

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
C     LODVEC NATLHES SUMSOL.
C
      SUBROUTINE LODVEC(VECT,U,T,KODE,IVAL)
      DIMENSION VECT(1),U(1),T(1),IBC(3)
      IF(KODE.LT.6)GO TO 18
      DO 17 I=1,3
      VECT(I+3)=T(I)
   17 VECT(I)=U(I)
      IVAL=IVAL+6
      RETURN
   18 CALL IBLOAD(KODE,IBC)
      DO 19 I=1,3
      IB=IBC(I)
      VECT(I)=U(I)
      IF(IB.EQ.2.OR.IB.EQ.4)VECT(I)=T(I)
   19 CONTINUE
      IVAL=IVAL+3
      RETURN
      END
C*************************************************************************************************************************
C
C     LOAD A VECTOR XOUT FROM LUMP ARRAY XIN.
C
      SUBROUTINE LSET(XOUT,XIN)
      DIMENSION XOUT(1),XIN(1)
      DO 1 I=1,3
    1 XOUT(I)=XIN(I)
      RETURN
      END
C*************************************************************************************************************************
      PARAMETER JQ=13,JU=69,JY=700,JY3=2100
C
C     EVALUATE LUMSUM AND LOAD VECT
C
      SUBROUTINE LUMSUM(SUM)
C     JACD                                                            *
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(3,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TRANS,TRAN2,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,P1,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,INITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E   ,IELT(JQ),KOD(JQ),LIND(JU),NUMEL,COH,PHI,THIK
C
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R   ,AREAR(JQ),IELTR(JQ),KOUR(JQ),LINDR(JU),NUMELR,COHR,PHIR,THIKR
C
      COMMON /BU/ BUF(1152),NUML(70)
C
      DIMENSION SUM(1),VECT(JY3)
```

```
      EQUIVALENCE (VECT,X)
      IF(JYS.NE.(JY*3))STOP ERROR IN DIMENSION OF JY (SEE T.DIERING)
      IVAL=1
      DO 20 NL=1,NLUMP
      LND=LINDR(NL)
      IF(LND.EQ.NIL)GO TO 20
      KODE=LICOD(NL)
      IF(KODE.LE.NIL)GO TO 20
      IF(LND.NE.1)GO TO 10
      IF(IBUF.GT.LIM)CALL DUMPO(IBUF,BUF,ISTAR,2)
      IF(KODE.LT.6)GO TO 2
      DO 1 I=1,3
      DO 1 J=1,3
      IBUF=IBUF+2
    1 SUM(I)=SUM(I)+BUF(IBUF)*TLUMP(J,NL)+BUF(IBUF-1)*ULUMP(J,NL)
      GO TO 20
    2 CALL IBLOAD(KODE,IBC)
      DO 503 I=1,3
      DO 503 J=1,3
      IBUF=IBUF+1
      IB=IBC(J)
      GO TO (11,12,11,12),IB
   11 SUM(I)=SUM(I)+BUF(IBUF)*ULUMP(J,NL)
      GO TO 503
   12 SUM(I)=SUM(I)+BUF(IBUF)*TLUMP(J,NL)
  503 CONTINUE
      GO TO 20
   10 IF(LND.EQ.3)GO TO 14
C
C     LOAD UP VECTOR
C
      CALL LODVEC(VECT(IVAL),ULUMP(1,NL),TLUMP(1,NL),KODE,IVAL)
      GO TO 20
   14 CALL LDBLK(NL,1,1)
      NUML(NL)=NUMEL
      DO 16 N=1,NUMEL
      KODE=KOD(N)
      CALL LODVEC(VECT(IVAL),U(1,N),T(1,N),KODE,IVAL)
   16 CONTINUE
   20 CONTINUE
      RETURN
      END

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

      PARAMETER JO=13,JU=89,JY=700
      SUBROUTINE MODBEN(MODUL)
C     JACD                                                              *
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),BLUMC(3,JU),LIKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIMI(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TRAN5,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,11,12,13,P1,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),INEAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,AIRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
C     SET ELASTIC MODULI AND RELATED CONSTANTS.   (BENCHMARKS)
C
```

```
      E=EMOD(MODUL)
      V=PR(MODUL)
      G=E/(TWO*(ONE+V))
      OMV=ONE-V
      OMTV=ONE-TWO*V
      F=4.
      FV=F*V
      FP=F*PI
      V1=ONE/(F*(ONE+V))
      VT=OMTV/OMV
      V2=ONE/(F*OMV)
      V3=3./OMTV
      V4=(3.-FV)/(F*OMV)
      CK=VT/(TWO*FP)
      CONS=ONE/(FP*G)
      V5=V3*V
      V6=(ONE-FV)/OMTV
      V7=V3*V
      V8=V3*5./3.
      V9=V7/3.
      CD=CK*2.*G
      RETURN
      END
*****************************************************************************************************************
      PARAMETER JQ=13,JU=89,JY=700
      SUBROUTINE MODSET(MODUL)
C     JACD                                                                                                  *
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TRANS,TRANZ,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATD,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
C     SET ELASTIC MODULI AND RELATED CONSTANTS.
C
      E=EMOD(MODUL)
      V=PR(MODUL)
      G=E/(TWO*(ONE+V))
      OMV=ONE-V
      OMTV=ONE-TWO*V
      F=4.
      FV=F*V
      FP=F*PI
      V1=ONE/(F*(ONE+V))
      VT=OMTV/OMV
      V2=ONE/(F*OMV)
      V3=3./OMTV
      V4=(3.-FV)/(F*OMV)
      CK=VT/(TWO*FP)
      CONS=ONE/(FP*G)
      RETURN
      END
*****************************************************************************************************************
```

```
C
C     GENERATE NODAL COORDINATES WHERE POSSIBLE
C
      SUBROUTINE NODGEN(X,NODE,ND,IREAD,IRITE)
      DIMENSION X(3,ND)
      NIL=0
      HALF=0.5
      READ(IREAD) NSTAR,NSTOP
C
C     MODE = 1 FOR NODAL POINTS
C     MODE = 2 FOR MIDPOINT NODES
C     MODE = 3 FOR CENTROIDAL POINTS
C     MODE = 4 FOR LINES OF NODES
C     MODE = 5 FOR PLANES OF NODES (NOT YET OPERATIONAL)
C
      TYPE "GROUPS OF NODAL COORDINATES ARE NOW READ IN "
    5 READ(IREAD)NUM,MODE
      IF(NUM.LE.NIL.OR.MODE.LE.0.OR.MODE.GT.5)GO TO 60
      TYPE "WILL READ ",NUM," LINES OF MODE",MODE," DATA"
      GO TO (10,20,30,40,50),MODE
   10 CONTINUE
      DO 1 N=1,NUM
      READ(IREAD)NN,(X(J,NN),J=1,3)
      IF(NN.GT.NODE)NODE=NN
    1 TYPE "AT NODE",NN
      GO TO 5
   20 CONTINUE
      DO 11 N=1,NUM
      READ(IREAD)NN,N1,N2
      TYPE "H NODE",NN
      IF(NN.GT.NODE)NODE=NN
      DO 11 J=1,3
   11 X(J,NN)=HALF*(X(J,N1)+X(J,N2))
      GO TO 5
   40 CONTINUE
      DO 21 NLIN=1,NUM
      READ(IREAD)NB,NE
      IF(NE.GT.NODE)NODE=NE
      NSEG=NE-NB
      IF(NSEG.LE.NIL)STOP LINE NODE ERROR
      NSM=NSEG-1
      NN=NB
      DO 21 N=1,NSM
      NN=NN+1
      TYPE "GEN LINE NODE ",NN
      IF(NN.GT.NODE)NODE=NN
      DO 22 J=1,3
   22 X(J,NN)=X(J,NB)+(FLOAT(N)/FLOAT(NSEG))*(X(J,NE)-X(J,NB))
   21 CONTINUE
      GO TO 5
C
C     PUT IN ARRAY GEN HERE LATER
C
   50 CONTINUE
      STOP NO PLANE MODE AS YET
   30 CONTINUE
      DO 41 N=1,NUM
      READ(IREAD)NN,N1,N2,N3,N4
      IF(NN.GT.NODE)NODE=NN
```

```
      TYPE "C NODE",NN
      IF(N4.EQ.NIL)CALL XCENT(N1,N2,N3,X(1,NN),X,ND)
      IF(N4.NE.NIL)CALL XCENQ(N1,N2,N3,N4,X(1,NN),X,ND)
   41 CONTINUE
      GO TO 5
   60 CONTINUE
      IF(NSTAR.EQ.NIL.OR.NSTOP.EQ.NIL)RETURN
      WRITE(IRITE,101)
  101 FORMAT(//2X,4("NODE   X-COORD   Y-COORD   Z-COORD "),/)
      WRITE(IRITE,102)((N,(X(I,N),I=1,3)),N=NSTAR,NSTOP)
  102 FORMAT((1X,4(I5,3(1X,F8.2))))
      TYPE "EXIT NODGEN"
      RETURN
      END
*************************************************************************************************************************
C
C     INITIALIZE AN ARRAY OF DIMENSION N.
C
      SUBROUTINE NULL(X,N)
      DIMENSION X(1)
      DO 1 I=1,N
    1 X(I)=0.
      RETURN
      END
*************************************************************************************************************************
C
C     INITIALIZE TWO ARRAYS
C
      SUBROUTINE NULL2(X,Y,N)
      DIMENSION X(1),Y(1)
      DO 1 I=1,N
      X(I)=0.
    1 Y(I)=0.
      RETURN
      END
*************************************************************************************************************************
C
C     INITIALIZE 4 ARRAYS
C
      SUBROUTINE NULL4(A,B,C,D,N)
      DIMENSION A(1),B(1),C(1),D(1)
      DO 1 I=1,N
      A(I)=0
      B(I)=0.
      C(I)=0.
    1 D(I)=0
      RETURN
      END
*************************************************************************************************************************
C
C     INITIALIZE AN INTEGER ARRAY OF DIMENSION N.
C
```

```
      SUBROUTINE NULLI(IX,N,IMUL)
      DIMENSION IX(1)
      NN=N*IMUL
      DO 1 I=1,NN
    1 IX(I)=0
      RETURN
      END
*****************************************************************************************************************
      SUBROUTINE PRINK(Z,IRITE)
C
C     CALCULATE PRINCIPAL STRESSES
C
C
C
      DIMENSION COEF(4),U(3),Z(3,3),X(3),DV(3)
      DIMENSION DIP(3),DIPD(3),DIR(3,3)
      I1=1
      I2=2
      I3=3
      KODD=33
      DO 1 I=1,3
      DO 1 J=I,3
      Z(J,I)=Z(I,J)
    1 CONTINUE
C
C     CALCULATE CUBIC POLYNOMIAL
C
      COEF(1)=-1
      COEF(2)=0.
      DO 2 I=1,3
    2 COEF(2)=COEF(2)+Z(I,I)
      COEF(3)=-Z(1,1)*(Z(2,2)+Z(3,3))-Z(2,2)*Z(3,3)
      COEF(3)=COEF(3)+Z(2,1)**2+Z(2,3)**2+Z(1,3)**2
      COEF(4)=Z(1,1)*Z(2,2)*Z(3,3)+2.*Z(1,2)*Z(2,3)*Z(1,3)
      COEF(4)=COEF(4)-Z(1,1)*Z(2,3)**2-Z(2,2)*Z(1,3)**2-Z(3,3)*Z(1,2)**2
C
C     CALCULATE QUADRATIC POLYNOMIAL.
C
      DV(1)=3.*COEF(1)
      DV(2)=2.*COEF(2)
      DV(3)=COEF(3)
C
C     TAKE INITIAL GUESS.
C
      X(1)=2000
      X(2)=-2000
      DO 4 I=1,3
      IF(I.EQ.3)X(3)=COEF(2)-X(1)-X(2)
    6 CONTINUE
      DO 5 N=1,800
      AT=X(I)
      TOP=COEF(1)*AT**3+COEF(2)*AT*AT+COEF(3)*AT+COEF(4)
      BOT=DV(1)*AT*AT+DV(2)*AT+DV(3)
      IF(BOT.NE.0.) GO TO 7
      X(I)=X(I)*1.01
      WRITE(10,102)
  102 FORMAT(" DENOMM = 0")
      GO TO 6
```

```
      7 CONTINUE
        RAT=TOP/BOT
        X(I)=X(I)-RAT
        TOL=ABS(COEF(2))*0.00001
        IF(ABS(RAT).LT.TOL) GO TO 8
      5 CONTINUE
        WRITE(IRITE,161)I
    161 FORMAT(1X,I4" PRINCIPAL STRESS ERROR")
      8 CONTINUE
      4 CONTINUE
        SUM=0
        DO 9 I=1,3
        SUM=SUM+X(I)
      9 CONTINUE
        WRITE(10,105)SUM,COEF(2)
    105 FORMAT(" FINAL TRACE =",F12.4,"  INITIAL TRACE WAS",F12.4,/)
        DO 11 N=1,3
        A1=Z(1,1)-X(N)
        A2=Z(1,2)
        A4=Z(2,2)-X(N)
        A5=Z(1,3)
        A6=Z(2,3)
        A3=X(N)-Z(3,3)
        DET1=ABS(A1*A4-A2*A2)
        DET2=ABS(A1*A6-A2*A5)
        DET3=ABS(A2*A6-A4*A5)
        DET=AMAX1(DET1,DET2,DET3)
        IT=3
        IF(DET.EQ.DET1)IT=1
        IF(DET.EQ.DET2)IT=2
        IF(A2.EQ.0.)IT=2
        IF(A5.EQ.0.)IT=1
        IF(IT.NE.1) GO TO 51
C       A1*(-A6-A4*COEF(2))+A2*A2*COEF(2)=-A5*A2
        COEF(2)=(A1*A6-A5*A2)/(A2*A2-A1*A4)
        COEF(1)=(-A6-A4*COEF(2))/A2
     51 CONTINUE
        IF(IT.NE.2) GO TO 52
C       A1*(A3-A6*COEF(2))+A2*A5*COEF(2)=-A5*A5
        COEF(2)=(-A1*A3-A5*A5)/(A2*A5-A1*A6)
        COEF(1)=(A3-A6*COEF(2))/A5
     52 CONTINUE
        IF(IT.NE.3) GO TO 53
C       A2*(A3-A6*COEF(2))+A4*A5*COEF(2)=-A6*A5
        COEF(2)=(-A2*A3-A6*A5)/(A4*A5-A2*A6)
        COEF(1)=(A3-A6*COEF(2))/A5
     53 CONTINUE
        RR=SQRT(1.+COEF(1)*COEF(1)+COEF(2)*COEF(2))
        COEF(1)=COEF(1)/RR
        COEF(2)=COEF(2)/RR
        COEF(3)=-1./RR
        WRITE(10,134)(X(I),I=1,3),RR,IT,DET,TOL
    134 FORMAT(4F10.3,I4,2F10.4)
        DO 11 K=1,3
        DIR(N,K)=COEF(K)
     11 CONTINUE
        PI=3.14159
        DO 17 I=1,3
        DIPD(I)=ATAN2(DIR(I,2),DIR(I,1))*180./PI
        DIP(I)=ATAN(DIR(I,3)/SQRT(1.-DIR(I,3)**2))*180./PI
```

```
   17 CONTINUE
      WRITE BINARY (3) X,DIP,DIPD
      WRITE(IRITE,201)I1,X(2),DIP(2),DIPD(2),I2,X(3),DIP(3),DIPD(3)
     1  ,I3,X(1),DIP(1),DIPD(1)
C     WRITE(IRITE,201)(X(I),DIP(I),DIPD(1),I =1,3)
C 201 FORMAT(/" MAX TENSILE  ="F10.3"   DIP ="F6.1"  DIP DIR ="F6.1,/
C    1       " MIN TENSILE  ="F10.3"   DIP ="F6.1"  DIP DIR ="F6.1,/
C    2       " INTERMEDIATE ="F10.3"   DIP ="F6.1"  DIP DIR ="F6.1,/)
  201 FORMAT(1X3("SIG-"I1" :"F7.2"  DIP ="F5.0"  DIR ="F5.0,3X))
      RETURN
      END

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

C
C     OPEN THE MAJOR BEM STORAGE FILES
C
      SUBROUTINE SETFILE
      CALL OPEN(1,"BEMDATA",1,IER)
      IF(IER.NE.1)TYPE "BEMDATA OPEN ER",IER
      CALL OPEN(2,"LUMSTOR.BE",3,IER)
      IF(IER.NE.1)TYPE "LUMSTOR OPEN ER",IER
      CALL OPEN(3,"BVECT.BE",3,IER)
      IF(IER.NE.1)TYPE "BVECT OP ER",IER
      CALL OPEN(4,"DIAGEL.BE",3,IER)
      IF(IER.NE.1)TYPE "DIAGEL OP ER",IER
      CALL OPEN(5,"COSINE.BE",3,IER)
      IF(IER.NE.1)TYPE "COSINE OPEN ERROR",IER
      RETURN
      END

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

C     JACD                                                              *
      PARAMETER JQ=13,JU=89,JY=700,JY3=2100
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TRAN5,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,P1,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E  ,IELT(JQ),KOD(JQ),LIND(JU),NUMEL,COH,PHI,THIK
C
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R  ,AREAR(JQ),IELTR(JQ),KOUR(JQ),LINDR(JU),NUMELR,COHR,PHIR,THIKR
C
      COMMON /BU/ BUF(1152),NUML(JU)
C
C     SOLIN ---- SOLUTION OF MAIN EQUATIONS
C
      DIMENSION TDIAG(3),DIS(3),TRAC(3),RES(3),PRS(3),DCO(3,3)
      DIMENSION SUM(3),SUML(3),UDIAG(3),W(3),VECT(JY3)
      EQUIVALENCE (VECT,X)
C
      ISWAP=30
```

```
      CALL KRONO(2,IRITE,TON,TINC)
      CALL SETFILE
      CALL OPEN(6,"XSTOR.BE",3,IER)
      IF(IER.NE.1)TYPE "XSTOR OPEN ER",IER
      WRITE BINARY (6) X
      REWIND 6
C
      TENS=ZERO
      IF(ISSW(14).EQ.1)TENS=9999.
      NIL=0
      DO 100 NUM=1,NITER
      NIL=0
      IF(NUM.EQ.2)DEL=DEL*AMAX
      AMAX=ZERO
      IBUF=NIL
      ISTAR=ICHAN
      JOUT=NIL
      IF(NUM.EQ.1)JOUT=1
      CALL DUMPO(IBUF,EUF,ISTAR,2)
C
C     B.E.  SECTION
C
      DO 511 MOD1=1,NMOD
      DO 511 ML=1,NLUMP
C     TEMPORARY
C     TYPE "ML",ML
      IF(LMOD(ML).NE.MOD1)GO TO 511
      KODL=LKOD(ML)
      IF(KODL.LE.NIL)GO TO 511
      CALL LDBLK(ML,1,2)
      CALL NULL(SUML,3)
C
C     LUMP=LUMP
C
      CALL LUMSUM(SUML)
C     TYPE ML,SUML
C
C     CALCULATE DETAILED SUMS.
C
      DO 1 M=1,NUMELR
      READ BINARY (4) UD1AG
      READ BINARY (3) SUM
      DO 33 I=1,3
   33 SUM(I)=SUML(I)+SUM(I)
      CALL SUMELT(SUM,ML)
C     TYPE "ELT",M,SUM
      IFF=IFACE(ML)
      IF(IFF.NE.NIL)CALL LDBLK(IFF,1,1)
      KODER=KODR(M)
      CALL IBLOAD(KODER,IBC)
      DO 14 I=1,3
      IBB=IBC(I)
      IF(IBB.NE.1)GO TO 504
      DIFF=OMEGA*SUM(1)/HALF
      UR(I,M)=UR(I,M)+DIFF
      IF(KODER.NE.8)GO TO 505
      U(I,M)=UR(I,M)
      GO TO 505
  504 IF(IBB.NE.2)GO TO 505
      DIFF=OMEGA*SUM(1)/UD1AG(I)
```

```
      TR(I,M)=TR(I,M)-DIFF
      IF(KODER.NE.7)GO TO 505
      T(I,M)=-TR(I,M)
  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.10)WRITE(10,20) UR(I,M),TR(I,M),ABSUM,I,M
   20 FORMAT(1X,3(F16.5,1X),2I4)
   10 CONTINUE
   14 CONTINUE
      IF(NUM.LT.NSLIP)GO TO 1
      IF(KODER.LT.7)GO TO 1
      READ BINARY (8) DCO,W
C
C     KODE  7 = INTACT FAULT (TRACTIONS     SPECIFIED)
C     KODE  8 = INTACT FAULT (DISPLACEMENTS SPECIFIED)
C     KODE  9 = MINED WITH COMPLETE CLOSURE RESTRICTION.
C     KODE 11 = SHEAR FAILED INTERFACE.
C     KODE 10 = TENSILE FAILED INTERFACE.
C
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*W(J)
      TRAC(I)=TRAC(I)+DCI*TR(J,M)
   53 DIS(I)=DIS(I)+DCI*(UR(J,M)-U(J,M))
C
C     CHANGE IBC 3 AND 4 UNKNOWNS.
C
      IBB=IBC(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(KODER.EQ.9)GO TO 42
      IF(KODER.EQ.10)GO TO 47
      SHSTR=COH-PHI*RES(3)
      SHEAR=SQRT(RES(1)**2+RES(2)**2)
```

```
C
      IF(KODER.EQ.11)GO TO 43
C     IF(SHEAR.GT.SHSTR)GO TO 43
      GO TO 45
   43 IF((RES(3)).GT.TENS)GO TO 44
C  C
C     SHEAR FAILURE
C
      DIS(3)=ZERO
      TRAC(1)=(RES(1)*SHSTR/SHEAR)+PRS(1)
      TRAC(2)=(RES(2)*SHSTR/SHEAR)+PRS(2)
C     KODER=11
C     TYPE "S",M
      GO TO 45
C  C
C     TENSILE FAILURE
C
   44 CONTINUE
      KODER=10
      CALL LSCT(TRAC,PRS)
C  C  TYPE "T",M
      GO TO 45
C
C  C  TOTAL CLOSURE ELEMENTS
C
   42 CONTINUE
C     IF(DIS(3).LT.THICK)GO TO 45
      DIS(3)=THICK
      GO TO 45
   47 CONTINUE
C
C     CHECK FOR CLOSURE OF AN OPEN FAULT ELEMENT.
C  C
      IF(DIS(3).LT.ZERO)GO TO 45
C     KODER=11???
C  C  TYPE "CLOS-F",M
      DIS(3)=ZERO
      IF(RES(3).GT.ZERO)GO TO 45
C     TRAC(3)=PRS(3)
   45 CONTINUE
C
C  C  BACK TRANSFORM TRAC AND DIS
C
      DO 50 J=1,3
C     TR(M,J)=ZERO
      DNEW=ZERO
      DO 51 I=1,3
C     DNEW=DNEW+DCO(I,J)*DIS(I)
   51 TR(M,J)=TR(M,J)+DCO(I,J)*TRAC(I)
      T(J,IFF)=-TR(M,J)
C     UR(M,J)=U(J,IFF)+DNEW
   50 CONTINUE
      N4=INDEX(M,4)
C     IF(N4.NE.NIL)KODER=-KODER
      KODR(M)=KODER
      KOD(IFF)=KODER
C  1 CONTINUE
      CALL TUL(ULUMP(1,ML),TLUMP(1,ML),ALUMP(ML))
      CALL LDBLK(ML,2,2)
C     IF(IFF.NE.NIL)CALL LDBLK(IFF,2,1)
      IBM=IBUF-2
```

```
      IF(NUM EQ 1)WRITE(IRITE,666)ISTAR,(BUF(I),I=IBM,IBUF),(LINDR(I)
     1   ,I=1,NLUMP)
  666 FORMAT(1X,I5,2X,3F8.5,2X,9511)
  511 CONTINUE
      REWIND 2
      REWIND 3
      REWIND 4
      REWIND 5
C     TYPE "DEL = ",DEL
      IF(NUM EQ 1)WRITE(IRITE,449)ISTAR
      WRITE(IRITE,448)NUM,JOUT,AMAX
      IF(JOUT EQ NIL)GO TO 101
  448 FORMAT(1X,I4" ITS",I8" ELTS OUT","  MAX ER =",F10.5)
  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)
      READ BINARY (6) X
      CALL KRUNO(5,IRITE,TON,TING)
      ISWAP=1
      CALL BACK
      STOP
      END
**********************************************************************************************************************
      PARAMETER JU=13,JU=89,JY=700,JY3=2100
      SUBROUTINE SUMELT(SUM,ML)
C     JACD                                                          *
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  ND1,  8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3  EM       PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TR     (RAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,ND1MEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E   ,IELT(JQ),KOD(JQ),LIND(JU),NUMEL,COH,PHI,THIK
C
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R   ,AREAR(JQ),IELTR(JQ),KODR(JQ),LINDR(JU),NUMELR,COHR,PHIR,THIKR
C
      COMMON /BU/ BUF(1152),NUML(JU)
C
C     EVALUATE INFLUENCE OF NEARBY ELTS (FOR SOLUTION)
C
      DIMENSION SUM(1),DIS(1),TRAC(1),VECT(JY3)
      EQUIVALENCE (VECT,X)
C
      IVAL=1
      DO 3 NL=1,NLUMP
      KODE=LKOD(NL)
      IF(KODE.LE.NIL)GO TO 3
      LND=LINDR(NL)
```

```
      IF(LND.LE.1)GO TO 3
      IF(LND.EQ.3)GO TO 4
      IF(IBUF.GT.LIM)CALL DUMPO(IBUF,BUF,ISTAR,2)
      CALL SUMSOL(SUM,VECT(IVAL),BUF,KODE,IBUF,IVAL)
      GO TO 3
    4 NUMEL=NUML(NL)
      DO 6 N=1,NUMEL
      IF(IBUF.GT.LIM)CALL DUMPO(IBUF,BUF,ISTAR,2)
      IF(ML.NE.NL)GO TO 10
      ISTOR=IVAL
      CALL LODVEC(VECT(ISTOR),UR(1,N),TR(1,N),KODE,ISTOR)
   10 CALL SUMSOL(SUM,VECT(IVAL),BUF,KODE,IBUF,IVAL)
    6 CONTINUE
    3 CONTINUE
      RETURN
      END
C*************************************************************************************************************************
C     SUMSOL MATCHES LODVEC
C
      SUBROUTINE SUMSOL(SUM,VECT,BUF,KODE,IBUF,IVAL)
      DIMENSION SUM(1),VECT(1),BUF(1)
      IF(KODE.GT.5)GO TO 5
      DO 1 I=1,3
      DO 1 J=1,3
      IBUF=IBUF+1
    1 SUM(J)=SUM(J)+BUF(IBUF)*VECT(J)
      IVAL=IVAL+3
      RETURN
    5 DO 2 I=1,3
      DO 2 J=1,3
      IBUF=IBUF+2
    2 SUM(I)=SUM(I)+BUF(IBUF-1)*VECT(J)+BUF(IBUF)*VECT(J+3)
      IVAL=IVAL+6
      RETURN
      END
C*************************************************************************************************************************
      PARAMETER JU=13,JU=89,JY=700,JY3=2100
C
C     SUMUP IS EQUIVALENT TO LOAD
C
      SUBROUTINE SUMUP(SUM,DIS,TRAC,KODE)
      JACD
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIM0(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CU,CONS,IDC(3),
     4  TRAN5,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /BUF/ BUF(1152)
C
      DIMENSION SUM(1),TRAC(1),DIS(1)
      IF(IBUF.GT.LIM)CALL DUMPO(IBUF,BUF,ISTAR,2)
```

```
      IF(NCODE.LE.6)GO TO 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
    2 DO 503 I=1,3
      DO 503 J=1,3
      IBUF=IBUF+1
      IB=IBC(J)
      GO TO (11,12,11,12),IB
   11 SUM(I)=SUM(I)+BUF(IBUF)*DIS(J)
      GO TO 503
   12 SUM(I)=SUM(I)+BUF(IBUF)*TRAC(J)
  503 CONTINUE
      RETURN
      END
```

```
C
C     SUBROUTINE TO FIX SYMMETRY OF A VECTOR
C
      SUBROUTINE SYMFIX(XSYM,YSYM,ZSYM,NMULT,VECT)
      DIMENSION NMULT(1),VECT(1)
      IF(NMULT(1).NE.1)VECT(1)=XSYM-VECT(1)
      IF(NMULT(2).NE.1)VECT(2)=YSYM-VECT(2)
      IF(NMULT(3).NE.1)VECT(3)=ZSYM-VECT(3)
      RETURN
      END
```

```
C
C     TRANSFORM A TENSOR FROM GLOBAL TO LOCAL COORDS.
C
      SUBROUTINE TGTOL(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
```

```
      PARAMETER JQ=13,JU=89,JY=700,JY3=2100
      SUBROUTINE TTUU(XED,XRD,WCK,WCR,DCO)
C     JACD                                                *
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
```

```
     4  TRAN5,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,P1,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      DIMENSION XED(3),RCOM(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
      II=4*I-3
      RMULI=NMULT(I,ISYM)
      RCI=RCOM(I)
      RIV3=RCI*V3
      RIV2=RCI*V2
      DCI=DCO(I)
      TT(II)=TT(II)+DRDN*(1.+RIV3*RCI)*RMULI
      UU(II)=UU(II)+WCR*(V4+RIV2*RCI)*RMULI
      IF(I.EQ.3)RETURN
      IPI=I+1
      DO 2 J=IPI,3
      JI=3*I-3+J
      IJ=3*J-3+I
      RCJ=RCOM(J)
      DCJ=DCO(J)
      RMULJ=NMULT(J,ISYM)
      TF=(DCI*RCJ-DCJ*RCI)*WCK
      TFUNC=DRDN*RIV3*RCJ
      TT(JI)=TT(JI)+(TFUNC+TF)*RMULJ
      TT(IJ)=TT(IJ)+(TFUNC-TF)*RMULI
      FAC=WCR*RIV2*RCJ
      UU(JI)=UU(JI)+FAC*RMULJ
    2 UU(IJ)=UU(IJ)+FAC*RMULI
      RETURN
      END
```

****

```
      PARAMETER JQ=13,JU=89,JY=700,JY3=2100
      SUBROUTINE TUL(ULUMP,TLUMP,ALUMP)
C
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R  ,AREAR(JQ),IELTR(JQ),KODR(JQ),LINDR(JU),NUMELR,COHR,PHIR,THIKR
C
      DIMENSION TLUMP(1),ULUMP(1)
C
C     CALCULATE AND STORE LUMP DISPLACEMENTS AND DISPLACEMENTS.
C
      DO 10 I=1,3
      ULUMP(I)=0.
   10 TLUMP(I)=0.
      DO 1 N=1,NUMELR
      ARE=AREAR(N)/ALUMP
      DO 1 I=1,3
```

```
      ULUMP(I)=ULUMP(I)+UR(I,N)*ARE
    1 TLUMP(I)=TLUMP(I)+TR(I,N)*ARE
      RETURN
      END
*****************************************************************************************************************
      PARAMETER JQ=13,JU=89,JY=700,JY3=2100
      FUNCTION TVAL(D1,R1)
C     JACD                                                                  *
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TRAN5,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,RR,R2,R3,R4,K1,K2,K3
C
      COMMON /E/ E1(3),E2(3),I,J
C
C     EVALUATE T COEFFICIENTS
C
      TVAL=ZERO
      DO 1 K=1,3
      AL=ALT(I,J,K)
      IF(AL.EQ.ZERO)GO TO 1
      TVAL=TVAL+AL*E1(K)*(ALOG(D1+R1))
    1 CONTINUE
      TVAL=-TVAL*CK
      RETURN
      END
*****************************************************************************************************************
      PARAMETER JQ=13,JU=89,JY=700,JY3=2100
      SUBROUTINE UDI
C     JACD                                                                  *
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TRAN5,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E  ,IELT(JQ),KOD(JQ),LIND(JU),NUMEL,COH,PHI,THIK
C
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R  ,AREAR(JQ),IELTR(JQ),KODR(JQ),LINDR(JU),NUMELR,COHR,PHIR,THIKR
C
      COMMON /E/ E1(3),E2(3),I,J
C
C     CALCULATE U-COEFFS DOWN THE MAIN DIAGONAL (FOR SOLIN)
C
      DIMENSION DIAG(3),DC(3,3),W(3)
```

```
      CALL SETFILE
      TYPE "INFO UD1"
      DO 5 MOD1=1,NMOD
      CALL MODSET(MOD1)
      DO 4 NL=1,NLIMP
      IF(LMOD(NL).NE.MOD1)GO TO 4
      IF(LKOD(NL).LE.NIL)GO TO 4
      CALL LOBLK(ML,1,1)
      DO 4 M=1,NUMEL
      TYPE "UD1 ELT",M,NL
      KODER=KOD(M)
      CALL LOONOD(N1,N2,N3,N4,INDEX(1,M))
      CALL DCOSP(N1,N2,N3,A,B,C,X,NU)
      IF(KODER.GE.7)CALL DCOSF(A,B,C,DC)
      CALL LSET(XCOM,XCEL(1,M))
      DEPTH=XCOM(3)
C
C     CALCULULATE PRIM TRACTIONS ON ELT M
C
      DO 3 I=1,3
      DIAG(I)=ZERO
      W(I)=ZERO
      DO 52 J=1,3
   52 W(I)=W(I)+(DEPTH*PRIM1(I,J)+PRIM0(I,J))*DC(3,J)
    3 CONTINUE
      NR=3
      IF(N4.NE.NIL)NR=4
      DO 1 NROT=1,NR
      L1=N1
      L2=N2
      IF(NROT.EQ.1)GO TO 30
      IF(N1.EQ.4)GO TO 20
      IF(NROT.EQ.2)L1=N3
      IF(NROT.EQ.2)L2=N1
      IF(NROT.EQ.3)L1=N2
      IF(NROT.EQ.3)L2=N3
      GO TO 30
   20 CONTINUE
      IF(NROT.NE.2)GO TO 31
      L1=N4
      L2=N1
      GO TO 30
   31 IF(NROT.NE.3)GO TO 32
      L1=N3
      L2=N4
      GO TO 30
   32 L1=N2
      L2=N3
   30 CONTINUE
      CALL DCOSL(L2,L1,A1,B1,C1,X,NU)
      E1(1)=A1
      E1(2)=B1
      E1(3)=C1
      E2(1)=B*E1(3)-C*E1(2)
      E2(2)=C*E1(1)-A*E1(3)
      E2(3)=A*E1(2)-B*E1(1)
C
C     CALCULATE DISTANCES
C
      D=ZERO
```

```
      DI=ZERO
      DJ=ZERO
      DO 2 L=1,3
      D =D +E2(L)*(X(L,L1)-XCOM(L))
      DI=DI+E1(L)*(X(L,L1)-XCOM(L))
      DJ=DJ+E1(L)*(X(L,L2)-XCOM(L))
    2 CONTINUE
C
      RI=SQRT(DIST2(XCOM,X(1,L1)))
      RJ=SQRT(DIST2(XCOM,X(1,L2)))
      DO 1 I=1,3
      J=1
      DIAG(I)=DIAG(I)-UVAL(DI,RI,D)+UVAL(DJ,RJ,D)
    1 CONTINUE
      WRITE BINARY (4) DIAG
      IF(KODER.GE.7)WRITE BINARY (5) DC,W
    4 CONTINUE
    5 CONTINUE
      CALL RESET
      TYPE "EXIT UDI"
      RETURN
      END
*****************************************************************************************************************
      PARAMETER JQ=13,JU=89,JY=700,JY3=2100
      FUNCTION UVAL(D1,U1,D)
C     JACD                                                                  *
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TRANS,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATD,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /E/ E1(3),E2(3),I,J
C
C     EVALUATE U COEFFICIENTS
C
      T1=(D1/D)
      T2=U1/D
      T3=D1/U1
      UVAL=ZERO
      IF(I.EQ.J)UVAL=V4*ALOG(T1+T2)
      UVAL=UVAL+V2*(E1(I)*E1(J)*(ALOG(T1+T2)-T3)
     1          +E2(I)*E2(J)*T3-(E1(I)*E2(J)+E2(I)*E1(J))/T2)
      UVAL=-UVAL*D*CONS
      RETURN
      END
*****************************************************************************************************************
C
C     SUBROUTINE TO FIND CENTROID OF AN ELEMENT
C
      SUBROUTINE XCEN(INDEX,X,XC,ND)
      DIMENSION INDEX(4),X(3,ND),XC(3)
```

```
      CALL LOONOD(NOD1,NOD2,NOD3,NOD4,INDEX)
      IF(NOD4.EQ.0)CALL XCENT(NOD1,NOD2,NOD3,XC,X,ND)
      IF(NOD4.NE.0)CALL XCENQ(NOD1,NOD2,NOD3,NOD4,XC,X,ND)
      RETURN
      END
*****************************************************************************************************************************
C
C     SUBROUTINE TO ESTIMATE CENTROID OF A NICE QUADRILATERAL.
C
      SUBROUTINE XCENQ(I,J,K,L,XNUM,X,ND)
      DIMENSION XNUM(3),X(3,ND)
      DO 1 N=1,3
    1 XNUM(N)=(X(N,I)+X(N,J)+X(N,K)+X(N,L))/4.
      RETURN
      END
*****************************************************************************************************************************
C
C     SUBROUTINE TO CALCULATE CENTROID OF A TRIANGLE.
C
      SUBROUTINE XCENT(I,J,K,XNUM,X,ND)
      DIMENSION XNUM(3),X(3,ND)
      DO 1 N=1,3
    1 XNUM(N)=(X(N,I)+X(N,J)+X(N,K))/3.
      RETURN
      END
*****************************************************************************************************************************
```

A P P E N D I X 3

PARTIAL LISTING OF PROGRAMME MBEM

```
*****************************************************************************************************************
C
C     ------------------------------------------------------------------------
C     PARTIAL SOURCE LISTING FOR PROGRAM MDEM
C
C     PROGRAM WRITTEN BY J. A. C. DIERING     1980/81
C
C     COPYRIGHT RESIDES WITH STEFFEN ROBERTSON AND KIRSTEN INC.
C
C     SUBPROGRAMS AND SUBROUTINES ARE GIVEN IN ALPHABETIC ORDER
C     EACH IS SEPARATED BY A LINE OF STARS (*****)
C
C     ------------------------------------------------------------------------

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

HEADR
MDEMLIST
ARCUT
LCM
DLM1
DLM4
DLM5
DLM6
DLMINF
DLOSA
DDINF
DDOUTQ
DDREAD
DDSOL
DDTT
GAUSS
GEOMET
INITL
INITT
INPUT
LCOOR
LESET
LGEOM
LOAD
LOAD1
LSET
LUMPR
MULL
PRIMDD
RIDEL
SIGBEN
SIGDD
ISW
SOLIN
SSDD
SUMDD
SUMELT
SUMUP
SYMFIX
TGTOL
TTUU
TUO
```

```
TOL
IVAL
VGTOL
VLTOG

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

C
C     OUTPUT A SQUARE ARRAY OF DD ELEMENTS.
C
      SUBROUTINE AROUT(X,NN,MM,IRITE)
      DIMENSION X(25,25)
      WRITE(IRITE,104)
  104 FORMAT(/1X,120("-"),/)
      KK=MM
      IF(KK.GT.12)KK=12
      KK1=KK+1
      WRITE(IRITE,101)
  101 FORMAT(" LEFT HALF OF LAYOUT",/)
      DO 1 I=1,NN
    1 WRITE(IRITE,100)(I,(X(I,J),J=1,KK))
      IF(KK.EQ.MM)RETURN
      WRITE(IRITE,102)
  102 FORMAT(" RIGHT HALF OF LAYOUT",/)
      DO 2 I=1,NN
    2 WRITE(IRITE,100)(I,(X(I,J),J=KK1,MM))
  100 FORMAT(I4,13(F9.4))
      RETURN
      END

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

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  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,WCHT(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,DLL,PI,A,B,C,R,RCO,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     PROGRAM BEM
C
      CALL JOBTIME("MBEM    ",6)
      CALL INITT
      CALL INITL
      CALL SWAP("BEM1.SV",IER)
      IF(IER.NE.1)TYPE " BEM1 SWAP ER",IER
      IF(ISSW(1).NE.1)CALL SWAP("BEMINF.SV",IER)
      IF(IER.NE.1)TYPE "BEMINF ER",IER
      IF(ISSW(1).NE.1)CALL SWAP("DDINF.SV",IER)
      IF(IER.NE.1)TYPE "DDINF ER",IER
      IF(ISSW(2).NE.1)CALL UDI
```

```
      CALL SWAP("SOLIN.SV",IER)
      IF(IER.NE.1)TYPE "SOLIN ER",IER
      IF(ISSW(6).NE.1)CALL SWAP("BEM4.SV",IER)
      IF(IER.NE.1)TYPE " BEM4 SWAP ER",IER
      IF(ISSW(6).NE.1)CALL UBOUTO(DX,DY,DZ,OZZT,NN,MM,IRITE)
      IF(ISSW(9).NE.1)CALL SWAP("BEM5.SV",IER)
      IF(IER.NE.1)TYPE "BEM5 SWAP ER",IER
      IF(NUMOR.LT.1)GO TO 1
      IF(ISSW(2).EQ.1)CALL LUMPR
      CALL SWAP("BEM6.SV",IER)
      IF(IER.NE.1)TYPE "BEM6 ER",IER
    1 CONTINUE
      CALL ALARM
      CALL JOBTIME("MBEM   ",6)
      STOP
      END
C*****************************************************************************************************************************
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),ZLANG(9,9),XLANG(9,9),YLANG(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),UC(3),ISUB(3),XNUM(3),RCOM(3),SYM(3),FALL(3),OR1G(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,NL,MODDD,NS,
     *  ZERO,ONE,IONE,11,12,13,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,LFAR,THIN,ESEAM,QSEAM,COH,PHI,
     2  IDFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,INF(1152),IREAD,IRITE,NLLEL,NRELO,EDUL,ISSW(15),ISW(15)
C
C     BEM1
C
C
      CALL INPUT
      CALL DBREAD
      CALL LOLOM
      CALL BACK
      STOP
      END
C*****************************************************************************************************************************
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),ZLANG(9,9),XLANG(9,9),YLANG(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),UC(3),ISUB(3),XNUM(3),RCOM(3),SYM(3),FALL(3),OR1G(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,NL,MODDD,NS,
     *  ZERO,ONE,IONE,11,12,13,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, CTAN, THIK, ESEAM, GSEAM, COH, PHI,
     2   IDFSZ, ICHAN, LIN, IDISK, IBUF, ISTAR, BUF(1152), IREAD, IWRITE, NELEL, NRECO, KODE, ISSW(15), ISW(15)
C
C     SUBPROGRAM TO OUTPUT THE RESULTS.
C
      DIMENSION XE(3)
      IF(ISSW(6).EQ.1)GO TO 2
      WRITE(IWRITE,101)
      DO 1 N=1,NELT
      CALL XCEN(INDEX,N,X,XE)
    1 WRITE(IWRITE,100)N,(XE(I),I=1,3),(U(N,I),I=1,3),(T(N,I),I=1,3),
     1KOD(N),AREA(N),IFACE(N),(IBC(N,K),K=1,3)
  100 FORMAT(1X,I3,1X,F8.2,2(F9.2,1X),3(F9.4,1X),3(F9.5,1X),I6,4X,F9.2,I6,
     *3X,3I2,1X)
  101 FORMAT(/5"X-COORD"T17"Y-COORD"T27"Z-COORD"T40"U-X"T48"U-Y"T60
     1"U-Z"T70"T-X"T80"T-Y"T90"T-Z"T99"KODE"T109"AREA"T116"IFACE   IBC",
     *//)
      CALL TTT(IWRITE)
    2 CONTINUE
      CALL BACK
      STOP
      END
```

```
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  NINE(25,25),DZ(25,25),DX(25,25),DY(25,25),OZZT(25,25),
     4  AK(9,9),BK(9,9),DK(9,9),DMA(9,9),ZLANG(9,9),XLANG(9,9),YLANG(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),NDIM(8,8),NMULT(8,3),
     8  TRAN42,TRAN21,TRAN7,TRAN5,TRANLL,TRANL,TRANDL,TRANID,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,MM,NLSX,NLSY,NDIML,NDIMH,NDIM,ISMAL,JSMAL,NUMOR,NOLIP,NL,NODNO,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,CTAN,THIK,ESEAM,GSEAM,COH,PHI,
     2  IDFSZ,ICHAN,LIN,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IWRITE,NELEL,NRECO,KODE,ISSW(15),ISW(15)
C
C     SUBPROGRAM TO OUTPUT THE RESULTS.
C
      DIMENSION XE(3)
      IF(ISSW(9).EQ.1)GO TO 2
      WRITE(IRITE,102)
  102 FORMAT(" RESULTS BELOW ARE IN GLOBAL COORDINATE SYSTEM",//)
      WRITE(IRITE,101)
      DO 1 N=1,NELT
      CALL XCEN(INDEX,N,X,XE)
      DO 3 I=1,3
      E2(I)=U(N,I)
      E1(I)=T(N,I)
    3 CONTINUE
      CALL VLTOG(XE,ALPHA,ORIG)
      CALL VLTOG(E1,ALPHA,FALL)
      CALL VLTOG(E2,ALPHA,FALL)
    1 WRITE(IRITE,100)N,XE,E2,E1,KOD(N),AREA(N),IFACE(N),(IBC(N,K),K=1,3)
  100 FORMAT(1X,I3,1X,F8.2,2(F9.2,1X),3(F9.4,1X),3(F9.5,1X),I6,4X,F9.2,I6,
     *3X,3I2,1X)
```

```
  101 FORMAT(T5"X-COORD"T17"Y-COORD"T27"Z-COORD"T40"U-X"T50"U-Y"T60
     1"U-Z"T70"T-X"T80"T-Y"T90"T-Z"T99"KODE"T109"AREA"T116"IFACE   IBC",
     *//)
      CALL TTT(IRITE)
    2 CONTINUE
      CALL BACK
      STOP
      END
*****************************************************************************************************************
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),LMUD(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),IC(3),ISUB(3),XNUN(3),RCOM(3),SYM(3),FALL(3),ORIG(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIR(6,6),NMULT(6,3),
     8  TRAN42,TRAN21,TRAN7,TRANS,TRANLL,TRANL,TRANDL,TRANDD,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,MM,NLSX,NLSY,NDIML,ND1MN,ND1M,ISMAL,JSMAL,NUMOR,NSLIP,NL,MODDD,NS,
     *  ZERO,ONE,ONE1,11,12,13,THIRD,HALF,TWO,NITER,OMEGA,DEL,PI,A,B,C,R,RSQ,EW,HL,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,NNECO,KODE,ISSW(15),ISW(15)
C
C     ..  ROGRAM FOR BENCHMARK STRESSES AND DISPLACEMENTS.
C
C
      DIMENSION XX(3),XEL(3),XC(3),DIS(3),TRAC(3),BCO(3),STRES(3,3),GX(64),GY(64),WGH(64)
      EQUIVALENCE (XNUM,XC),(STRES,TT)
      EQUIVALENCE (GX(1),BUF(1)),(GY(1),BUF(100)),(WGH(1),BUF(200))
      WRITE(IRITE,190)
C
C     CONSTANTS.
C
      TRAN42=TRAN42*2.
      TRANX=EW/3.
      TRAN21=TRAN21*1.5
C
      CALL OPEN(2,"BENCH ",IER)
      IF(IER.NE.1)TYPE "OPEN ER",IER
      DO 1 NK=1,NUMOR
      READ(2,102)XX,NODUL
  102 FORMAT(F11.3,2F12.3,I4)
      ABX=ABS(XX(3))
      IXP=XX(11)/EW+11
      IYP=XX(12)/EW+11
      IF(ABX.GT.(EW*2.))GO TO 70
      XX(11)=(IXP-HALF)*EW
      XX(12)=(IYP-HALF)*EW
      IF(ABX.GT.TRANX)GO TO 70
      WRITE(IRITE,106)NK
      XT=TRANX
      IF(XX(3).LT.ZERO)XT=-XT
      XX(13)=XT
   70 CONTINUE
      ZD=XX(13)
      CALL NODBEN(NODUL)
```

```
      NS=NSYM(MODUL)
      CALL NULL(W,FALL)
      CALL TUO(TT,UU)
      DO 3 NL=1,NLUMP
      ARL=ALUMP(NL)
      MOD2=LMOD(NL)
      IF(MOLZ.NE.MODUL)GO TO 3
      NEL=LKOD(NL)
      DO 111 ISYM=1,NS
C
C     LOAD SYMMETRY CONSTANTS.
C
      IDIR=NDIR(ISYM,NS)
      NIL=0
      IF(IDIR.EQ.NIL)GO TO 111
      CALL LESET(NL,XEL,DCO)
      RSQ=DIST2(XX,XEL)
      RATIO=RSQ/ARL
      IF(RATIO.LT.TRANL)GO TO 7
      TYPE "LUMP",NL
      R=SQRT(RSQ)
      CALL TUL(NL,TRAC,DIS)
      CALL SYMFIX(FALL,NMULT,TRAC,ISYM)
      CALL SYMFIX(FALL,NMULT,DIS,ISYM)
      CALL SIGDEN(TRAC,DIS,XX,XEL,ARL,DCO)
      GO TO 111
    7 CONTINUE
      DO 6 NE=1,NEL
      N=LUMPO(NL,NE)
      DO 52 L=1,3
      DIS(L)=U(N,L)
    2 TRAC(L)=T(N,L)
      NOD1=INDEX(N,1)
      NOD2=INDEX(N,2)
      NOD3=INDEX(N,3)
      NOD4=INDEX(N,4)
      CALL DCOSV(NOD1,NOD2,NOD3,DCO,X)
      AR=AREA(N)
      NQ=1
      IF(KOU(N).GT.NIL)GO TO 50
      NQ=2
      AR=AREAF(NOD1,NOD2,NOD3,X)
   50 CONTINUE
      CALL SYMFIX(FALL,NMULT,DCO,ISYM)
      CALL SYMFIX(FALL,NMULT,TRAC,ISYM)
      CALL SYMFIX(FALL,NMULT,DIS,ISYM)
      DO 11 NTQ=1,NQ
      IF(NTQ.EQ.1)GO TO 53
      AR=AREAF(NOD1,NOD3,NOD4,X)
      NOD2=NOD4
   53 CONTINUE
      NINTB=11
      NINTE=11
      W1=AR
      MINT=1
      NUMTRI=1
      IN=NOD1
      JN=NOD2
      KN=NOD3
C
```

```
      CALL XCENT(NOD1,NOD2,NOD3,XC,X)
      CALL SYMFIX(SYM,NMULT,XC,ISYM)
      RSQ=DIST2(XX,XC)
      RATIO=RSQ/AR
      IF(RATIO.GE.TRAN5)GO TO 20
      MINT=3
      NINTB=8
      NINTE=10
      IF(RATIO.GE.TRAN7)GO TO 20
      MINT=7
      NINTB=1
      NINTE=7
      IF(RATIO.GT.TRAN21)GO TO 20
      MINT=21
      KN=NODE+1
      AR=AR/3.
      NUMTRI=3
      CALL XCENT(NOD1,NOD2,NOD3,XEL,X)
      X(KN,1)=XEL(1)
      X(KN,2)=XEL(2)
      X(KN,3)=XEL(3)
      IF(RATIO.GT.TRAN42)GO TO 20
      MINT=42
      NUMTRI=6
      AR=AR/2.
      KN=NODE+1
      JN=NODE+2
   20 CONTINUE
      IF(MINT.GT.1)TYPE MINT,N,ISYM
      IF(MINT.GT.7)TYPE RATIO,NTQ
      DO 11 NTRI=1,NUMTRI
      IF(NUMTRI.NE.6)GO TO 43
      IF(NTRI.EQ.1.OR.NTRI.EQ.6)IN=NOD1
      IF(NTRI.EQ.2.OR.NTRI.EQ.3)IN=NOD2
      IF(NTRI.EQ.4.OR.NTRI.EQ.5)IN=NOD3
      IF(NTRI.GT.2)GO TO 41
      DO 441 L=1,3
  441 X(JN,L)=(X(NOD1,L)+X(NOD2,L))*HALF
      GO TO 21
   41 IF(NTRI.GT.4)GO TO 42
      DO 442 L=1,3
  442 X(JN,L)=(X(NOD2,L)+X(NOD3,L))*HALF
      GO TO 21
   42 DO 443 L=1,3
  443 X(JN,L)=(X(NOD3,L)+X(NOD1,L))*HALF
      GO TO 21
   43 CONTINUE
      IF(NTRI.EQ.1)GO TO 21
      IF(NTRI.EQ.2)IN=NOD3
      IF(NTRI.LT.3)GO TO 21
      IN=NOD1
      JN=NOD3
   21 CONTINUE
      DO 11 NUM=NINTB,NINTE
      IF(NINTB.EQ.11)GO TO 64
      W1=WGHT(NUM)*AR
      DO 26 L=1,3
   26 XNUM(L)=COORD(NUM,1)*X(IN,L)+COORD(NUM,2)*X(JN,L)+COORD(NUM,3)
     1*X(KN,L)
      CALL SYMFIX(SYM,NMULT,XNUM,ISYM)
```

```
C
      RSQ=DIST2(XNUM,XX)
C  64 R=SQRT(RSQ)
      CALL SIGBEN(TRAC,DIS,XX,XC,W1,DCO)
   11 CONTINUE
C   6 CONTINUE
  111 CONTINUE
    3 CONTINUE
C     RNUM=ISMAL*JSMAL
      H2=H*H
      AR=4.*H2
C     CALL GAUSS(GX,GY,WGH,H,AR)
      ARL=4.*HL*HL
      TRNDD=2.5*ARL
C     TRN49=.9*AR
      TRN9=3.61*AR
      TRN4=6.25*AR
C     TRNX=EW/3.5
C     WRITE(IRITE,985)KK
C985 FORMAT(/" BENCH MARK STRING NUMBER"I3" FOR STRESSES AND DISPLACEMENTS",/
C    1       " ***********************************************************",/)
C
C     SCALE X AND Y COORDS FROM BLOCK TO LOCAL COORDS.
C
C     XB=(I2*XB-I1)*H
C     YB=(I2*YB-I1)*H
C     XE=(I2*XE-I1)*H
C     YE=(I2*YE-I1)*H
  245 FORMAT(/" POINT NO.",I4," AT ROW=",I4," COL=",I4)
C 246 FORMAT("     ZD      UX",6X,"UY"6X"UZ"5X"MAX"5X"DIP    DIPD     MIN"
C    1"     DIP   DIPD     INT"5X"DIP    DIPD"6X"VPS ")
C
C     IP AND JP ARE BLOCK COORDS.
C
C     IP=XX(I1)/EW+ONE
C     JP=XX(I2)/EW+ONE
      IF(MODUL.NE.MODDD)GO TO 5
C
C     ADD DD CONTRIBUTIONS.
C
      DO 60 K=1,NLSX
      DO 60 L=1,NLSY
      CALL LCOOR(XEL,K,L,HL)
      RSQ=DIST2(XX,XEL)
      IF(RSQ.LT.TRNDD)GO TO 61
      R=SQRT(RSQ)
      DIS(I1)=XLARG(K,L)/RNUM
      DIS(I2)=YLARG(K,L)/RNUM
      DIS(I3)=ZLARG(K,L)/RNUM
      CALL SIGDD(TRAC,DIS,XX,XEL,ARL)
      GO TO 60
   61 CONTINUE
      KB=(K-I1)*ISMAL+I1
      KE=K*ISMAL
      LB=(L-I1)*JSMAL+I1
      LE=L*JSMAL
      DO 62 I=KB,KE
      DO 62 J=LB,LE
      CALL LCOOR(XC,I,J,H)
      RSQ=DIST2(XX,XC)
C
```

```
      DIS(I1)=DX(I,J)
      DIS(I2)=DY(I,J)
      DIS(I3)=DZ(I,J)
      IF(RSQ.LT.TRN4)GO TO 22
      R=SQRT(RSQ)
      CALL SIGDD(TRAC,DIS,XX,XC,AR)
      GO TO 62
   22 IF(RSQ.LE.TRN9)GO TO 24
      IB=1
      IE=4
      GO TO 25
   24 IF(RSQ.LE.TRN49)GO TO 23
      IB=5
      IE=13
      GO TO 25
   23 IB=15
      IE=63
      TYPE "49",I,J
   25 CONTINUE
      XEL(3)=ZERO
      DO 63 NINT=IB,IE
      W1=WGH(NINT)
      XEL(I1)=XC(I1)+GX(NINT)
      XEL(I2)=XC(I2)+GY(NINT)
      RSQ=DIST2(XX,XEL)
      R=SQRT(RSQ)
      TYPE "NINT=",NINT
      CALL SIGDD(TRAC,DIS,XX,XEL,W1)
   63 CONTINUE
   62 CONTINUE
   60 CONTINUE
    5 CONTINUE
C
C     GET PRIMITIVE STRESSES.
C
      CALL NULL(DC,XX)
      CALL VLTOG(DC,ALPHA,ORIG)
      DEPTH=RHO*9.814DC(3)
  106 FORMAT(/" POINT NUMBER "I3" IS TOO CLOSE TO THE REEF PLANE")
      WRITE(IRITE,245)KK,IXP,IYP
  190 FORMAT(/" BENCH MARK TIME",/" ***************",//
     1" STRESSES IN UNITS OF DEN*MLTRES*9.8/1000000",/
     2" STRESSES IN MPA AND DISPLACEMENTS IN METRES"
     1" IF DENSITY IS IN KG/M**3",/
     3" COMPRESSION IS +VE",/
     1/" UX,UY,UZ, ARE DISPLACEMENTS",/
     3" THE COORDINATE SYSTEM IS AS FOLLOWS",/
     4"        Z AXIS NORMAL TO REEF PLANE ( +VE UP )",/
     5"        X AXIS IS +VE  DOWN DIP",/
     6"        Y AXIS IS TO THE RIGHT (ALONG STRIKE)",/
     I" VPS IS THE PRIMITIVE VERTICAL NORMAL STRESS",/
     6" DIPD AND DIP RELATIVE TO ABOVE COORD SYSTEM",/
     7" ZD IS DIST FROM REEF PLANE TO BENCHMARK",/)
C 991 FORMAT(1X,6F10.2)
      WRITE(IRITE,105)
  105 FORMAT(//"    NO",T10"X"T18"Y"T26"Z"T36"U"T44"V"T52"W"T66"OXX"
     1T76"OXY"T84"OXZ"T92"OYY"T100"OYZ"T108"OZZ",/)
      WRITE(IRITE,101)KK,XX,W,((STRES(I,J),J=1,3),I=1,3)
  101 FORMAT(1X,I4,2X,3(F7.3,1X),2X,3(F7.4,1X),T65,6(F7.3,1X))
      IF(ISSH(8).EQ.1)DEPTH=-1
```

```
      DO 30 I=1,3
      DO 30 J=I,3
   30 STRES(I,J)=STRES(I,J)+DEPTH*PRIM(I,J)
      WRITE(IRITE,103)((STRES(I,J),J=I,3),I=1,3)
  103 FORMAT(30X"TOTAL STRESSES ARE."T65,6(F7.3,1X))
      IF(ISSW(7).EQ.NIL)CALL PRINK(STRES,IRITE)
      IF(ISSW(9).EQ.1)GO TO 1
      CALL VLTOG(XX,ALPHA,ORIG)
      CALL VLTOG(W,ALPHA,FALL)
      WRITE(IRITE,334)
  334 FORMAT(/" GLOBAL VALUES ARE")
      WRITE(IRITE,101)KK,XX,W
    1 CONTINUE
      CALL TTT(IRITE)
      CALL BACK
      STOP
      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  PRIN(3,3),PRIM(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),ISOU(3),XNUM(3),NCOM(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,NDINL,NDINN,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,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
C
C     SUBPROGRAM TO CALCULATE INFL COEFFICIENTS ON BOUNDARY ELEMENTS.
C
      DIMENSION KIND(9,9),GX(64),GY(64),WGH(64)
      DIMENSION XC(3),XEL(3),XNL(3),XE(3),XR(3),TEMP(3),DLC(3)
      EQUIVALENCE (XC,XNUM),(KIND,OZZT),(GX,XLARG),(GY,YLARG),(WGH,ZLARG)
      ISTAR=ICHAN
      IBUF=NIL
      IDISK=NIL
      CALL OPEN(6,"LINDEX",3,IER)
C     IF(IER.NE.1)TYPE "LINDEX OPEN ER",IER
      CALL OPEN(7,"BVECT",3,IER)
C     IF(IER.NE.1)TYPE "BVECT OPEN ER",IER
C     TYPE "TRANLL,TRANL,NIL,IDISK",TRANLL,TRANL,NIL,IDISK
C
C     B.E. AS RECEIVERS, B.E. AS EMITTERS.
C
      DO 512 I=1,9
      DO 512 J=1,9
  512 KIND(I,J)=NIL
      DO 511 ML=1,NLUMP
C     TYPE "MLUMP=",ML
      MEL=LKOD(ML)
      MOD1=LMOD(ML)
      NS=NSYM(MOD1)
```

```
C
      CALL MODSET(MOD1)
C     CALL LSET(ML,XRL,XLUMP)
      NS=NSYM(MOD1)
      DO 19 NL=1,NLUMP
C     TYPE "NLUMP=",NL
      LIND(NL)=NIL
      MOD2=LMOD(NL)
      IF(MOD2.NE.MOD1)GO TO 19
      ARL=ALUMP(NL)
      CALL LSET(NL,XRL,XLUMP)
      RSQ=DIST2(XRL,XRL)
      RATIO=RSQ/ARL
      IF(RATIO.LT.TRANLL)GO TO 19
      LIND(NL)=1
      CALL TUO(TT,UU)
      DO 18 ISYM=1,NS
      IDIR=NDIR(ISYM,NS)
      IF(IDIR.EQ.NIL)GO TO 18
      CALL LESET(NL,XEL,DLC)
      RSQ=DIST2(XRL,XEL)
C
C     LUMP-LUMP COEFFICIENTS.
C
      R=SQRT(RSQ)
      WCK=CK*ARL/RSQ
      WCR=ARL*CONS/R
C     TYPE "LUMP-LUMP",ML,NL,ISYM
      CALL TTUU(XEL,XRL,WCK,WCR,DLC)
   18 CONTINUE
      CALL LOAD
   19 CONTINUE
C
C      INSERT DD LUMP COEFFS HERE LATER.
C
      DLC(1)=ZERO
      DLC(2)=ZERO
      DLC(3)=ONE
      AR=4.*PHI
      TRN4=3.*AR
      TRN9=1.*AR
      TRN49=.6*AR
      ARL=4.*RL*RL
      DO 70 K=1,NLSX
      DO 70 L=1,NLSY
      KIND(K,L)=NIL
      IF(MOD1.NE.MODDD)GO TO 70
      CALL LDOOR(XEL,K,L,RL)
      RSQ=DIST2(XEL,XRL)
      RATIO=RSQ/ARL
      IF(RATIO.LT.TRANDL)GO TO 70
      CALL DOTT(XRL,XEL,ARL,DLC,I1,I1)
C     TYPE "DD LUMP-LUMP",NL,K,L,RATIO
      KIND(K,L)=ONE
   70 CONTINUE
C
C     LUMP-ELT COEFFICIENTS.
C
      TYPE "START 1 LOOP"
      DO 1 ME=1,MEL
      M=LUMPD(ML,ME)
```

```
C
      KODR=IABS(KOD(N))
C     CALL NULL(TEMP,FALL)
      CALL XCEN(INDEX,M,X,XR)
      TYPE "3 LOOP ML=",ML
C     DO 3 NL=1,NLUMP
      IF(LIND(NL).EQ.1)GO TO 3
      ARL=ALUMP(NL)
C     MOD2=LMOD(NL)
      IF(MOD2.NE.MOD1)GO TO 3
      NEL=LKOD(NL)
C  C  TYPE "INTO LSET"
      CALL LSET(NL,XEL,XLUMP)
   C  USE XRL HERE TO GIVE ONE LIND VALUE ONLY
C     RSQ=DIST2(XRL,XLL)
      RATIO=RSQ/ARL
      IF(RATIO.LT.TRANL)GO TO 7
C     LIND(NL)=2
      CALL TUO(TT,UU)
      TYPE "3 9 L LUMP-ELT NL=",NL
C     DO 9 ISYM=1,NS
      IDIR=NDIR(ISYM,NS)
      IF(IDIR.EQ.NIL)GO TO 9
C     CALL LSET(NL,XEL,DLC)
      RSQ=DIST2(XR,XEL)
      R=SQRT(RSQ)
C     WCK=CK*ARL/RSQ
      WCR=ARL*CONS/R
      CALL TTUU(XEL,XR,WCK,WCR,DLC)
C   9 CONTINUE
      CALL LOAD
      GO TO 3
C     7 CONTINUE
C
C     ELT-ELT COEFFICIENTS
C
C     DO 6 NE=1,NEL
      N=LUMPO(NL,NE)
C     KODE=IABS(KOD(N))
      CALL TUO(TT,UU)
      DO 111 ISYM=1,NS
C     IDIR=NDIR(ISYM,NS)
      IF(IDIR.EQ.NIL)GO TO 111
      IF(M.NE.N)GO TO 30
C     IF(ISYM.GT.1)GO TO 30
      CALL DIAGCO(M)
      GO TO 111
C  30 CONTINUE
      NOD1=INDEX(N,1)
      NOD2=INDEX(N,2)
C     NOD3=INDEX(N,IS)
      NOD4=INDEX(N,4)
      CALL DCOSV(NOD1,NOD2,NOD3,DLC,X)
C     CALL SYMFIX(FALL,NMULT,DLC,ISYM)
      AR=AREA(N)
      NQ=1
C     IF(KOD(N).GT.0)GO TO 50
      NQ=2
      AR=AREAF(NOD1,NOD2,NOD3,X)
C  50 CONTINUE
      DO 11 NTO=11,NQ
C
```

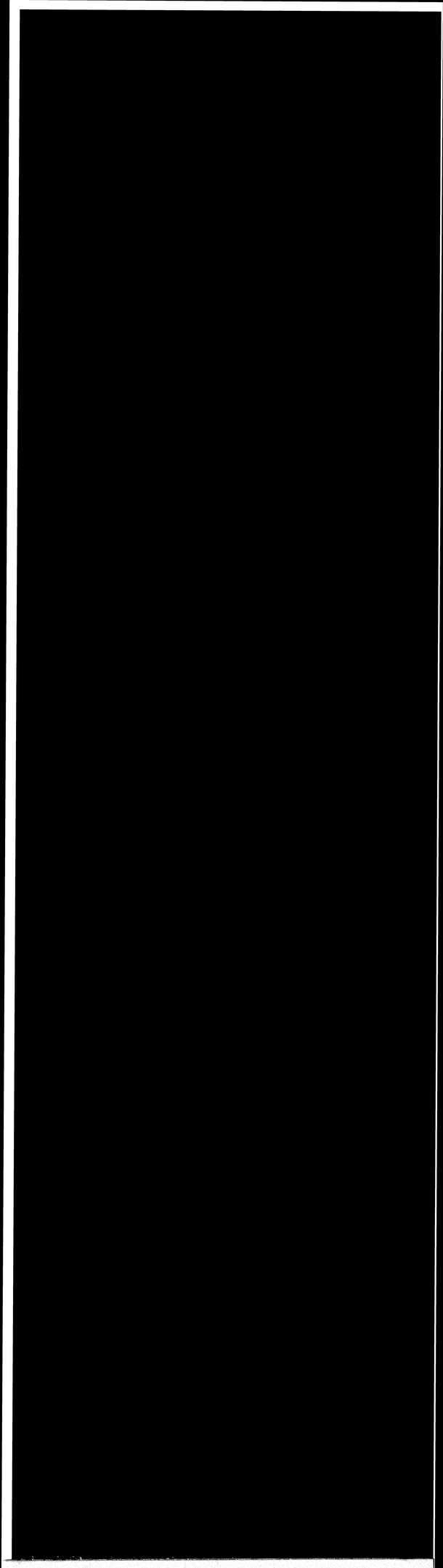
```
      IF(NTQ.EQ.1)GO TO 53
      AR=AREAF(NOD1,NOD3,NOD4,X)
      NOD2=NOD4
   53 CONTINUE
      CALL XCENT(NOD1,NOD2,NOD3,XC,X)
      CALL SYMFIX(SYM,NMULT,XC,ISYM)
      NINTB=11
      NINTE=11
      NUMTRI=1
      MINT=1
      IN=NOD1
      JN=NOD2
      KN=NOD3
      U1=AR
C
      RSQ=DIST2(XR,XC)
      RATIO=RSQ/AR
      RCL=RSQ/20.
C     IF(RATIO.LT.0.1)WRITE(IRITE,100)RATIO,M,N,NIL
C100 FORMAT(" CHECK ELTS TOO CLOSE RATIO,M,N,ER TYPE="F6.3,3I5)
      IF(RATIO.GE.TRAN5)GO TO 20
      MINT=3
      NINTB=3
      NINTE=10
      IF(RATIO.GE.TRAN7)GO TO 20
      MINT=7
      NINTB=1
      NINTE=7
      IF(RATIO.GT.TRAN21)GO TO 20
      MINT=21
      KN=NODE+1
      AR=AR/3.
      NUMTRI=3
      CALL XCENT(NOD1,NOD2,NOD3,XEL,X)
      X(KN,1)=XEL(1)
      X(KN,2)=XEL(2)
      X(KN,13)=XEL(3)
      IF(RATIO.GT.TRAN42)GO TO 20
      MINT=42
      NUMTRI=6
      AR=AR/2.
      KN=NODE+1
      JN=NODE+2
   20 CONTINUE
C
C     SUBSECTION FOR NUMERICAL QUADRATURE.
C
C     INFLUENCE OF ELEMENT N ON ELEMENT M (N,NE,M).
C
C     TYPE MINT,N
      DO 11 NTRI=I1,NUMTRI
      IF(NUMTRI.NE.6)GO TO 43
      IF(NTRI.EQ.1.OR.NTRI.EQ.6)IN=NOD1
      IF(NTRI.EQ.2.OR.NTRI.EQ.3)IN=NOD2
      IF(NTRI.EQ.4.OR.NTRI.EQ.5)IN=NOD3
      IF(NTRI.GT.2)GO TO 41
      DO 441 L=11,13
  441 X(JN,L)=(X(NOD1,L)+X(NOD2,L))*HALF
      GO TO 21
   41 IF(NTRI.GT.4)GO TO 42
```

```
      DO 442 L=I1,I3
  442 X(JN,L)=(X(NOD2,L)+X(NOD3,L))*HALF
      GO TO 21
   42 DO 443 L=I1,I3
  443 X(JN,L)=(X(NOD3,L)+X(NOD1,L))*HALF
      GO TO 21
   43 CONTINUE
      IF(NTR1.EQ.1)GO TO 21
      IF(NTR1.EQ.2)IN=NOD3
      IF(NTR1.LT.3)GO TO 21
      IN=NOD1
      JN=NOD3
   21 CONTINUE
      DO 11 NUM=NINTB,NINTE
      W1=WGHT(NUM)*AR
      IF(NINTB.EQ.1)GO TO 64
      DO 24 L=I1,I3
   24 XNUM(L)=COORD(NUM,1)*X(IN,L)+COORD(NUM,2)*X(JN,L)+COORD(NUM,I3)
     1*X(KN,L)
      CALL SYMFIX(SYM,NMULT,XNUM,ISYM)
      RSQ=DIST2(XR,XNUM)
C     IF(RSQ.LT.RCL)WRITE(IRITE,100)RATIO,M,N,IONE
   64 R=SQRT(RSQ)
C
C     INFL FOR FREE SURFACE ELEMENTS.
C
      WCK=W1*CK/RSQ
      IF(KODE.NE.KODK)GO TO 500
      IF(KODE.NE.8)GO TO 500
      WCK=WCK/R
      TT(3,1)=TT(3,1)+WCK*(XNUM(1)-XR(1))
      TT(3,2)=TT(3,2)+WCK*(XNUM(2)-XR(2))
      GO TO 11
  500 CONTINUE
C
C     INFL FOR NORMAL ELEMENTS.
C
      WCR=W1*CONS/R
      CALL TTUU(XNUM,XR,WCK,WCR,DLC)
   11 CONTINUE
  111 CONTINUE
      CALL LOAD1(N,TEMP,KODR)
    6 CONTINUE
    3 CONTINUE
      TYPE "START DD LOOPS"
C
C     START DD LOOPS WITH DD AS EMITTERS.
C
      IF(NOD1.NE.NODDD)GO TO 5
      DLC(1)=ZERO
      DLC(2)=ZERO
      DLC(3)=ONE
      ARL=4.*HL*HL
C
C     START LUMP LOOPS
C
      AR=4.*HK*H
      CALL GAUSS(GX,GY,WGH,H,AR)
      DO 60 K=1,NLSX
      DO 60 L=1,NLSY
```

```
C
      IF(KIND(K,L).EQ.1)GO TO 60
C     TYPE "DD:K,L",K,L
      CALL LCOOR(XEL,K,L,HL)
C     XRL HERE TO GIVE ONE VALUE OF KIND
      RSQ=DIST2(XRL,XEL)
      RATIO=RSQ/ARL
      IF(RATIO LT. TRANDD)GO TO 61
      KIND(K,L)=12
      RSQ=DIST2(XR,XEL)
      CALL DDTT(XR,XEL,ARL,DLC,I1,I1)
      GO TO 60
   61 CONTINUE
C
C     ELT-ELT COEFFS
C
      KB=(K-1)*ISMAL+1
      KE=K*ISMAL
      LB=(L-1)*JSMAL+1
      LE=L*JSMAL
      DO 62 I=KB,KE
      DO 62 J=LB,LE
C     TYPE "ELT-ELT I,J,M",I,J,M
      CALL LCOOR(XEL,I,J,M)
      RSQ=DIST2(XR,XEL)
      IF(RSQ LT. TRN4)GO TO 22
      CALL DDTT(XR,XEL,AR,DLC,I1,I1)
      GO TO 62
   22 IF(RSQ.LE.TRN9)GO TO 28
      IB=1
      IE=4
      GO TO 25
   28 IF(RSQ.LE.TRN49)GO TO 23
      IB=5
      IE=13
      GO TO 25
   23 IB=15
      IE=63
   25 TYPE IB
      XC(3)=ZERO
      IBE=IE-IB+1
      IDUMP=NIL
      DO 63 NINT=IB,IE
      IDUMP=IDUMP+1
      WI=WGH(NINT)
      XC(I1)=XEL(I1)+GX(NINT)
      XC(I2)=XEL(I2)+GY(NINT)
      RSQ=DIST2(XR,XC)
      CALL DDTT(XR,XC,WI,DLC,IDUMP,IBE)
   63 CONTINUE
   62 CONTINUE
   60 CONTINUE
    5 CONTINUE
      WRITE BINARY (7) TEMP
    1 CONTINUE
      IBM=IBUF-2
      WRITE(IRITE,200)ISTAR,IBUF,(IBUF(J),J=IBM,IBUF),LIND
  200 FORMAT(216,3F10.6,1X,6011)
      WRITE BINARY (6) LIND,KIND
  511 CONTINUE
      LAST=ISTAR-1
```

```
      TYPE "END BEMINF"
      CALL RESET
      CALL BACK
      STOP
      END
*****************************************************************************************************************
C
C     CALCULATE TRANSFORMATION MATRIX FOR LOCAL COORD SYSTEM.
C
C     O ARRAY IS THE ORIGIN OF THE LOCAL SYSTEM IN GLOBAL COORDS.
C     X IS A POINT ON THE LOCAL + X-AXIS,Y ON THE THE LOCAL + Y-AXIS
C     A IS THE TRANSFORMATION MATRIX CONSISTING OF DIRECTION COSINES.
C     EACH ROW CONSISTS OF DC'S OF A LOCAL COORD AXIS REL TO GLOBAL AXES.
C
      SUBROUTINE DCOSA(O,X,Y,A)
      DIMENSION O(3),X(3),Y(3),A(3,3)
      RX=SQRT(DIST2(O,X))
      RY=SQRT(DIST2(O,Y))
      DO 1 I=1,3
      A(1,I)=(X(I)-O(I))/RX
    1 A(2,I)=(Y(I)-O(I))/RY
      A(3,1)=A(1,2)*A(2,3)-A(1,3)*A(2,2)
      A(3,2)=A(1,3)*A(2,1)-A(1,1)*A(2,3)
      A(3,3)=A(1,1)*A(2,2)-A(1,2)*A(2,1)
C
C     CHECK VALIDITY OF CHOICE
C
      PROD=A(3,1)**2+A(3,2)**2+A(3,3)**2
      TYPE "CROSS PRODUCT OF X- AND Y-AXES IS",PROD
      RETURN
      END
*****************************************************************************************************************
C     SUBPROGRAM TO CALCULATE INFL COEFFS FOR DD ELEMENTS.
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  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,TRANUL,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,DLL,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,NLLEL,NRECO,KODE,ISSW(15),ISW(15)
C
      DIMENSION XEL(3),TEMP(3),DLC(3),XC(3),XRL(3),XE(3),XR(3)
      EQUIVALENCE (XC,XNUM)
C
      IDISK=0
      CALL OPEN(6,"LINDD",3,IER)
      IF(IER.NE.1)TYPE "LINDD OP ER",IER
      CALL OPEN(7,"BVECDD",3,IER)
```

```
      IF(IER.NE.1)TYPE "BVECDD OP ER",IER
      CALL MODSET(MODDD)
      NS=NSYM(MODDD)
      KOOR=ICNE
      DO 511 K=1,NLSX
      DO 511 L=1,NLSY
      CALL LCOOR(XRL,K,L,HL)
C
C     LUMP-LUMP COEFFS.
C
      DO 19 NL=1,NLUMP
      NIL=0
      LIND(NL)=NIL
      MOD2=LMOD(NL)
      IF(MOD2.NE.MODDD)GO TO 19
      ARL=ALUMP(NL)
      CALL LSET(NL,XEL,XLUMP)
      RSQ=DIST2(XEL,XRL)
      RATIO=RSQ/ARL
C     TYPE "DDINF",K,L,NL,RATIO",K,L,NL,RATIO,ARL,TRANEL
      IF(RATIO.LT.TRANEL)GO TO 19
      LIND(NL)=1
      CALL TWO(TT,UU)
      DO 18 ISYM=1,NS
      IDIR=NDIR(ISYM,NS)
      IF(IDIR.EQ.NIL)GO TO 18
      CALL LESET(NL,XEL,ELC)
      RSQ=DIST2(XRL,XEL)
      R=SQRT(RSQ)
      CALL SSDD(XRL,XEL,ELC,ARL)
   18 CONTINUE
      CALL LOAD
   19 CONTINUE
C
      LUMP-ELT COEFFS
C
      KB=(K-11)*ISMAL+1
      KE=K*ISMAL
      LB=(L-11)*JSMAL+1
      LE=L*JSMAL
      DO 1 I=KB,KE
      DO 1 J=LB,LE
      CALL LCOOR(XRL,I,J,H)
      CALL NULL(TEMP,FALL)
      DO 3 NL=1,NLUMP
      IF(LIND(NL).EQ.1)GO TO 3
      ARL=ALUMP(NL)
      MOD2=LMOD(NL)
      IF(MOD2.NE.MODDD)GO TO 3
      NEL=LKCD(NL)
      CALL LSET(NL,XEL,XLUMP)
C     XRL,NOT XR TO KEEP LUMP SIMPLE
      RSQ=DIST2(XRL,XEL)
      RATIO=RSQ/ARL
      IF(RATIO.LT.TRANL)GO TO 7
      TYPE "LUMP-ELT",I,J,NL,RATIO
      LIND(NL)=2
      CALL TWO(TT,UU)
      DO 9 ISYM=1,NS
      IDIR=NDIR(ISYM,NS)
```



```
      AR=AR/3.
      NUMTRI=3
      CALL XCENT(NOD1,NOD2,NOD3,XEL,X)
      X(KN,1)=XEL(1)
      X(KN,2)=XEL(2)
      X(KN,3)=XEL(3)
      IF(PAT10.GT.TRAN42)GO TO 20
      MINT=42
      NUMTRI=4
      AR=AR/2.
      KN=NODE+1
      JN=NODE+2
   20 CONTINUE
C     IF(MINT.GT.1)TYPE MINT,N,ISYM
      DO 11 NTRI=1,NUMTRI
      IF(NUMTRI.NE.6)GO TO 43
      IF(NTRI.EQ.1.OR.NTRI.EQ.6)IN=NOD1
      IF(NTRI.EQ.2.OR.NTRI.EQ.3)IN=NOD2
      IF(NTRI.EQ.4.OR.NTRI.EQ.5)IN=NOD3
      IF(NTRI.GT.2)GO TO 41
      DO 441 LO=1,3
  441 X(JN,LO)=(X(NOD1,LO)+X(NOD2,LO))*HALF
      GO TO 21
   41 IF(NTRI.GT.4)GO TO 42
      DO 442 LO=1,3
  442 X(JN,LO)=(X(NOD2,LO)+X(NOD3,LO))*HALF
      GO TO 21
   42 DO 443 LO=1,3
  443 X(JN,LO)=(X(NOD3,LO)+X(NOD1,LO))*HALF
      GO TO 21
   43 CONTINUE
      IF(NTRI.EQ.1)GO TO 21
      IF(NTRI.EQ.2)IN=NOD3
      IF(NTRI.LT.3)GO TO 21
      IN=NOD1
      JN=NOD3
   21 CONTINUE
C     TYPE I,J,N,MINT
      DO 11 NUM=NINTB,NINTE
      IF(NINTB.EQ.11)GO TO 64
      W1=WGHT(NUM)*AR
      DO 24 LO=1,3
   24 XNUM(LO)=COORD(NUM,1)*X(IN,LO)+COORD(NUM,2)*X(JN,LO)+COORD(NUM,3)
     1*X(KN,LO)
      CALL SYMFIX(SYM,NMULT,XNUM,ISYM)
      RSQ=DIST2(XR,XNUM)
   64 R=SQRT(RSQ)
      CALL SSDD(XR,XNUM,DLC,W1)
   11 CONTINUE
  111 CONTINUE
      CALL LOAD1(N,TEMP,KODR)
    6 CONTINUE
    3 CONTINUE
      WRITE BINARY (7) TEMP
    1 CONTINUE
      WRITE BINARY (6) LIND
  511 CONTINUE
      CALL DUMPO(IBUF,IDISK,BUF,ISTAR,1)
      CALL RESET
      CALL BACK
```

```
      STOP
      END
*****************************************************************************************************************
C
C     OUTPUT DD ARRAYS.
C
      SUBROUTINE DDOUTQ(DX,DY,DZ,OZZT,NN,MM,IRITE)
      DIMENSION DX(25,25),DY(25,25),DZ(25,25),OZZT(25,25)
      WRITE(IRITE,100)
  100 FORMAT(" X-AXIS RIDE COMPONENTS ARE",/)
      CALL AROUT(DX,NN,MM,IRITE)
      WRITE(IRITE,101)
  101 FORMAT(/" Y-AXIS RIDE COMPONENTS ARE",/)
      CALL AROUT(DY,NN,MM,IRITE)
      WRITE(IRITE,102)
  102 FORMAT(/" CLOSURES OF DD ELEMENTS ARE",/)
      CALL AROUT(DZ,NN,MM,IRITE)
      WRITE(IRITE,103)
  103 FORMAT(/" TOTAL NORMAL STRESSES ON DD ELEMENTS ARE",/)
      CALL AROUT(OZZT,NN,MM,IRITE)
      RETURN
      END
*****************************************************************************************************************
      SUBROUTINE DDREAD
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  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),GRIG(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     READ IN DD LAYOUT.
C
      DO 3 I=1,NN
    3 READ(IREAD,103)(MINE(I,J),J=1,MM)
  103 FORMAT(40A1)
C
C     WRITE OUT LAYOUT
C
      IF(ISW(7).NE.2)WRITE(IRITE,117)
      DO 16 I=1,NN
      IF(ISW(7).EQ.2)GO TO 16
      WRITE(IRITE,12)I,(MINE(I,L),L=1,MM)
   16 CONTINUE
C
   12 FORMAT(" ROW",I3,3X,5(10A1,1X)," DEPTH ="F6.1,3X"NORMAL"
     1" ="F6.1,"  DIP AND STR SHEAR ="F6.1,2X,F6.1)
```

```
C
117   FORMAT(/1X,"DD LAYOUT IS AS BELOW: ",/)
      RETURN
      END

*****************************************************************************************************************************
C     SUBROUTINE TO SOLVE FOR DD VALUES
C
      SUBROUTINE DDSOL(JOUT,VECT,AMAX)
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  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  IDFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELEL,NREGO,KODE,ISSW(15),ISW(15)
C
      DIMENSION VECT(220,3),TRAC(3),DIS(3),SUM(3),SUML(3),TEMP(3)
C
C     SIGN CHANGES IN THIS SUBROUTINE TO GIVE TENSION + CONVENTION.
C     CHANGES ABOVE   C.........*
C
      TYPE "INTO DDSOL"
      DC(1)=ZERO
      DC(2)=ZERO
      DC(3)=ONE
      IFLT=2H+
      IMIN=2HM
      ISOL=2H.
      ISHR=2H*
      ITOT=2HC
      ITEN=2HT
      RSMAL=FLOAT(ISMAL)
      CFAR=RSMAL**3
      DO 511 K=1,NLSX
      KB=(K-IONE)*ISMAL+IONE
      KE=K*ISMAL
      DO 511 L=1,NLSY
      LB=(L-IONE)*JSMAL+IONE
      LE=L*JSMAL
      READ BINARY (2) LIND
C     TYPE "LIND",K,L
C
C     LUMP-LUMP SUMS
C
      CALL NULL(SUML,FALL)
      DO 19 NL=1,NLUMP
      IF(LIND(NL).NE.1)GO TO 19
      CALL TUL(NL,TRAC,DIS)
      CALL SUMUP(SUML,DIS,TRAC)
C     TYPE NL,SUML,"  LUMP SUMS"
```

```
C
   19 CONTINUE
      SUMX=ZERO
      SUMY=ZERO
      SUMZ=ZERO
      DO 50 KL=1,NLSX
      IDIFF=IABS(K-KL)+IONE
      DO 50 LL=1,NLSY
      JDIFF=IABS(L-LL)+IONE
      IF(IDIFF.GT.2)GO TO 51
      IF(JDIFF.LE.2)GO TO 50
   51 XL=XLARG(KL,LL)
      YL=YLARG(KL,LL)
      ZL=ZLARG(KL,LL)
      DCOEF=DK(IDIFF,JDIFF)
      IF(KL.LT.K)DCOEF=-DCOEF
      IF(LL.LT.L)DCOEF=-DCOEF
      SUMX=SUMX+BMA(IDIFF,JDIFF)*XL-DCOEF*YL
      SUMY=SUMY+BMA(JDIFF,IDIFF)*YL-DCOEF*XL
      SUMZ=SUMZ+AK(IDIFF,JDIFF)*ZL
   50 CONTINUE
      SUML(1)=-SUML(1)+SUMX/CPAR
      SUML(2)=-SUML(2)+SUMY/CPAR
      SUML(3)=-SUML(3)+SUMZ/CPAR
C.....*
C
C     START INNER LOOPS.
C
      DO 1 I=KB,KE
      DO 1 J=LB,LE
      READ BINARY (13) TEMP
C     TYPE "TEMP",TEMP
      CALL NULL(SUM,FALL)
      CALL SUMELT(SUM,VECT,MODED,KODR,J)
      SUMX=-SUM(1)-TEMP(1)+SUML(1)
      SUMY=-SUM(2)-TEMP(2)+SUML(2)
      SUMZ=-SUM(3)-TEMP(3)+SUML(3)
C.....*....*
      DO 60 KL=1,NLSX
      IDIFF=IABS(K-KL)+IONE
      DO 60 LL=1,NLSY
      JDIFF=IABS(L-LL)+IONE
      IF(IDIFF.GT.2)GO TO 60
      IF(JDIFF.GT.2)GO TO 60
      IB=(KL-IONE)*ISMAL+IONE
      IE=KL*ISMAL
      JB=(LL-IONE)*JSMAL+IONE
      JE=LL*JSMAL
      DO 62 II=IB,IE
      M=IABS(I-II)+IONE
      DO 62 JJ=JB,JE
      N=IABS(J-JJ)+IONE
      XL=DX(II,JJ)
      YL=DY(II,JJ)
      ZL=DZ(II,JJ)
      DCOEF=DK(M,N)
      IF(II.LT.I)DCOEF=-DCOEF
      IF(JJ.LT.J)DCOEF=-DCOEF
      SUMX=SUMX+BMA(M,N)*XL-DCOEF*YL
      SUMY=SUMY+BMA(N,M)*YL-DCOEF*XL
   62 SUMZ=SUMZ+AK(M,N)*ZL
```

```
   60 CONTINUE
      WRITE(10,233)MINE(I,J),I,J
  233 FORMAT(1X,A1,2I3)
      OZZT(I,J)=SUMZ
      DENOMZ=AK(IONE,IONE)
      DENOMX=BMA(IONE,IONE)
      CALL LCOOR(XNUM,I,J,H)
      CALL PRIMDD(ALPHA,PRIM,OR16,XNUM,DC,W,RHO,ISSW)
C
C     UNMINED SEAM ELEMENT.
C
      IF(MINE(I,J).NE.ISOL)GO TO 20
      DENOMX=DENOMX-GSEAM/THIK
      DENOMZ=DENOMZ-ESEAM/THIK
      SUMX=SUMX-GSEAM*DX(I,J)/THIK
      SUMY=SUMY-GSEAM*DY(I,J)/THIK
      SUMZ=SUMZ-ESEAM*DZ(I,J)/THIK
      OZZT(I,J)=OZZT(I,J)+W(3)
      GO TO 100
   20 CONTINUE
      IF(MINE(I,J).NE.IMIN.AND.MINE(I,J).NE.ITOT)GO TO 40
      SUMX=SUMX+W(1)
      SUMY=SUMY+W(2)
      SUMZ=SUMZ+W(3)
      OZZT(I,J)=SUMZ
   40 CONTINUE
  100 CONTINUE
C
C     INCREASE RELAXION FACTOR FOR DD ELEMENTS.
C
      DX(I,J)=DX(I,J)-OMEGA*1.5*SUMX/DENOMX
      DY(I,J)=DY(I,J)-OMEGA*1.5*SUMY/DENOMX
      DZ(I,J)=DZ(I,J)-OMEGA*1.5*SUMZ/DENOMZ
C
C     CHECK FOR TOTAL CLOSURE
C
      IF(MINE(I,J).EQ.ITOT)MINE(I,J)=IMIN
      IF(DZ(I,J).LT.THIK)GO TO 13
      DZ(I,J)=THIK
      IF(MINE(I,J).EQ.IMIN)MINE(I,J)=ITOT
      GO TO 14
   13 CONTINUE
      AVE=(ABS(SUMX)+ABS(SUMY)+ABS(SUMZ))/2.5
      IF(AVE.GT.AMAX)AMAX=AVE
      IF(DEL.GT.AVE)GO TO 14
      JOUT=JOUT+IONE
C     IF(AVE.GT.(DLL*4.))TYPE "I,J,DZ,SUMX,Y,Z",I,J,DZ(I,J),SUMX,SUMY,SUMZ
   14 CONTINUE
    1 CONTINUE
  511 CONTINUE
      TYPE "LEAVING DDSOL"
      RETURN
      END

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

C
C     SUBROUTINE TO CALCULATE AND DUMP DD INFL COEFFICIENT.
C
      SUBROUTINE DDTT(XRD,XLD,AR,DCO,IDUMP,IBE)
```

```
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),DU(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
      DIMENSION XRD(3),XED(3),DCO(3)
      IF(IDUMP.NE.1)GO TO 5
      CALL TUC(TT,UU)
      IF(IBUF.GT.LIM)CALL DUMPO(IBUF,IDISK,BUF,ISTAR,1)
    5 R=SQRT(RSQ)
      WCK=AR*CK/RSQ
      DO 1 K=1,3
    1 RCOM(K)=(XED(K)-XRD(K))/R
      DRDN=WCK*RCOM(3)
      DO 2 I=1,3
      RCI=RCOM(I)
      RIV3=RCI*V3
      DCI=DCO(I)
      TT(I,I)=TT(I,I)+DRDN*(1.+RIV3*RCI)
      IF(I.EQ.3)GO TO 4
      IPI=I+IONE
      DO 2 J=IPI,3
      RCJ=RCOM(J)
      DCJ=DCO(J)
      TF=(DCI*RCJ-DCJ*RCI)*WCK
      TFUNC=DRDN*RIV3*RCJ
      TT(I,J)=TT(I,J)+TFUNC+TF
    2 TT(J,I)=TT(J,I)+TFUNC-TF
    4 CONTINUE
      IF(IDUMP.LT.IDE)RETURN
      DO 3 I=1,3
      DO 3 J=1,3
      IBUF=IBUF+1
    3 BUF(IBUF)=TT(I,J)
      RETURN
      END
```

****

```
C
C     SET UP GAUSS QUADRATURE CONSTANTS AND WEIGHTS.
C
      SUBROUTINE GAUSS(GX,GY,W,H,AR)
      DIMENSION GX(64),GY(64),W(64),G(7),X(7)
C
C     CONSTANTS.
C
      V1=0.57735*H
      V2=0.7746*H
```

```
C
      V3=0.0
C     W1=0.25*AR
      W2=0.27777777
      W3=0.4444444
C     G(1)=0.94910*H
      G(2)=0.74153*H
      G(3)=0.40584*H
C     G(4)=V3
      G(5)=-G(3)
      G(6)=-G(2)
C     G(7)=-G(1)
      X(1)=0.12948/2.
      X(2)=0.27970/2.
C     X(3)=0.38183/2.
      X(4)=0.41795/2.
      X(5)=X(3)
C     X(6)=X(2)
      X(7)=X(1)
C
C     COORDS
C
C 4
C     GX( 1)= V1
      GY( 1)= V1
      GX( 2)= V1
C     GY( 2)=-V1
      GX( 3)=-V1
      GY( 3)= V1
C     GX( 4)=-V1
      GY( 4)=-V1
C 9
C     GX( 5)= V3
      GY( 5)= V3
      GX( 6)=-V2
C     GY( 6)= V2
      GX( 7)= V3
      GY( 7)= V2
C     GX( 8)= V2
      GY( 8)= V2
      GX( 9)=-V2
C     GY( 9)= V3
      GX(10)= V2
      GY(10)= V3
C     GX(11)=-V2
      GY(11)=-V2
      GX(12)= V3
C     GY(12)=-V2
      GX(13)= V2
      GY(13)=-V2
C C 1
      GX(14)=V3
      GY(14)=V3
C C 49
      NINT=14
      DO 1 I=1,7
C     DO 1 J=1,7
      NINT=NINT+1
      GX(NINT)=G(I)
C     GY(NINT)=G(J)
    1 W(NINT)=X(I)*X(J)*AR
```

```
C
C     WEIGHTS
C
C 4
      W( 1)=W1
      W( 2)=W1
      W( 3)=W1
      W( 4)=W1
C 9
      W( 5)=W3*W3*AR
      W( 6)=W2*W2*AR
      W( 7)=W2*W3*AR
      W( 8)=W(6)
      W( 9)=W(7)
      W(10)=W(7)
      W(11)=W(6)
      W(12)=W(7)
      W(13)=W(6)
C 1
      W(14)=AR
      RETURN
      END
*****************************************************************************************************************
      SUBROUTINE GEOMET
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),LUMPQ(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),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,NREGO,KODE,ISSW(15),ISW(15)
C
C     SUBROUTINE TO CHECK GEOMETRY OF ELEMENTS.
C
      CALL EXEC(3,0)
      TOL=0.001
      DO 10 N=1,NELT
      IF(KOD(N).EQ.0)TYPE "ERROR 4",N
      DO 10 J=1,4
      IF(INDEX(N,J).GT.NODE)TYPE "ERROR 5",N,J
      IF(J.LT.4.AND.INDEX(N,J).LT.1)TYPE "ERROR 6",N,J
   10 CONTINUE
      DO 1 N=1,NELT
      I=INDEX(N,1)
      J=INDEX(N,2)
      K=INDEX(N,3)
      L=INDEX(N,4)
      IF(L.EQ.0.AND.KOD(N).LE.0)TYPE "ERROR 7",N
      CALL DCOSP(I,J,K,A,B,C,X)
      IF(KOD(N).GT.0)GO TO 5
```

```
      CALL DCOSP(I,K,L,A1,B1,C1,X)
      IF(ABS(A-A1).GT.TOL)TYPE "ERROR 8",N
      IF(ABS(B-B1).GT.TOL)TYPE "ERROR 8",N
      IF(ABS(C-C1).GT.TOL)TYPE "ERROR 8",N
    5 CONTINUE
      AREA(N)=AREAF(I,J,K,X)
      IF(KOD(N).LT.0)AREA(N)=AREA(N)+AREAF(I,K,L,X)
      IF(AREA(N).LT.TOL)TYPE "ERROR 9",N
    1 CONTINUE
      TYPE "3,1"
      RETURN
      END

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

      SUBROUTINE INITL
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),ILUMC(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),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  IRAN42,IRAN21,IRAN7,IRAN5,IRANLL,IRANL,IRANDL,IRANDD,FAC,WGHT(11),COORD(11,3),
     9  NODE,NEL1,NLUMP,NN,NM,NLSX,NLSY,NDIML,NDIMN,NDIM,ICMAL,JCMAL,NUMOR,NSLIP,NIL,MODDD,NS,
     *  ZERO,ONE,IONE,I1,I2,I3,THIRD,HALF,TWO,NITER,OMEGA,DEL,PI,A,B,C,R,K50,EM,EL,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,G,COHD,CTAR,THIK,&SEAM,6SEAM,COH,PHI,
     2  IBFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),INEAD,IRITE,NELEL,NRECO,KODE,ISW(15),ISW(15)
C
C     SUBROUTINE TO INITIALIZE THE LARGE ARRAYS.
C
      NN=25
      DO 1 I=1,NN
      DO 1 J=1,NN
      DX(I,J)=0
      DY(I,J)=0
      DZ(I,J)=0
      OZZT(I,J)=0
      MINE(I,J)=0
    1 CONTINUE
      RETURN
      END

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

      SUBROUTINE INITT
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  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),NDIR(8,8),NMULT(8,3),
     8  IRAN42,IRAN21,IRAN7,IRAN5,IRANLL,IRANL,IRANDL,IRANDD,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, LU, H, HL,
     1  E, V, V1, V2, V3, V4, V5, V6, V7, V8, V9, CONS, CK, CD, G, COND, CLAR, THIK, ESEAM, USEAM, COH, PHI,
     2  IBFSZ, ICHAN, LIM, IDISK, IBUF, ISTAK, BUF(1152), IREAD, IRITE, NELIL, NRECO, KODE, ISSW(15), ISW(15)
C
C     SUBROUTINE TO INITIALIZE THE MATRICES
C
      CALL EXEC(1,0)
      NDIM=220
      NDIMN=215
      NDIML=50
      IBFSZ=1152
      LIM=IBFSZ-18

      ZERO=0.
      NIL=0
      ONE=1.
      IONE=1
      TWO=2.
      HALF=0.5
      THIRD=1./3.
      I1=1
      I2=2
      I3=3
C
      DO 20 I=1,IBFSZ
   20 BUF(I)=ZERO
      DO 21 I=1,NDIML
   21 LIND(I)=NIL
      DO 1 I=1,3
      FALL(I)=ZERO
      DO 1 J=1,3
      DELTA(I,J)=ZERO
      DELTA(I,I)=ONE
      DO 1 K=1,3
    1 ALT(I,J,K)=ZERO
      ALT(1,2,3)=ONE
      ALT(2,3,1)=ONE
      ALT(3,1,2)=ONE
      ALT(1,3,2)=-ONE
      ALT(3,2,1)=-ONE
      ALT(2,1,3)=-ONE
      DO 2 I=1,NDIM
      AREA(I)=ZERO
      KOD(I)=NIL
      IFACE(I)=NIL
      INDEX(I,4)=NIL
      DO 2 J=1,3
      U(I,J)=ZERO
      T(I,J)=ZERO
      INDEX(I,J)=NIL
      IBC(I,J)=NIL
    2 CONTINUE
      THIRD=ONE/3.
      A1=0.05961587
      A2=0.79742699
      B1=0.47014206
      B2=0.10128651
      W1=0.225
      W2=0.1323941
```

```
      W3=0.12590913
C
      COORD(1,1)=THIRD
      COORD(1,2)=THIRD
      COORD(1,3)=THIRD
C
      COORD(2,1)=A1
      COORD(2,2)=B1
      COORD(2,3)=B1
C
      COORD(3,1)=B1
      COORD(3,2)=A1
      COORD(3,3)=B1
C
      COORD(4,1)=B1
      COORD(4,2)=B1
      COORD(4,3)=A1
C
      COORD(5,1)=A2
      COORD(5,2)=B2
      COORD(5,3)=B2
C
      COORD(6,1)=B2
      COORD(6,2)=A2
      COORD(6,3)=B2
C
      COORD(7,1)=B2
      COORD(7,2)=B2
      COORD(7,3)=A2
C
      COORD(8,1)=HALF
      COORD(8,2)=HALF
      COORD(8,3)=ZERO
C
      COORD(9,1)=HALF
      COORD(9,2)=ZERO
      COORD(9,3)=HALF
C
      COORD(10,1)=ZERO
      COORD(10,2)=HALF
      COORD(10,3)=HALF
C
      COORD(11,1)=THIRD
      COORD(11,2)=THIRD
      COORD(11,3)=THIRD
C
      WGHT(1)=W1
      WGHT(2)=W2
      WGHT(3)=W2
      WGHT(4)=W2
      WGHT(5)=W3
      WGHT(6)=W3
      WGHT(7)=W3
      WGHT(8)=THIRD
      WGHT(9)=THIRD
      WGHT(10)=THIRD
      WGHT(11)=ONE
C
      DO 3 J=1,3
      IF(J.LT.4)NMULT(1,J)=ICONE
```

```
      NDIR(1,J)=1
      DO 3 I=2,8
      IF(J LT 4)NMULT(I,J)=-1
      NDIR(I,J)=0
      NDIR(I,8)=1
    3 NDIR(J,J)=1
      NDIR(2,5)=IONE
      NDIR(2,6)=IONE
      NDIR(3,5)=IONE
      NDIR(3,7)=IONE
      NDIR(4,6)=IONE
      NDIR(4,7)=IONE
      NMULT(2,2)=IONE
      NMULT(2,3)=IONE
      NMULT(3,1)=IONE
      NMULT(3,3)=IONE
      NMULT(4,1)=IONE
      NMULT(4,2)=IONE
      NMULT(5,3)=IONE
      NMULT(6,2)=IONE
      NMULT(7,1)=IONE
      CALL EXEC(1,1)
C
      RETURN
      END
```

```
      SUBROUTINE INPUT
C
C     JACD
C
      COMMON KOD(220),IEC(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),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),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,TRANS,TRANLL,TRANL,TRANUL,TRANDU,FAC,WGHT(11),COORD(11,3),
     9  NODE,NELT,NLUMP,NN,MM,NLSX,NLSY,NDINL,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NSLIP,NIL,MODDD,NS,
     *  ZERO,ONE,IONE,II,IZ,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,CK,CD,G,COND,CPAR,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 ID(3),SIG(3,3),IDATA(5),XE(3)
C
C
C     SUBROUTINE TO READ INPUT VARIABLES AND SET CONSTANTS.
C
      CALL EXEC(2,0)
      IREAD = 1
      IDATA(1)=2HBE
      IDATA(2)=2HND
      IDATA(3)=2HAT
      IDATA(4)=2HA.
      IDATA(5)=2H
  100 FORMAT(30A2)
      CALL OPEN(IREAD,IDATA,IER)
      IF(IER NE 1)TYPE "OPEN ERROR",IER
```

```
      READ(IREAD,202)ISSW
      READ(IREAD,202)ISW
      IRITE=12
      IF(ISSW(5).EQ.1)IRITE=10
      READ(IREAD,100)(LMOD(I),I=1,30)
      WRITE(IRITE,254)(ISSW(I),I=1,9)
      CALL SISW
      WRITE(IRITE,154)
  154 FORMAT("1")
      WRITE(IRITE,153)
  153 FORMAT(/" ***************************************************",
     1"***************************************",/)
      WRITE(IRITE,106)(LMOD(I),I=1,30)
  106 FORMAT(1X,"",30A2)
      CALL ITT(IRITE)
      WRITE(IRITE,107)
  107 FORMAT(/1X," MIXED BOUNDARY ELEMENT METHOD 3 D ANALYSIS")
      WRITE(IRITE,153)
C
C     SET UP STANDARD CONSTANTS.
C
      RHO=2650.
      E=68700.
      V=0.22
      NUMOR=0
      NITER=100
      OMEGA=1.4
      DEL=0.001
      TRAN42=0.65
      TRANLL=30
C
C     READ AND WRITE INPUT DATA.
C
C
  254 FORMAT("     SWITCH CONTROLS ARE AS FOLLOWS",/
     *" ISSW 1 = 1 TO SKIP CALCULATION OF COEFFICIENTS.",T60,I1,/
     *" ISSW 2 = 1 TO SKIP CALCULATION OF T AND U.",T60,I1,/
     *" ISSW 2 = 2 TO CONTINUE CALCULATION OF T AND U",/
     *" ISSW 3 = 1 TO SKIP OUTPUT OF GEOMETRY.",T60,I1,/
     *" ISSW 4 = 1 TO SKIP OUTPUT OF LUMP GEOMETRY.",T60,I1,/
     *" ISSW 5 = 1 FOR OUTPUT TO SCREEN.",T60,I1,/
     *" ISSW 6 = 1 TO SKIP OUTPUT OF RESULTS IN LOACAL COORD SYSTEM",T60,I1,/
     *" ISSW 7 = 1 TO SKIP PRINCIPAL STRESS ROUTINE",T60,I1,/
     *" ISSW 8 = 1 IF PRIMITIVE STRESSES ARE NOT DEPTH DEPENDENT",T60,I1,/
     9" ISSW 9 = 1 TO SKIP RESULT OUTPUT IN GLOBAL COORD SYSTEM",T60,I1)

C
      READ(IREAD)ICHAN,TRANL,TRAN42,TRAN21,TRAN7,TRANS
      READ(IREAD)NITER,OMEGA,DEL,NSLIP
      READ(IREAD)NELT,NODE,NUMOR,NLUMP,LAYER
      READ(IREAD)XSYM,YSYM,ZSYM,THIK,COH,PHI
      READ(IREAD)((PRIM(I,J),J=1,3),I=1,3)
C
C
  202 FORMAT(25I1)
  101 FORMAT(/
     4        10X"MAX  ITERATIONS      = ",6X,I4,/
     5        10X"RELAXION FACTOR      = ",F10.2,/
     6        10X"CONVERGENCE LIMIT    = ",F10.5,/
     1        10X"X PLANE OF SYMMETRY = ",F10.2,/
```

```
     2          10X"Y PLANE OF SYMMETRY = ",F10.2,/
     3          10X"Z PLANE OF SYMMETRY = ",F10.2,/
     4          10X"THICKNESS OF SEAM   = ",F10.2,/
     5          10X"COHESION OF FAULT   = ",F10.2,/
     6          10X"PHI OF FAULT        = ",F10.1,/)
102 FORMAT(10X"NO OF ELEMENTS       = ",6X,I4,/
     1          10X"NO OF NODES          = ",6X,I4,/
     2          10X"NO OF BENCHMARKS     = ",6X,I4,/
     3          10X"NO OF LUMPS          = ",6X,I4,/
     4          10X"STARTING DISK BLOCK  = "6X,I4,/
     5          10X"TRANS FOR 42 PT      = "F10.2,/
     7          10X"TRANS FOR 21 PT      = ",F10.2,/
     6          10X"TRANS FOR  7 PT      = ",F10.2,/
     7          10X"TRANS FOR  3 PT      = ",F10.2,/
     8          10X"TRANS FOR LUMPING    = ",F10.2,/
     *,/)
C
      TRANLL=TRANL*3.
      TRANDL=TRANLL
      TRANDU=TRANL
C
      IF(LAYER.GT.10)STOP 1
      WRITE(IRITE,113)LAYER
113 FORMAT(" NO OF DIFFERENT ZONES =",I4,/)
      RHO=ZERO
      NIL=0
      MAXSYM=NIL
      DO 104 I=1,LAYER
      READ(IREAD)J,EMOD(J),PR(J),DEN(J),NSYM(J)
      IF(NSYM(J).GT.9.OR.NSYM(J).LE.0)STOP 22
      IF(NSYM(J).GT.MAXSYM)MAXSYM=NSYM(J)
      IF(I.NE.J)STOP 2
      RHO=RHO+DEN(I)
      WRITE(IRITE,105)EMOD(I),PR(I),DEN(I),I,NSYM(I)
104 CONTINUE
105 FORMAT(" E ="F10.2"   POISSONS RATIO ="F6.2"    RHO ="F10.4
   1"     ZONE NO ="I4,"   SYMMETRY CODE ="I4)
      RHO=RHO/FLOAT(LAYER)
      WRITE(IRITE,139)RHO
139 FORMAT(/" AVERAGE DENSITY USED FOR OVERBURDEN STRESS ="F10.4)
C
C     SET RECURSIVE CONSTANTS.
C
      PI=3.14159
C
C     READ IN COORDINATES AND BOUNDARY CONDITIONS.
C
      IF(NUMOR.LE.NIL)GO TO 4
      CALL OPEN(2,"BENCH.",3,IER)
      IF(IER.NE.1)TYPE "BENCH OPEN ERROR",IER
      IF(NUMOR.GT.220)NUMOR=220
      DO 5 I=1,NUMOR
      READ(IREAD)XX,YY,ZZ,MODUL
      WRITE(2,103)XX,YY,ZZ,MODUL
103 FORMAT(3F12.3,I4)
  5 CONTINUE
  4 CONTINUE
      NTEMP=NELT
      NLTEM=NLUMP
      IF(NELT.GT.NDIM.OR.NODE.GT.NDIMN.OR.NLUMP.GT.NDIML)STOP 5
```

```
      DO 1 NL=1,NLUMP
      TYPE NL,"=LUMP NO"
      READ(IREAD)LUMP,NUMEL,MOD1,MOD2
      IF(MOD1.GT.10.OR.MOD2.GT.10)STOP 3
      IF(NUMEL.GT.12)TYPE "ERROR 1",LUMP
      LNOD(LUMP)=NUMEL
      IF(LUMP.NE.NL)STOP 4
      LMOD(NL)=MOD1
      IWILL=0
      IF(MOD2.EQ.0.OR.MOD2.EQ.MOD1)GO TO 60
      NLTEM=NLTEM+1
      LNOD(NLTEM)=NUMEL
      LMOD(NLTEM)=MOD2
      IWILL=1
   60 CONTINUE
      DO 1 I=1,NUMEL
      READ(IREAD)N,(INDEX(N,J),J=1,4),KODE
      IF(INDEX(N,4).NE.NIL.AND.KODE.GT.N-1)KODE=-KODE
      IF(KOD(N).NE.NIL)TYPE "ERROR 2",N
      KOD(N)=KODE
      LUMPO(NL,I)=N
C
C     GENERATE SECOND ELEMENT AT INTERFACE
      IF(IWILL.EQ.0)GO TO 1
      NTEMP=NTEMP+1
      IFACE(N)=NTEMP
      IFACE(NTEMP)=N
      KOD(NTEMP)=KODE
      LUMPO(NLTEM,I)=NTEMP
      NNN=3
      IF(KODE.LT.0)NNN=4
      NNP=NNN+1
      DO 61 J=1,NNN
   61 INDEX(NTEMP,J)=INDEX(N,NNP-J)
    1 CONTINUE
      NELT=NTEMP
      NLUMP=NLTLM
      WRITE(IRITE,101)NITER,OMEGA,DEL,XSYM,YSYM,ZSYM,THIK,COH,PHI
      IF(PHI.GT.5.)PHI=TAN(PI*PHI/180.0)
      WRITE(IRITE,102)NELT,NODE,NUMOR,NLUMP,ICHAN,TRAN42,TRAN21
     1,TRAN7,TRANS,TRANL
      IF(NELT.GT.NDIM.OR.NODE.GT.NDIMN.OR.NLUMP.GT.NDIML)STOP 6
      SYM(1)=XSYM*TWO
      SYM(2)=YSYM*TWO
      SYM(3)=ZSYM*TWO
      DO 2 I=1,NODE
      TYPE "NODE = ",I
      READ(IREAD)NM,(X(I,J),J=1,3)
      IF(NM.NE.I)TYPE "ERROR 3",I,NM
    2 CONTINUE
      CALL BELL
      CALL GEOMET
C
C     CALCULATE B.C.'S
C
C     1  = SPECIFIED TRACTIONS.
C     2  = SPECIFIED DISPLACEMENTS.
C     3  =    "      Z-DISP
C     4  =    "      Y-DISP
C     5  =    "      X-DISP
```

```
C     6  =        B.C.
C     7  = INTERFACE ELEMENTS
C     8  = FREE SURFACE (ZERO TRACTIONS)
C
C     IBC = 1 FOR TRACTIONS , 2 FOR DISPLACEMENTS.
C
      IF(ISSW(3).NE.1)WRITE(IRITE,180)
      DO 3 N=1,NELT
      N1=INDEX(N,1)
      N2=INDEX(N,2)
      N3=INDEX(N,3)
      N4=INDEX(N,4)
      CALL DCOSV(N1,N2,N3,DC,X)
      IF(N4.EQ.NIL)CALL XCENT(N1,N2,N3,XE,X)
      IF(N4.NE.NIL)CALL XCENQ(N1,N2,N3,N4,XE,X)
      KODE=IABS(KOD(N))
      Z=XE(3)
      DEPTH=Z*9.8*RHO
      IF(ISSW(8).EQ.1)DEPTH=1.
      IF(KODE.NE.9.AND.KODE.NE.1.AND.KODE.NE.8)GO TO 11
      IF(KODE.EQ.8.AND.(DC(3).LT.0.99))TYPE "DC ER FOR KOD 8 ELT",N
      IF(KODE.EQ.8.AND.MAXSYM.GT.1)STOP 7
      DO 22 I=1,3
      IBC(N,I)=1
      DO 21 J=1,3
      SIG(I,J)=DEPTH*PRIM(I,J)
   21 T(N,I)=SIG(I,J)*DC(J)+T(N,1)
   22 CONTINUE
   11 CONTINUE
      IF(KODE.NE.2)GO TO 12
      DO 31 I=1,3
      IBC(N,I)=2
   31 U(N,I)=ZERO
   12 CONTINUE
      IF(KODE.LT.3.OR.KODE.GT.5)GO TO 14
      DO 41 I=1,3
      IBC(N,I)=IONE
      DO 41 J=1,3
      T(N,I)=0.
      IF(I.NE.(KODE-2))GO TO 41
      T(N,I)=ZERO
      IBC(N,I)=2
   41 CONTINUE
   14 CONTINUE
      IF(KODE.NE.6)GO TO 16
      READ(IREAD)NO,W,IB
      DO 26 I=1,3
      IF(IB(I).EQ.2)U(N,I)=W(I)
      IF(IB(I).EQ.1)T(N,1)=W(I)
      IBC(N,I)=IB(I)
   26 CONTINUE
   16 CONTINUE
      IF(KODE.LT.7.OR.KODE.EQ.8.OR.KODE.EQ.9)GO TO 17
      DO 27 I=1,3
      IBC(N,I)=2
   27 IBC(IFACE(N),I)=I1
   17 CONTINUE
  180 FORMAT(/T5"ELT"T12"X-COORD"T22"Y-COORD"T32"Z-COORD"T40"    NODE "
     1," NUMBERS"T58"AREA"T67"DIR-X"T77"DIR-Y"T87"DIR-Z",10X,"B-C",
     1T112"KODE",//)
```

```
      IF(ISSW(3).EQ.1)GO TO 3
      WRITE(IRITE,181)N,(XE(I),I=1,3),(INDFX(N,I),I=1,4)
     1,AREA(N),(DC(I),I=1,3),(IBC(N,I),I=1,3),KOD(N)
  181 FORMAT(4X,I3,F11.2,F10.2,F10.2,T40,4I4,T53,4(F9.2,1X),3I4,T105,I4)
    3 CONTINUE
C
C     TRANSFORM X TO LOCAL COORDS
C
      CALL RIDEL
      TYPE "TRANS NODES"
      DO 10 N=1,NODE
      W(1)=X(N,1)
      W(2)=X(N,2)
      W(3)=X(N,3)
      CALL VGTOL(W,ALPHA,ORIG)
      DO 10 J=I1,I3
   10 X(N,J)=W(J)
      CALL EXEC(3,1)
      RETURN
      END
*****************************************************************************************************************************
C
C     SUBROUTINE TO FIND COORDINATES OF DD ELEMENT OR LUMP.
C
      SUBROUTINE LCOOR(X,I,J,H)
      DIMENSION X(3)
      X(3)=0.
      X(1)=(2*I-1)*H
      X(2)=(2*J-1)*H
      RETURN
      END
*****************************************************************************************************************************
      SUBROUTINE LESET(ML,XEL,DLC)
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  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),KOOM(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,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,RGB,EW,H,HL,
     1  E,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CONS,CK,CD,G,COND,CHAR,THIK,ESEAM,GSEAM,COH,PHI,
     2  IBFSZ,ICHAN,LIM,IDISK,ISUF,ISTAR,BUF(1152),IREAD,IRITE,NELIL,NREC0,KODE,ISSW(15),ISW(15)
C
C
C     SUBROUTINE TO LOAD XEL AND DLC
C
      DIMENSION XEL(3),DLC(3)
      DO 1 I=1,3
      DLC(I)=DLUMC(ML,I)
    1 XEL(I)=XLUMP(ML,I)
```

```
      CALL SYMFIX(SYM,NMULT,XEL,ISYM)
      CALL SYMFIX(FALL,NMULT,DLC,ISYM)
      RETURN
      END
*****************************************************************************************************************
      SUBROUTINE LGEOM
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  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,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
      ATOT=ZERO
      NIL=0
      RATB=0.05
      RATT=0.8
      DO 1 N=1,NLUMP
      NUMEL=LKOD(N)
      ALUMP(N)=ZERO
      DO 10 I=1,3
      DLUMC(N,I)=ZERO
   10 XLUMP(N,I)=ZERO
      DO 1 NE=1,NUMEL
      NEL=LUMPO(N,NE)
    1 ALUMP(N)=ALUMP(N)+AREA(NEL)
      DO 2 N=1,NLUMP
      ATOT=ATOT+ALUMP(N)
      NUMEL=LKOD(N)
      DO 2 NE=1,NUMEL
      NEL=LUMPO(N,NE)
      RAT=AREA(NEL)/ALUMP(N)
      N1=INDEX(NEL,1)
      N2=INDEX(NEL,2)
      N3=INDEX(NEL,3)
      N4=INDEX(NEL,4)
      CALL DCOSV(N1,N2,N3,W,X)
      IF(N4.EQ.NIL)CALL XCENT(N1,N2,N3,XNUM,X)
      IF(N4.NE.NIL)CALL XCENQ(N1,N2,N3,N4,XNUM,X)
      IF(RAT.LT.RATB)TYPE "ERROR 10",N,NEL
      IF(RAT.GT.RATT)TYPE "ERROR 11",N,NEL
      DO 2 I=1,3
      XLUMP(N,I)=XLUMP(N,I)+XNUM(I)*RAT
    2 DLUMC(N,I)=DLUMC(N,I)+W(I)*RAT
      DO 20 NL=1,NLUMP
      SUM=ZERO
      DO 21 J=1,3
   21 SUM=SUM+DLUMC(NL,J)**2
```

```
      IF(SUM.LT.0.9)TYPE "ERROR 12",NL,SUM
   20 CONTINUE
C
C     OUTPUT LUMP GEOM
C
      IF(ISSW(4).EQ.1)RETURN
      WRITE(IRITE,102)ATOT
  102 FORMAT(/" TOTAL AREA =",F10.1,//)
      WRITE(IRITE,100)
  100 FORMAT(/" LUMP"15X"COORDS"18X"AREA"16X"DC",20X"MEDIUM",//)
      DO 4 N=1,NLUMP
    4 WRITE(IRITE,101)N,(XLUMP(N,I),I=1,3),ALUMP(N),(DLUMC(N,I),I=1,3)
     1,LMOD(N)
  101 FORMAT(I4,7(1X,F10.3),3X,I4)
      RETURN
      END
*******************************************************************************************************************************
      SUBROUTINE LOAD
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  MINC(25,25),DZ(25,25),DX(25,25),DY(25,25),DZZT(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  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,TRANDD,FAC,UGNT(11),COORD(11,3),
     9  NOBC,NELT,NLUMP,NN,MM,NLSX,NLSY,ND1ML,ND1MN,ND1M,ISMAL,JSMAL,NUMON,NSLIP,NIL,MODID,NS,
     *  ZERO,ONE,IONE,I1,I2,I3,THIRD,HALF,TWO,NITER,OMEGA,DEL,PI,A,B,C,R,RSQ,EM,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  IBFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELCL,NRLCO,KODE,ISSW(15),ISW(15)
C
C
C     SUBROUTINE TO LOAD AND DUMP BUFFER
C
      IF(IBUF.GT.LIM)CALL DUMPO(IBUF,IDISK,BUF,ISTAR,1)
      DO 1 I=1,3
      DO 1 J=1,3
      IBUF=IBUF+1
      BUF(IBUF)=TT(I,J)
      IBUF=IBUF+1
    1 BUF(IBUF)=UU(I,J)
      RETURN
      END
*******************************************************************************************************************************
C
C     SUBROUTINE TO LOAD BUF FROM TT AND UU
C
      SUBROUTINE LOAD1(N,TEMP,KODK)
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),PRINI(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),QR1G(3),
     7  EMOD(10),DEN(10),PR(10),THETA,DEPTH,RHO,ISYM,NSYM(10),NDIR(3,3),NMULT(3,3),
     8  TRAN4Z,TRAN2I,TRAN2,TRANS,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,CTAR,THIK,ESEAM,GSEAM,COH,PHI,
     2  IBFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NLLLL,NRECO,KODE,ISSW(15),ISW(15)
C
C
C     SUBPROGRAM TO LOAD INFL COEFFICIENTS ON BOUNDARY ELEMENTS.
C
      DIMENSION TEMP(3)
      IF(IBUF.GT.LIM)CALL DUMPO(IBUF,IDISK,BUF,ISTAR,1)
      IF(KODE.NE.8)GO TO 503
      IF(KODE.NE.KODR)GO TO 503
      IBUF=IBUF+2
      BUF(IBUF-I1)=TT(I3,I1)
      BUF(IBUF)=TT(I3,I2)
      RETURN
  503 CONTINUE
      DO 4 I=I1,I3
      DO 4 J=I1,I3
      IF(KODE.LT.7)GO TO 501
      IF(KODE.EQ.8)GO TO 501
      IBUF=IBUF+1
      BUF(IBUF)=TT(I,J)
      IBUF=IBUF+1
      BUF(IBUF)=UU(I,J)
      GO TO 4
  501 IBU=IBC(N,J)
      IF(IBU.NE.1)GO TO 502
      TEMP(I)=TEMP(I)+UU(I,J)*T(N,J)
      IBUF=IBUF+1
      BUF(IBUF)=TT(I,J)
      GO TO 4
  502 TEMP(I)=TEMP(I)+TT(I,J)*U(N,J)
      IBUF=IBUF+1
      BUF(IBUF)=UU(I,J)
    4 CONTINUE
      RETURN
      END

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

C
C     LOAD A VECTOR X FROM LUMP ARRAY XLUMP.
C
      SUBROUTINE LSET(N,X,XLUMP)
      DIMENSION X(3),XLUMP(50,3)
      DO 1 I=1,3
    1 X(I)=XLUMP(N,I)
      RETURN
      END

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

      SUBROUTINE LUMPR
```

```
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),NLUMC(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,NM,NLSX,NLSY,NDIML,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NSLIP,NL,NODOD,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
C     EVALUATE LUMP RIDES AND CLOSURES.
C
  100 FORMAT(" LEAVING LUMPR")
      DO 621 K=1,NLSX
      K1=(K-1)*ISMAL+1
      K2=K*ISMAL
      DO 621 L=1,NLSY
      L1=(L-1)*JSMAL+1
      L2=L*JSMAL
      XL=0.
      YL=0.
      ZL=0.
      DO 620 KK=K1,K2
      DO 620 LL=L1,L2
      XL=XL+DX(KK,LL)
      YL=YL+DY(KK,LL)
  620 ZL=ZL+DZ(KK,LL)
      XLARG(K,L)=XL
      YLARG(K,L)=YL
      ZLARG(K,L)=ZL
  621 CONTINUE
C
      WRITE(10,100)
      RETURN
      END

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

C
C     INITIALIZE AN ARRAY OF DIMENSION 3.
C
      SUBROUTINE NULL(X,VAL)
      DIMENSION X(3),VAL(3)
      X(1)=VAL(1)
      X(2)=VAL(2)
      X(3)=VAL(3)
      RETURN
      END

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

C
C     CALCULATE PRIMITIVE TRACTIONS IN LOCAL COORDINATE SYSTEM.
```

```
C
C     P = PRIM STRESS (NORMMALIZED) IN LOCAL SYSTEM.
C     A = TRANSFORMATION MATRIX.
C     O = ORIGIN OF LOCAL SYSTEM.
C     X = COORDS OF POINT IN LOCAL SYSTEM
C     DC= DC OF POINT IN LOCAL SYSTEM
C     W = TRACTIONS IN LOCAL SYSTEM.
C
      SUBROUTINE PRIMDD(A,P,O,X,DC,W,RHO,ISSW)
      DIMENSION A(3,3),P(3,3),O(3),ISSW(15),DC(3),W(3),X(3)
      CALL VLTOG(X,A,O)
      DEPTH=X(3)*RHO*9.81
      IF(ISSW(8).EQ.1)DEPTH=1.
      DO 1 I=1,3
      W(I)=0
      DO 1 J=1,3
    1 W(I)=W(I)+DEPTH*P(I,J)*DC(J)
      RETURN
      END
C
*****************************************************************************************************************
C
      SUBROUTINE RIDEL
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  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,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,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
      DIMENSION XAXIS(3),YAXIS(3)
C
C     SUBROUTINE TO HANDLE IN-ARD OUTPUT.
C
      MODDD=1
      TYPE "RIDEL"
      READ(IREAD)EW,ESEAM,GSEAM,NELEL,NN,MM,LUM
      IER=1
      IF(NELEL.GT.0)CALL OPEN(1,"ONREEF",3,IER)
      IF(IER.NE.1)TYPE "ONREEF OP ER",IER
      READ(IREAD)ORIG,XAXIS,YAXIS
      IF(NN.LT.30)ISW(12)=1
      DEPTH=ORIG(3)
      IF(NELEL.LE.0)GO TO 91
C  93 FORMAT("    NUMBER OF ON REEF BENCHMARKS = ",I4,/)
      DO 92 N=1,NELEL
      READ(IREAD)I,J
      WRITE(1,105)I,J
  105 FORMAT(1X,2I4)
   92 CONTINUE
   91 CONTINUE
```

```
      ISMAL=LUN
      JSMAL=LUM
      H=EW/2.
      NL=IR/ISMAL
C
      NLSX=NN/ISMAL
      NLSY=MM/JSMAL
      CALL MODDEN(MODDD)
C
C     WRITE OUT VARIOUS CONSTANTS.
C
      WRITE(IRITE,108)ESEAM,GSEAM,V,THIK,EW
      WRITE(IRITE,215)DEPTH
      CONU=-G/(4.*PI*(1.-V)*H)
C
  108 FORMAT(/10X,"YOUNGS MODULUS     =",F10.2,
     1/10X,"SHEAR MODULUS      = ",F10.2,
     1/10X,"POISSONS RATIO     = ",F10.2,
     1/10X,"STOPE THICKNESS     = ",F10.2,
     1/10X,"SQUARE WIDTH       = ",F10.2)
C 130 FORMAT(1X,3F10.4)
C 131 FORMAT(1X,3F10.1)
  215 FORMAT(/1X,"DEPTH TO ORIGIN OF MINE COORD. SYSTEM IS ",F8.2,"M
     1. ",/)
C 221 FORMAT(/" NORMAL MODULUS OF SEAM ELEMENTS =",F10.1,//
C     1" SHEAR  MODULUS OF SEAM ELEMENTS =",F10.1,/)
C
CC    CALCULATE THE TRANSFORMATION MATRIX FOR CHANGE OF AXES.
C
C
CC    WRITE DATA FOR PRIMITIVE STRESS.
C
      WRITE(IRITE,721)
      WRITE(IRITE,132)((PRIM(I,J),J=1,3),I=1,3)
      CALL DCOSA(ORIG,XAXIS,YAXIS,ALPHA)
      CALL TGTOL(PRIM,ALPHA)
      WRITE(IRITE,722)
      WRITE(IRITE,132)((PRIM(I,J),J=1,3),I=1,3)
      WRITE(IRITE,723)
      WRITE(IRITE,132)((ALPHA(I,J),J=1,3),I=1,3)
C
      WRITE(IRITE,724)ORIG
  724 FORMAT(/" ORIGIN OF DISPLACMENT DISCONTINUITY GRID IS",3F10.2)
      WRITE(IRITE,725)XAXIS,YAXIS
  725 FORMAT(" X- AND Y-AXIS MARKERS FOR LOCAL COORD SYSTEM ARE"2(/1X,3F10.2))
C
CC    SET UP BOUNDARY CONDITIONS.
C
  132 FORMAT(3F8.4,2X)
  721 FORMAT(/10X,"  PRIMITIVE STRESS TENSOR IS GIVEN BY:",/)
  722 FORMAT(//10X,"AFTER TRANSFORMATION TO MINE COORDINATES,THE STRESS
     1 TENSOR IS:",/)
  723 FORMAT(/10X,"THE TRANSFORMATION MATRIX IS  ",/)
C
C     DO ELT-ELT COEFFS.
C
      IADIM=9
      DO 10 I=1,IADIM
      PPI=FLOAT(2*I-1)
      PNI=FLOA((2*I-3)
```

```
      PPI2=PPI*PPI
      PMI2=PMI*PMI
      DO 10 J=1,IADIM
      RPI=FLOAT(2*J-1)
      RMI=FLOAT(2*J-3)
      RPI2=RPI*RPI
      RMI2=RMI*RMI
      Q1=SQRT(PPI2+RPI2)
      Q2=SQRT(PMI2+RMI2)
      Q3=SQRT(PPI2+RMI2)
      Q4=SQRT(PMI2+RPI2)
C
CC    EVALUATE FUNCTIONS F AND FF.
C
      FA=COND*Q1/(PPI*RPI)
      FB=COND*Q2/(PMI*RMI)
      FC=COND*Q3/(PPI*RMI)
      FD=COND*Q4/(PMI*RPI)
      GFA=COND/Q1
      GFB=COND/Q2
      GFC=COND/Q3
      GFD=COND/Q4
C
C     EVALUATE INFLUENCE COEFFICIENTS.
C
      AK(I,J)=FA+FB-FC-FD
      BK(I,J)=(PPI*GFA/RPI+PMI*GFB/RMI-PPI*GFC/RMI-PMI*GFD/RPI)*V
      DK(I,J)=(GFA+GFB-GFC-GFD)*V
      BMA(I,J)=AK(I,J)-BK(I,J)
      BK(I,J)=AK(I,J)-BK(I,J)*(1.-2.*V)/V
10    CONTINUE
      RETURN
      END

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

      SUBROUTINE SIGBEN(TRAC,DIS,XX,XC,W1,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(30,3),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,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,CFAR,THIK,ESEAM,GSEAM,COH,PHI,
     2  IBFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELEL,NRECO,KODE,ISSW(15),ISW(15)
      DIMENSION XX(3),XEL(3),XC(3),DIS(3),TRAC(3),DCO(3),STRES(3,3)
      EQUIVALENCE (STRES,TT)
      R3=R*RSQ
      WCK=-W1*CK/RSQ
      WCR=W1*CONS/R
      WS=W1*CD/R3
      WD=-WCK
      DRDN=ZERO
```

```
C
      DO 5 L=1,3
      RCOM(L)=(XC(L)-XX(L))/R
    5 DRDN=DRDN+RCOM(L)*DCO(L)
      DO=DRDN*3.
      DO 6 I=1,3
      RCI=RCOM(I)
      RIV3=RCI*V3
      RIV2=RCI*V2
      DCI=DCO(I)
      RI3=RCI*3.
      DO 6 J=1,3
      RCJ=RCOM(J)
      DCJ=DCO(J)
      DIJ=DELTA(I,J)
      RIRJ=RCI*RCJ
      DV6=DIJ*V6
      RNV7=V7*(DCI*RCJ+DCJ*RCI)
      TFUNC=WCK*(DRDN*(DIJ+RIV3*RCJ)+DCI*RCJ-DCJ*RCI)*DIS(J)
      W(I)=W(I)+WCR*(DIJ*V4+RIV2*RCJ)*TRAC(J)-TFUNC
      IF(J.LT.I)GO TO 6
      SUM=ZERO
      DO 7 K=1,3
      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
      DKIJ=WD*(DUM-DIJRK+V3*RIJK)*TRAC(K)
    7 SUM=SUM-WS*(D2*(DIJRK+V9*DUM-V8*RIJK)+RNV7*RCK+3.*RIRJ*DCK
     1+DCJ*DIK+DCI*DJK-V6*DCK*DIJ)*DIS(K)+DKIJ
      STRES(I,J)=STRES(I,J)-SUM
C                          -SUM FOR COMP + CONVENTION.
    6 CONTINUE
      RETURN
      END
C
*****************************************************************************************************************************
C
      SUBROUTINE SIGDD(TRAC,DIS,XX,XC,W1)
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),BK(9,9),DK(9,9),BMA(9,9),ZLARG(9,9),XLARG(9,9),YLARG(9,9),
     5  PRIM(3,3),PRIMI(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,MN,MM,NLSX,NLSY,NDIML,NDIMN,NDIM,ISMAL,JSMAL,NUMOR,NSLIP,NIL,MODDD,NS,
     *  ZERO,ONE,TONE,11,12,13,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,CHAR,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     SUMMATION FOR BENCHMARK VALUES.
C
      DIMENSION DIS(3),XX(3),XC(3),STRES(3,3)
```

```
      EQUIVALENCE (STRES,TT)
      R3=R*RSQ
      WCK=+W1*CK/RSQ
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*SUMI+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),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),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  CMOD(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,CONDC,CFAR,THIK,ESEAM,GSEAM,COIL,PHI,
     2  IBFSZ,ICHAN,LIM,IDISK,IBUF,ISTAR,BUF(1152),IREAD,IRITE,NELEL,NRECO,KODE,ISSW(15),ISW(15)
C
      WRITE(IRITE,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,TRANUL,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
```

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.