM(K)
      DCK=DCO(K)
      DIK=DELTA(I,K)
      DJK=DELTA(J,K)
      RIJK=RIRJ*RCK
      DUM=DIK*RCJ+DJK*RCI
      DIJRK=DIJ*RCK
      UU(I,K)=UU(I,K)+WD*(DUM-DIJRK+V3*RIJK)*RMULK
    7 TT(I,K)=TT(I,K)-WS*(DS*(DIJRK+V9*DUM-V8*RIJK)+RNV7*RCK+3.*RIRJ*DCK
     1+DCJ*DIK+DCI*DJK-V6*DCK*DIJ)*RMULK
    6 CONTINUE
      RETURN
      END
*****************************************************************************************************************************
C
C     SUBROUTINE TO SUM UP DD EFFECTS ONLY
C
      SUBROUTINE SUMDD(SUM,DIS)
C
C     JACD
C
      COMMON KOD(220),IBC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),BLUMC(50,3),LIND(50),LUMPO(50,12),LKOD(50),LMOD(50),
     3  NINE(25,25),D2(25,25),DX(25,25),DY(25,25),DZZT(25,25),
     4  AK(9,9),DK(9,9),DK(9,9),BMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIM(3,3),PRIM1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),DNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),DC(3),ISUB(3),XNUM(3),RCOM(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,BND,ISYM,NSYM(10),NDIR(8,8),NMULT(8,3),
     8  TRAN42,TRAN21,TRAN7,TRANS,TRANLL,TRANL,TRANDL,TRANDQ,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,MM,NLSX,NLSY,NDIML,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NSLIP,NL,MODDD,NS,
     *  ZERO,ONE,IONE,I1,I2,I3,THIRD,HALF,TWO,NITER,OMEGA,DEL,PI,A,B,C,R,RSQ,EW,H,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,G,COND,CPAR,THIK,ESEAM,GSEAM,COH,PHI,
     2  IDFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELEL,NRECO,KODE,ISSU(15),ISW(15)
C
      DIMENSION SUM(3),DIS(3)
C
      IF(IBUF.GT.LIM)CALL DUMPO(IBUF,IDISK,BUF,ISTAR,2)
      DO 1 I=1,3
      DO 1 J=1,3
      IBUF=IBUF+1
    1 SUM(I)=SUM(I)+BUF(IBUF)*DIS(J)
      RETURN
      END
*****************************************************************************************************************************
```

```
C
C     SUBROUTINE FOR B.E.  ELT-ELT SUMMATION.
C
      SUBROUTINE SUMEL1(SUM,VECT,MOD1,KODR,N)
C
C     JAUD
C
      COMMON MOD(220),IBC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),DLUMC(50,3),LIND(50),LUMPO(50,12),LKOD(50),LMOD(50),
     3  MINE(25,25),DZ(25,25),DX(25,25),DY(25,25),OZZT(25,25),
     4  AK(9,9),BK(9,9),DK(9,9),BMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIM(3,3),PRIM1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),DNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),DC(3),ISUB(3),XNUM(3),RCON(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIR(8,8),NMULT(8,8),
     8  TRAN42,TRAN21,TRAN7,TRAN5,TRANLL,TRANL,TRANDL,TRANDO,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,NM,NLSX,NLSY,NDIML,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NSLIP,N1L,MODDD,NS,
     *  ZERO,ONE,TONE,I1,I2,I3,THIRD,HALF,TWO,NITER,OMEGA,DEL,PI,A,B,C,R,RSQ,EW,H,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,G,COND,CFAR,THIK,ESEAM,GSEAM,COH,PHI,
     2  IEFS2,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELLL,NRECO,KODE,ISSW(15),ISW(15)
C
C
      DIMENSION VECT(220,3),SUM(3),TRAC(3),DIS(3)
C
      DO 3 NL=1,NLUMP
      IF(LIND(NL).EQ.1)GO TO 3
      IF(LMOD(NL).NE.MOD1)GO TO 3
      NEL=LKOD(NL)
      IF(LIND(NL).NE.2)GO TO 7
      CALL TUL(NL,TRAC,DIS)
      CALL SUMUP(SUM,DIS,TRAC)
      GO TO 3
    7 CONTINUE
      DO 6 NE=1,NEL
      N=LUMPO(NL,NE)
      KODE=IABS(KOD(N))
      IF(IBUF.GT.LIM)CALL DUMPO(IBUF,IDISK,BUF,ISTAR,2)
      IF(KODE.NE.3)GO TO 508
      IF(KODE.NE.KODR)GO TO 500
      IF(N.EQ.N)GO TO 500
      IBUF=IBUF+11
      SUM(I1)=SUM(I1)-BUF(IBUF)*VECT(N,3)
      IBUF=IBUF+11
      SUM(I2)=SUM(I2)-BUF(IBUF)*VEC(N,3)
      SUM(I3)=SUM(I3)+BUF(IBUF-1)*VECT(N,1)+BUF(IBUF)*VECT(N,2)
      GO TO 502
  508 CONTINUE
      IF(KODE.LT.7)GO TO 500
      DO 501 I=1,3
      DO 501 J=1,3
      IBUF=IBUF+2
      BBU=BUF(IBUF-1)
  501 SUM(I)=SUM(I)+BUF(IBUF)*T(N,J)+BBU*U(N,J)
      GO TO 502
  500 DO 503 I=1,3
      DO 503 J=1,3
      IBUF=IBUF+1
  503 SUM(I)=SUM(I)+BUF(IBUF)*VECT(N,J)
  502 CONTINUE
    6 CONTINUE
    3 CONTINUE
```

```
      RETURN
      END
*****************************************************************************************************************************
C
C     SUMUP IS THE EQUIVALENCE OF LOAD
C
      SUBROUTINE SUMUP(SUM,DIS,TRAC)
C
C     JACD
C
      COMMON NOD(220),IBC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),DLUMC(50,3),LIND(50),LUMPO(50,12),LROD(50),LMOD(50),
     3  MINE(25,25),DZ(25,25),DX(25,25),DY(25,25),OZZT(25,25),
     4  AI(9,9),BI(9,9),DK(9,9),DMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIN(3,3),PRIM1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),DNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),BC(3),ISUB(3),XNUM(3),RCON(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIR(8,8),NMULT(8,3),
     8  TRAN42,TRAN21,TRAN7,TRAN5,TRANLL,TRANL,TRANDL,TRANDD,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,MM,NLSX,NLSY,NDIML,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NSLIP,NIL,MODUD,NS,
     *  ZERO,ONE,TONE,I1,I2,I3,THIRD,HALF,TWO,NITER,OMEGA,DEL,PI,A,B,C,R,RSQ,ED,H,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,G,COND,CFAR,TIME,ESEAM,GSEAM,COH,PHI,
     2  IUF52,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELEL,NREGO,KODE,ISSW(15),ISW(15)
C
      DIMENSION SUM(3),TRAC(3),DIS(3)
      IF(IBUF.GT.LIM)CALL DUMPO(IBUF,IDISK,BUF,ISTAR,2)
      DO 1 I=1,3
      DO 1 J=1,3
      IBUF=IBUF+2
    1 SUM(I)=SUM(I)+BUF(IBUF)*TRAC(J)+BUF(IBUF-1)*DIS(J)
      RETURN
      END
*****************************************************************************************************************************
C
C     SUBROUTINE TO FIX SYMMETRY OF A VECTOR
C
      SUBROUTINE SYMFIX(SYM,NMULT,VECT,IS)
      DIMENSION SYM(3),NMULT(8,3),VECT(3)
      DO 1 I=1,3
      IF(NMULT(IS,I).NE.1)VECT(I)=SYM(I)-VECT(I)
    1 CONTINUE
      RETURN
      END
*****************************************************************************************************************************
C
C     TRANSFORM A TENSOR FROM GLOBAL TO LOCAL COORDS.
C
      SUBROUTINE IGTOL(T,A)
      DIMENSION T(3,3),A(3,3),S(3,3)
      DO 1 K=1,3
      DO 1 L=K,3
      S(K,L)=0.
      DO 1 I=1,3
      DO 1 J=1,3
    1 S(K,L)=S(K,L)+A(K,I)*A(L,J)*T(I,J)
```

```
      DO 2 K=1,3
      DO 2 L=K,3
      T(L,K)=S(K,L)
    2 T(K,L)=S(K,L)
      RETURN
      END
*****************************************************************************************************************************
      SUBROUTINE TTUU(XED,XRD,WCK,WCR,DCO)
C
C     JACD
C
      COMMON KOD(220),IBC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),DLUMC(50,3),LIND(50),LUMPO(50,12),LFOD(50),LMOD(50),
     3  MINE(25,25),DZ(25,25),DX(25,25),DY(25,25),DZZT(25,25),
     4  AK(9,9),BK(9,9),DK(9,9),EMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIM(3,3),PRIM1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),DNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),DC(3),ISUB(3),XNUM(3),RCOM(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,NHO,ISYM,NSYM(10),NDIR(8,8),NMULT(8,3),
     8  TRAN42,TRAN21,TRAN7,TRAN5,TRANLL,TRANL,TRANDL,TRANDD,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,NM,NLSX,NLSY,ND1ML,ND1MN,ND1M,ISMAL,JSMAL,NUMOR,NSLIP,NIL,MODDD,NS,
     *  ZERO,ONE,TONE,11,12,13,THIRD,HALF,TWO,NITER,ONEGA,DEL,PI,A,B,C,K,RSQ,EW,H,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,G,COND,CFAR,THIK,ESEAM,GSEAM,COH,PHI,
     2  IDFS7,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELEL,NRLCO,KODE,ISSW(15),ISW(15)
C
      DIMENSION XED(3),XRD(3),DCO(3)
C
C     SUBROUTINE TO CALCULATE TT AND UU COEFFS.
C
      DRDN=ZERO
      DO 1 K=1,3
      RCOM(K)=(XED(K)-XRD(K))/R
    1 DRDN=DRDN+RCOM(K)*DCO(K)
      DRDN=DRDN*WCK
      DO 2 I=1,3
      RMULI=NMULT(ISYM,I)
      RCI=RCOM(I)
      RIV3=RCI*V3
      RIV2=RCI*V2
      DCI=DCO(I)
      TT(I,I)=TT(I,I)+DRDN*(1.+RIV3*RCI)*RMULI
      UU(I,I)=UU(I,I)+WCR*(V4+RIV2*RCI)*RMULI
      IF(I.EQ.3)RETURN
      IPI=I+1
      DO 2 J=IPI,3
      RCJ=RCOM(J)
      DCJ=DCO(J)
      RMULJ=NMULT(ISYM,J)
      TF=(DCI*RCJ-DCJ*RCI)*WCK
      TFUNC=DRDN*RIV3*RCJ
      TT(I,J)=TT(I,J)+(TFUNC+TF)*RMULJ
      TT(J,I)=TT(J,I)+(TFUNC-TF)*RMULI
      FAC=WCR*RIV2*RCJ
      UU(I,J)=UU(I,J)+FAC*RMULJ
    2 UU(J,I)=UU(J,I)+FAC*RMULI
      RETURN
      END
*****************************************************************************************************************************
```

```
C
C     SUBROUTINE TO SET TT AND UU TO ZERO
C
      SUBROUTINE TUO(TT,UU)
      DIMENSION TT(9),UU(9)
      DO 1 I=1,9
      TT(I)=0
    1 UU(I)=0
      RETURN
      END
```

```
      SUBROUTINE TUL(N,TRAC,DIS)
C
C     JACD
C
      COMMON KOD(220),IBC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),DLUMC(50,3),LIND(50),LUMPO(50,12),LKOD(50),LMOD(50),
     3  MINE(25,25),DZ(25,25),DX(25,25),DY(25,25),DZZT(25,25),
     4  AK(9,9),DR(9,9),DK(9,9),DMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIM(3,3),PRIM1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),DNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),DC(3),ISUB(3),XNUM(3),RCOM(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIR(8,8),NMULT(8,3),
     8  TRAN42,TRAN21,TRAN7,TRANS,TRANLL,TRANL,TRANDL,TRANDD,FAL,WGHT(11),COORD(11,3),
     9  NOUE,NELT,NLUMP,NN,MM,NLSX,NLSY,NGIML,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NGLIP,NIL,MODDD,NS,
     *  ZERO,ONE,TONE,I1,I2,I3,THIRD,HALF,TWO,NINTH,OMEGA,DEL,PI,A,B,C,R,RSQ,EU,H,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,G,COND,CFAR,TINK,ESEAM,GSEAM,COH,PHI,
     2  IBFSZ,ICHAN,LIM,IDISK,IIUF,ISTAR,BUF(1152),IREAD,IRITE,NELEL,NRELO,KODE,ISSW(15),ISW(15)
C
C
      DIMENSION TRAC(3),DIS(3)
      NUMEL=LKOD(N)
      AR=ALUMP(N)
      DO 10 I=1,3
      DIS(I)=ZERO
   10 TRAC(I)=ZERO
      DO 1 NE=1,NUMEL
      NEL=LUMPO(N,NE)
      ARE=AREA(NEL)/AR
      DO 1 I=1,3
      DIS(I)=DIS(I)+U(NEL,I)*ARE
    1 TRAC(I)=TRAC(I)+T(NEL,I)*ARE
      RETURN
      END
```

```
      FUNCTION TVAL(B1,R1,I,J)
C
C     JACD
C
      COMMON KOD(220),IBC(220,3),T(220,3),U(220,3),AREA(220),IFACE(220),INDEX(220,4),
     2  X(220,3),ALUMP(50),XLUMP(50,3),DLUMC(50,3),LIND(50),LUMPO(50,12),LKOD(50),LMOD(50),
     3  MINE(25,25),DZ(25,25),DX(25,25),DY(25,25),DZZT(25,25),
     4  AK(9,9),DR(9,9),DK(9,9),DMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIM(3,3),PRIM1(3,3),ALPHA(3,3),UU(3,3),TT(3,3),DNORM(3,3),DELTA(3,3),ALT(3,3,3),
     6  E1(3),E2(3),W(3),DC(3),ISUB(3),XNUM(3),RCOM(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIR(8,8),NMULT(8,3),
```

```
     8  TRAN42, TRAN21, TRAN7, TRAN9, TRANLL, TRANL, TRANDL, TRANDD, FAC, WGHT(11), CORD(11,3),
     9  NODE, NELT, NLOMP, NN, MM, NLSX, NLSY, NDIML, NDIMN, NDIM, ISMAL, JSMAL, NUMC, NLIP, NIL, MODDD, NS,
     *  ZERO, ONE, TONE, I1, I2, I3, THIRD, HALF, TWO, NINTH, OMEGA, DEL, PI, A, B, C, R, RSQ, CW, H, IL,
     1  E, V, V1, V2, V3, V4, V5, V6, V7, V8, V9, CONS, CK, CD, G, CONE, CTAN, THIK, ESEAM, GSEAM, COIL, PHI,
     2  IBFSZ, ICHAN, LIM, IDISK, IBUF, ISTAR, BUF(1152), IREAD, IRITE, NCLEL, NRECO, KODE, ISSW(15), ISW(15)
      C
       TVAL=ZERO
       DO 1 K=1,3
       IF(ALT(I,J,K).EQ.ZERO)GO TO 1
       TVAL=TVAL+ALT(I,J,K)*E1(K)*(ALOG(D1*R1))
     1 CONTINUE
       TVAL=-TVAL*CK
       RETURN
       END

*****************************************************************************************************************************

      C
      C     TRANSFORM A VECTOR FROM GLOBAL TO LOCAL COORDS.
      C
            SUBROUTINE VGTOL(X,A,O)
            DIMENSION X(3),A(3,3),O(3),V(3)
            DO 1 K=1,3
            V(K)=0.
            DO 1 I=1,3
          1 V(K)=V(K)+A(K,I)*(X(I)-O(I))
            DO 2 K=1,3
          2 X(K)=V(K)
            RETURN
            END

*****************************************************************************************************************************

      C
      C     TRANSFORM A VECTOR FROM LOCAL TO GLOBAL COORDS.
      C
            SUBROUTINE VLTOG(X,A,O)
            DIMENSION X(3),A(3,3),O(3),V(3)
            DO 1 I=1,3
            V(I)=O(I)
            DO 1 K=1,3
          1 V(I)=V(I)+A(K,I)*X(K)
            DO 2 I=1,3
          2 X(I)=V(I)
            RETURN
            END

*****************************************************************************************************************************
```

APPENDIX 4

PARTIAL LISTING OF PROGRAMME MINAPH

```
C
C     ------------------------------------------------------------
C     PARTIAL SOURCE LISTING FOR PROGRAM MINAP
C
C     PROGRAM WRITTEN BY S.L. CROUCH (1976)
C     PROGRAM MODIFIED BY J.A.C. DIERING (1980/81)
C
C     ONLY PORTIONS RELEVANT TO THE USE OF SUBREGIONS INCLUDED HERE
C     ------------------------------------------------------------
C
C
C     DEFINE LOCATIONS, SIZES, ORIENTATIONS AND BOUNDARY CONDITIONS OF
C     DISPLACEMENT DISCONTINUITIES.
C     ALSO SET MATERIAL PROPERTIES FOR DIFFERENT SUBREGIONS
C
      IF(ISSW(3).EQ.0)WRITE(IRITE,39)
      IF(ISSW(3).EQ.0)WRITE(IRITE,40)
      NUMBD=0
      NBS=NUMDS
      DO 115 N=1,NUMDS
      READ(IFILE)NUM,XB,YB,XE,YE,KODE,TN,TS,THIK
      READ(IFILE) ESEAM,GSEAM,COHESN,PHI,MED1,MED2
      IF(KODE.EQ.9)STOP KODE 9 AFTER STEP 1 ONLY
      IF(KODE.EQ.13)STOP KODE 13 ILLEGAL
      IF(MED1.NE.MED2)NBS=NBS+1
      XD=(XE-XB)/NUM
      YD=(YE-YB)/NUM
      SW=SQRT(XD*XD+YD*YD)
      DO 115 NI=1,NUM
      NUMBD=NUMBD+1
      M=NUMBD
      L(M)=N
      X(M)=XB+0.5*(2.*NI-1.)*XD
      Y(M)=YB+0.5*(2.*NI-1.)*YD
      A(N)=0.5*SW
      SINALP(N)=YD/SW
      COSALP(N)=XD/SW
C     KODE 15 REPRESENTS AN INTERFACE
      IF(KODE.EQ.15)KODE=-1
      KOD(M)=KODE
      MED(M)=MED1
      BN(M)=TN
      BS(M)=TS
      THICK(N)=THIK
      EEL(N)=ESEAM
      GEL(N)=GSEAM
      COHES(M)=COHESN
      TANPHI(N)=TAN(PHI*PI/180.)
C     MED1 AND MED2 ARE SUBREGION NUMBERS
      IF(MED1.EQ.MED2)GO TO 115
      ISLIP=30
      NUMBD=NUMBD+1
      NN=NUMBD
      L(NN)=NBS
```

```
C
C     TYPE "NN,M",NN,M
C     IFACE CONTAINS THE SECOND ELEMENT IN AN INTERFACE
      IFACE(M)=NN
      IFACE(NN)=M
      X(NN)=X(M)
      Y(NN)=Y(M)
      A(NBS)=A(N)
      SINALF(NBS)=-SINALF(N)
      COSALF(NBS)=-COSALF(N)
C
C     KOD = -1 FOR STRESS SPECIFICATION AT AN INTERFACE
C     KOD = -2 FOR DISPLACEMENT SPECIFICATION AT AN INTERFACE
C
      KOD(NN)=-2
      IF(KOD(M).NE.-1)KOD(NN)=KOD(M)
      MED(NN)=MED2
      BN(NN)=ZERO
      BS(NN)=ZERO
      THICK(NBS)=THIK
      EEL(NBS)=ESEAM
      GEL(NBS)=GSEAM
      COHES(NN)=COHESN
      TANPHI(NBS)=TANPHI(N)
  115 CONTINUE
C
C
*****************************************************************************************************************************
C
C     COMPUTE INFLUENCE COEFFICIENTS,
C     BUT BE SURE TO INCLUDE ELEMENTS IN THE CURRENT SUBREGION
C
      DO 237 I=1,NUMBD
      TYPE "I =",I,ISTAR
      XI=X(I)
      YI=Y(I)
      KODE=KOD(I)
      II=L(I)
      CALI=COSALF(II)
      SALI=SINALF(II)
      SALI2=SALI*SALI
      CALI2=CALI*CALI
      SCAL=SALI*CALI
      MEDR=MED(I)
      CALL MODSET(E,V,INF,ISYM,NSYM,MEDR)
      DO 237 J=1,NUMBD
      IF(I.NE.J.AND.ISSW(1).EQ.1)GO TO 237
      JJ=L(J)
      MEDE=MED(J)
C     TYPE MEDR,MEDE,NSYM
      IF(MEDR.NE.MEDE)GO TO 237
*****************************************************************************************************************************
C
C     LOAD BUF FOR INTERFACE CODES,
C     BUF IS THE MAIN BUFFER STORAGE ARRAY FOR INFLUENCE COEFFICIENTS
C
      IF(KODE.NE.-1)GO TO 226
```

```
      BUF(NBUF+2)= DXN*CALI+DYN*SALI
      BUF(NBUF+3)= DXS*CALI+DYS*SALI
      BUF(NBUF+1)=-DXS*SALI+DYS*CALI
      BUF(NBUF)=  -DXN*SALI+DYN*CALI
      BNN=BUF(NBUF)
      BSS=BUF(NBUF+3)
      NBUF=NBUF+4
      GO TO 244
  226 CONTINUE
      IF(KODE.NE.-2)GO TO 244
      BUF(NBUF+2)=(YYN-TXXN)*SCAL+TXYN*(CALI2-SALI2)
      BUF(NBUF+3)=(YYS-TXXS)*SCAL+TXYS*(CALI2-SALI2)
      BUF(NBUF+1)=TXXS*SALI2-2.*TXYS*SCAL+TYYS*CALI2
      BUF(NBUF)  =TXXN*SALI2-2.*TXYN*SCAL+TYYN*CALI2
      BNN=BUF(NBUF)
      BSS=BUF(NBUF+3)
      NBUF=NBUF+4
  244 CONTINUE
      IF(NBUF.LT.LIM)GO TO 238
C     STORE BUF ON DISK
      CALL WRBLK(1,ISTAR,BUF,NBLOK,IER)
      IF(IER.NE.1)TYPE "WRBLK",IER
      ISTAR=ISTAR+NBLOK
      TYPE ISTAR,I,J
      NBUF=1
  238 CONTINUE
      IF(I.NE.J) GO TO 237
      CCN(I)=BNN
      CCS(I)=BSS
      IF(KODE.GT.0)GO TO 237
C     SET BCN AND BCS FOR INTERFACE ELEMENTS
      BCN(I)=BNN
      BCS(I)=BSS
      IF(BNN.LT.ZERO)TYPE "BNN",BNN
      IF(BSS.LT.ZERO)TYPE "BSS",BSS
  237 CONTINUE

C*****************************************************************************************************************************

C
C     ONE LINE OF SUCCESSIVE OVER RELAXION METHOD
C
      DO 380 I=1,NUMBD
      II=L(I)
C
C     SET RECURSIVE VALUES FOR I AND II
C
      THIK=THICK(II)
      GSTF=GEL(II)/THIK
      ESTF=EEL(II)/THIK
      CLOS=DN(I)
      RIDE=DS(I)
      PRIMN=PN(I)
      PRIMS=PS(I)
      MEDR=MLD(I)
      KODE=KOD(I)
      TMS=-CCS(I)*DS(I)
      TMN=-CCN(I)*DN(I)
      DIAGN=CCN(I)
      DIAGS=CCS(I)
```

```
      DMN=0.
      DMS=0
C
C     EVALUATE SUMS TMN, TMS, DMN, DMS   FOR ELEMENTS IN CURRENT SUBREGION
C
      DO 310 J=1,NUMBD
      IF(MED(J).NE.MEDR)GO TO 310
      TMN=TMN-BUF(NB)*DN(J)-BUF(NB+1)*DS(J)
      TMS=TMS-BUF(NB+2)*DN(J)-BUF(NB+3)*DS(J)
      NB=NB+4
      IF(KODE.GT.0)GO TO 312
      DMN=DMN-BUF(NB)*DN(J)-BUF(NB+1)*DS(J)
      DMS=DMS-BUF(NB+2)*DN(J)-BUF(NB+3)*DS(J)
      NB=NB+4
  312 CONTINUE
C
C     LOAD UP BUF FROM DISK FILE
C
      IF(NB.LT.LIM)GO TO 310
      CALL RDBLK(1,ISTAR,BUF,NBLOK,IER)
      IF(IER.NE.1)TYPE "RDBLK",IER
      NB=1
      ISTAR=ISTAR+NBLOK
  310 CONTINUE
C
C
*****************************************************************************************************************************

C
C     WORK OUT SOR ERROR AND CHECK FOR SLIP ON AN INTERFACE
C
      DIAGN=CCN(I)
      DIAGS=CCS(I)
C
C     INTERFACE ELEMENTS
C
      IF(KODE.NE.-1)GO TO 361
      ITF=IFACE(I)
      DISN=-DMN
      DISS=-DMS
      SIGMN=-TMN+BN(I)
      SIGMS=-TMS+BS(I)
      ONN(I)=SIGMN+PN(I)
      OSS(I)=SIGMS+PS(I)
C
C     SWITCH 10 FOR DEPENDENCE ON TOTAL STRESSES.
C
      IF(ISSW(10).NE.1)GO TO 369
      SIGMN=ONN(I)
      SIGMS=OSS(I)
  369 CONTINUE
C     BN(ITF)=-DISN
C     BS(ITF)=-DISS
C
C     SEAM ELEMENTS
C
      DN(ITF)=(-DISN-SIGMN*THIK/EEL(II))*0.5+0.5*BN(ITF)
      DS(ITF)=(-DISS-SIGMS*THIK/GEL(II))*0.5+0.5*BS(ITF)
      IF(N.LT.ISLIP)GO TO 361
```

```
      SIGSF=COHES(I)+ONN(I)*TANPHI(II)
      IF(ONN(I).LT.ZERO)SIGSF=COHES(I)
      IF(IYIELD(I).NE.0)GO TO 363
      IF(ABS(USS(I)).LT.SIGSF)GO TO 361
  363 USS=SIGSF
      IF(USS(I).LT.ZERO)USS=-USS
      IYIELD(I)=1
      IYIELD(ITF)=1
      US(ITF)=USS+PS(I)
  361 CONTINUE
      IF(ICODE.NE.-2)GO TO 362
      ITF=IFACE(I)
      DISN=-TMN+DN(I)
      DISS=-TMS+DS(I)
      SIGMN=-BMN
      SIGMS=-BMS
      ONN(I)=SIGMN+PN(I)
      OSS(I)=SIGMS+PS(I)
      DN(ITF)=ALPH*DN(ITF)+(1.-ALPH)*SIGMN
      DS(ITF)=ALPH*DS(ITF)+(1.-ALPH)*SIGMS
      IF(N.LT.ISLIP)GO TO 362
      IF(IYIELD(I).EQ.0)GO TO 362
      USS=COHES(I)+ONN(I)*TANPHI(II)
      IF(ONN(I).LT.ZERO)USS=COHES(I)
      IF(OSS(I).LT.ZERO)USS=-USS
      TMS=USS-PS(I)-SIGMS
C
C
      DIAGS=DCS(I)
  362 CONTINUE

*****************************************************************************************************************************

C
C     EVALUATE STRESSES AND DISPLACEMENT CONTRIBUTIONS FOR ELEMENTS IN CURRENT
C     SUBREGION
C
      DO 850 N=1,NUMOS
      NUMP=0
      MEDR=MEDB(N)
      WRITE(IRITE,62)N,MEDR
      WRITE(IRITE,53)
      CALL MOVSET(E,V,INF,ISYM,NSYM,MEDR)
      EVAL=E(MEDR)
      VVAL=V(MEDR)
      XD=(XEI(N)-XBI(N))/NUMI(N)
      YD=(YEI(N)-YBI(N))/NUMI(N)
      NNP=NUMI(N)
      DO 850 NI=1,NNP
      NUMP=NUMP+1
      XOL=XBI(N)+0.5*(2.*NI-1.)*XD
      YOL=YBI(N)+0.5*(2.*NI-1.)*YD
      SIGXX=AXX*YOL+BXX
      SIGYY=AYY*YOL+BYY
      SIGXY=AXY*YOL+BXY
      PXX=SIGXX
      PYY=SIGYY
      PXY=SIGXY
      UX=0.
      UY=0.
```

```
C
      DO 630 J=1,NUMBD
      JJ=L(J)
      MEDL=MED(J)
C     SKIP ELEMENTS NOT IN CURRENT SUBREGION
C
      IF(MEDE.NE.MEDL)GO TO 630
C
C
```

****

Author Diering J A C
Name of thesis Further developments of the boundary element method with applications in mining 1981

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.