APPENDIX-II

(A) FEM PROGRAMME LISTING

```
$TITLE: MEHTA
$large
      PROGRAM MEHTA
       INTEGER *2 LNAM1,LNAM2,LUN5,LUN7,LUN8,LUN9,LUN10,LUN11
      CHARACTER *4 TITLE(12)
      COMMON /CONTROL/ NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,NGAUS,NPRO
     1P,NMATS,NEVAB,ICASE,NCASE,ITEMP,IPROB,NPROB
      COMMON/LGDATA/ COORD(3500,2),PROPS(15,5),PRESC(1750,2),ASDIS(750,1),
     1ELOAD(525,16),ESTIF(3,4725),NOFIX(1750),IFPRE(1750,2),LNODS(525,8)
     2,MATNO(525)
      COMMON/WORK/ELCOD(3,8),SHAPE(8),DERIV(2,8),DMATX(3,3),CARTD(2,8),
     1DEPAT(3,16),BMATX(3,16),SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9)
     2,NEROF 50)
        OPEN(UNIT=5,FILE='NHCDC.DAT',STATUS='OLD',
     1  FORM='FORMATTED',
        OPEN(UNIT=7,FILE='NHCDC.OUT',STATUS='NEW',FORM='FORMATTED')
        OPEN(UNIT=8,FILE='NHCDC.TM1',STATUS='NEW',FORM='UNFORMATTED')
        OPEN(UNIT=9,FILE='NHCDC.TM2',STATUS='NEW',FORM='UNFORMATTED')
        OPEN(UNIT=10,FILE='NHCDC.TM3',STATUS='NEW',FORM='UNFORMATTED')
        OPEN(UNIT=11,FILE='NHCDC.TM4',STATUS='NEW',FORM='UNFORMATTED')
      READ(5,900)NPROB
  900 FORMAT(I5)
      WRITE(7,905) NPROB
  905 FORMAT(1H0,5X,23HTOTAL NO. OF PROBLEMS =,I5)
      DO 20 IPROB=1,NPROB
      REWIND 8
      REWIND 9
      REWIND 10
      REWIND 11
      READ(5,910) TITLE
  910 FORMAT(12A6)
      WRITE(7,915) IPROB,TITLE
  915 FORMAT(//// ,6X,12HPROBLEM NO. ,I3,10X,12A6)
C
C***  CALL THE SUBROUTINE WHICH READS MOST OF
C     THE PROBLEM DATA
C
      CALL INPUT
C
C***  NEXT CREATE THE ELEMENT STIFFNESS FILE.
C
      CALL STIFPS
      DO 10 ICASE=1,NCASE
C
C***  COMPUTE LOADS, AFTER READING THE RELEVANT
C     EXTRA DATA
C
      CALL LOADPS
C
C***  MERGE AND SOLVE THE RESULTING EQUATIONS
C     BY THE FRONTAL SOLVER
C
      CALL FRONT
```

```
 C
 C***  COMPUTE THE STRESSES IN ALL THE ELEMENTS.

C
      CALL STREES
   10 CONTINUE
   20 CONTINUE
C       CALL CLOS$A(LUN5)
C       CALL CLOS$A(LUN7)
C       CALL CLOS$A(LUN8)
C       CALL CLOS$A(LUN9)
C       CALL CLOS$A(LUN10)
C       CALL CLOS$A(LUN11)
        CLOSE(UNIT=5,STATUS='KEEP')
        CLOSE(UNIT=7,STATUS='KEEP')
        CLOSE(UNIT=8,STATUS='KEEP')
        CLOSE(UNIT=9,STATUS='KEEP')
        CLOSE(UNIT=10,STATUS='KEEP')
        CLOSE(UNIT=11,STATUS='KEEP')
      STOP
      END
      SUBROUTINE INPUT
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,NGAUS,NPRO
     1P,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROP,NPROB
      COMMON/LGDATA/COORD(350,2),PROPS(25,8),PRESC(175,2),ASDIS(7000),
     1ELOAD(525,16),STRIN(3,4725),NOFIX(175),IFPRE(175,2),LNODS(525,8),
     2,MATNO(525)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(3,3),CARTD(2,8),
     1BMATX(3,16),DBMAT(3,16),SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9,
     2,NERGH(50)
C
C***  READ THE FIRST DATA CARD, AND ECHO IT
C     IMMEDIATELY.
C
      READ(5,900) NPOIN,NELEM,NVFIX,NCASE,NTYPE,
     1NNODE,NDOFN,NMATS,NPROP,NGAUS,NDIME,NSTRE
  900 FORMAT(12I5)
      NEVAB=NDOFN*NNODE
      WRITE(7,905) NPOIN,NELEM,NVFIX,NCASE,NTYPE,
     1NNODE,NDOFN,NMATS,NPROP,NGAUS,NDIME,
     1NSTRE,NEVAB
  905 FORMAT(//8H NPOIN =,I4,4X,8H NELEM =,I4,
     14X,8H NVFIX =,I4,4X,8H NCASE =,I4,4X,
     18H NTYPE =,I4,4X,8H NNODE =,I4,4X,
     18H NDOFN =,I4,/ 8H NMATS =,I4,4X,
     18H NPROP =,I4,4X,8H NGAUS =,I4,4X,
     18H NDIME =,I4,4X,8H NSTRE =,I4,4X,
     18H NEVAB =,I4)
      CALL CHECK1
C
C***  READ THE ELEMENT NODAL CONNECTIONS, AND
C     THE PROPERTY NUMBERS.
C
      WRITE(7,910)
  910 FORMAT(//8H ELEMENT,3X,8HPROPERTY,6X,
     112HNODE NUMBERS)
      DO 10 IELEM=1,NELEM
      READ(5,900) NUMEL,MATNO(NUMEL),
```

```
     1(LNODS(NUMEL,INODE) INODE=1,NNODE)
    1) WRITE(7,913) NUMEL,MATNO(NUMEL),
     1(LNODS(NUMEL,INODE),INODE=1,NNODE)
  913 FORMAT(1X,I5,I9,5X,8I5)
C
C###  ZERO ALL OUT NODAL COORDINATES,PRIOR
C     TO READING SOME OF THEM,
C
      DO 20 IPOIN=1,NPOIN
      DO 20 IDIME=1,NDIME
   20 COORD(IPOIN,IDIME)=0.0
C
C###  READ SOME NODAL COORDINATES. FINISHING
C     WITH THE LAST NODE OF ALL
C
      WRITE(7,920)
  920 FORMAT(//24H NODAL POINT COORDINATES)
      WRITE(7,925)
  925 FORMAT(5H NODE,7X,1HX,9X,1HY)
   30 READ(5,930) IPOIN,(COORD(IPOIN,IDIME),
     1IDIME=1,NDIME)
  930 FORMAT(I5,3F10.0)
      IF(IPOIN.NE.NPOIN) GO TO 30
C
C###  INTERPOLATE THE COORDINATES OF MID SIDE NODES
C
      IF(NDIME.EQ.1) GO TO 40
      CALL MODEXY
   40 CONTINUE
      DO 50 IPOIN=1,NPOIN
   50 WRITE(7,935) IPOIN,(COORD(IPOIN,IDIME),
     1IDIME=1,NDIME)
  935 FORMAT(1X,I5,3F10.3)
C
C###  READ THE FIXED VALUES
C
      WRITE(7,940)
  940 FORMAT(//17H RESTRAINED NODES.
      WRITE(7,945)
  945 FORMAT(5H NODE,1X,4HCODE,6X,
     1'PRESCRIBED VALUES')
      IF(NDOFN.NE.2) GO TO 70
      DO 60 IVFIX=1,NVFIX
      READ(5,950) NOFIX(IVFIX),(IFPRE(IVFIX,
     1IDOFN),IDOFN=1,NDOFN),(PRESC(IVFIX,IDOFN
     1),IDOFN=1,NDOFN)
      WRITE(7,950) NOFIX(IVFIX),(IFPRE(IVFIX,
     1IDOFN),IDOFN=1,NDOFN),(PRESC(IVFIX,IDOFN
     1),IDOFN=1,NDOFN)
  950 FORMAT(1X,I4,7X,2I1,2F10.6)
      GO TO 90
   70 DO 80 IVFIX=1,NVFIX
      READ(5,955) NOFIX(IVFIX),(IFPRE(IVFIX,
     1IDOFN),IDOFN=1,NDOFN),(PRESC(IVFIX,
     1IDOFN),IDOFN=1,NDOFN)
```

```
   8  WRITE(7,955) NOFIX,(NFIX(I),IFFRE(I,IFIX),
     1(IFFRE(I,IFIX),IDOFN=1,NDOFN)
  955 FORMAT(1X,I4,2X,I1,3F10.6)
    9  CONTINUE
C
C***  READ THE AVAILABLE SELECTION OF ELEMENT
C     PROPERTIES.
C
      WRITE(7,960)
  960 FORMAT(//20H MATERIAL PROPERTIES)
      WRITE(7,965)
  965 FORMAT(7H NUMBER,7X,10HPROPERTIES)
      DO 100 IMATS=1,NMATS
      READ(5,6001)NUMAT,(PROPS(NUMAT,IPROP),
     1IPROP=1,NPROP)
 6001 FORMAT(I5,5F10.2)
  100 WRITE(7,970) NUMAT,(PROPS(NUMAT,IPROP),IPROP=1,NPROP)
  970 FORMAT(1X,I5,7X,5F14.6)
C
C***  SET UP GAUSSIAN INTEGRATION CONSTANTS
C
      CALL GAUSSQ
      CALL CHECK2
      RETURN
      END
      SUBROUTINE NODEXY
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,NGAUS,NPRO
     1P,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IFROP,NPROB
      COMMON/LGDATA/COORD(3500,2),PROPS(15,5),PRESC(1750,2),ABDIS(7000,
     1ELOAD(525,16),STRIN(3,4725),NOFIX(1750),IFFRE(1750,2),LNODS(525,8)
     2,MATNO(525)
      COMMON/WORK/ELCOD(3,8),SHAPE(8),DERIV(2,8),DMATX(3,3),CARTD(2,8),
     1DBMAT(3,16),BMATX(3,16),SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(3,9)
     2,NEROR(50)
C
C***  LOOP OVER EACH ELEMENT
C
      DO 30 IELEM=1,NELEM
C
C***  LOOP OVER EACH ELEMENT EDGE
C
      DO 20 INODE=1,NNODE,2
C
C***  COMPUTE THE NODE NUMBER OF THE FIRST NODE
C
      NODST=LNODS(IELEM,INODE)
      IGASH=(NODE+2
      IF(IGASH.GT.NNODE) IGASH=1
C***  COMPUTE THE NODE NUMBER OF THE LAST NODE
C
      NODFN=LNODS(IELEM,IGASH)
      NIDFT=INODE+1
C
C***  COMPUTE THE NODE NUMBERS OF THE
```

```
C     INTERMEDIATE NODE
C
      XCOORD=COORD(IELEM,MIDPT)
      TOTAL=ABS(COORD(NODEMO,1)-
     1-ES(COORD(NODML,2))
C
C***  IF THE COORDINATES OF THE INTERMEDIATE
C     NODE ARE WITH ZERO INTERPOLATE BY A
C     STRAIGHT LINE
C
      IF(TOTAL.GT.0.0) GO TO 2
      KOUNT=1
   10 COORD(NODMD,KOUNT)=(COORD(NODST,KOUNT)+
     1COORD(NODFN,KOUNT))/2.0
      KOUNT=KOUNT+1
      IF(KOUNT.EQ.2) GO TO 10
   20 CONTINUE
   30 CONTINUE
      RETURN
      END
      SUBROUTINE GAUSSO
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,NGAUS,NPRO
     1P,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,NPROB
      COMMON/LGDATA/COORD(3500,2),PROPS(15,5),FREED(1750,2),ASDIS(7000),
     1ELOAD(525,16),STRIN(3,4725),NOFIX(1750),IFREE(1750,2),LNODS(525,8)
     2,MATNO(525)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),JMATX(2,2),CARTD(2,8),
     1QBMAT(3,16),BMATX(3,16),SMATX(3,16,9),POSGP(9),WEIGP(9),GPCOD(2,9)
     2,NEROR(50)
      IF(NGAUS.GT.2) GO TO 10
      POSGP(1)=-0.57735026918962G
      WEIGP(1)=1.0
      GO TO 20
   10 POSGP(1)=-0.77459666924143
      POSGP(2)=0.0
      WEIGP(1)=0.555555555555556
      WEIGP(2)=0.888888888888889
   20 KGAUS=NGAUS/2
      DO 30 IGASH=1,KGAUS
      JGASH=NGAUS+1-IGASH
      POSGP(JGASH)=-POSGP(IGASH)
      WEIGP(JGASH)=WEIGP(IGASH)
   30 CONTINUE
      RETURN
      END
      SUBROUTINE SHAPE2
      DIMENSION DERIV(2,16)
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,NGAUS,NPRO
     1P,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,NPROB
      COMMON/LGDATA/COORD(3500,2),PROPS(15,5),FREED(1750,2),ASDIS(7000),
     1ELOAD(525,16),STRIN(3,4725),NOFIX(1750),IFREE(1750,2),LNODS(525,8)
     2,MATNO(525)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),JMATX(2,2),CARTD(2,8),
     1QBMAT(3,16),BMATX(3,16),SMATX(3,16,9),POSGP(9),WEIGP(9),GPCOD(2,9)
     2,NEROR(50)
```

```
C
C***  LOOP OVER EACH ELEMENT  --
C
      DO 70 IELEM=1,NELEM
      LPROP=MATNO(IELEM)
C
C***  EVALUATE THE COORDINATES OF THE ELEMENT
C     NODAL POINTS
C
      DO 10 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      DO 10 IDIME=1,NDIME
   10 ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
C
C***  EVALUATE THE D-MATRIX
C
      CALL MODPS(LPROP)
      THICK=PROPS(LPROP,1)
C
C***  INITIALIZE THE ELEMENT STIFFNESS MATRIX
C
      DO 20 IEVAB=1,NEVAB
      DO 20 JEVAB=1,NEVAB
   20 ESTIF(IEVAB,JEVAB)=0.0
      KGASP=0
C
C***  ENTER LOOPS FOR AREA NUMERICAL INTEGRATION
C
      DO 50 IGAUS=1,NGAUS
      DO 50 JGAUS=1,NGAUS
      KGASP=KGASP+1
      EXISP=POSGP(IGAUS)
      ETASP=POSGP(JGAUS)
C
C***  EVALUATE THE SHAPE FUNCTIONS,ELEMENTAL
C     VOLUME,ETC
C
      CALL SHP2(EXISP,ETASP)
      CALL JACOB2(IELEM,DJACB,KGASP)
      DVOLU=DJACB*WEISP(IGAUS)*WEIGP(JGAUS)
      IF(THICK.NE.0.0) DVOLU=DVOLU*THICK
C
C***  EVALUATE THE B AND DB MATRICES
C
      CALL BMATPS
      CALL DBE
C
C***  CALCULATE THE ELEMENT STIFFNESSES
C
      DO 30 IEVAB=1,NEVAB
      DO 30 JEVAB=IEVAB,NEVAB
      DO 30 ISTRE=1,NSTRE
   30 ESTIF(IEVAB,JEVAB)=ESTIF(IEVAB,JEVAB)+
     1BMATX(ISTRE,IEVAB)*DBMAT(ISTRE,
     1JEVAB)*DVOLU
```

```
C
C***  STORE THE COMPONENTS OF THE [illegible]
C     THE ELEMENT
C
      DO 40 ISTRE=1,NSTRE
      DO 40 IEVAB=1,NEVAB
   40 SMATX(ISTRE,IEVAB,IGAUS)=DEMAT[illegible]
   50 CONTINUE
C
C***  CONSTRUCT THE LOWER TRIANGLE OF THE
C     STIFFNESS MATRIX
C
      DO 60 IEVAB=1,NEVAB
      DO 60 JEVAB=1,NEVAB
   60 ESTIF(JEVAB,IEVAB)=ESTIF(IEVAB,JEVAB)
C
C***  STORE THE STIFFNESS MATRIX,STRESS MATRIX
C     AND SAMPLING POINT COORDINATES FOR EACH
C     ELEMENT ON DISC FILE
C
      WRITE(8) ESTIF
        WRITE(10) SMATX,GPCOD
   70 CONTINUE
      RETURN
      END
      SUBROUTINE MODPS(LPROP)
      COMMON/CONTRO/NFOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,NGAUS,NPRO
     1P,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IFROM,NPROB
      COMMON/LGDATA/COORD(3500,2),PROPS(15,5),PRESC(1750,2),ASDIS(7000),
     1ELOAD(525,16),STRIN(3,4725),NOFIX(1750),IFPRE(1750,2),LNODS(525,8)
     2,MATNO(525)
      COMMON/WORK/ELCOD(3,8),SHAPE(8),DERIV(2,8),DMATX(3,3),CARTD(2,8),
     1DBMAT(3,16),BMATX(3,16),SMATX(3,16,9),FORGP(3),WEIGP(3),GPCOD(3,9)
     2,NEROR(50)
      YOUNG=PROPS(LPROP,1)
      POISS=PROPS(LPROP,2)
      DO 10 ISTRE=1,NSTRE
      DO 10 JSTRE=1,NSTRE
      DMATX(ISTRE,JSTRE)=0.0
   10 CONTINUE
      IF(NTYPE.NE.1) GO TO 20
C
C***  D MATRIX FOR PLANE STRESS CASE
C
      CONST=YOUNG/(1.0-POISS*POISS)
      DMATX(1,1)=CONST
      DMATX(2,2)=CONST
      DMATX(1,2)=CONST*POISS
      DMATX(2,1)=CONST*POISS
      DMATX(3,3)=(1.0-POISS)*CONST/2.0
      GO TO 30
   20 IF(NTYPE.NE.2) GO TO 30
C
C***  D MATRIX FOR PLANE STRAIN CASE
C
```

```
      CONST=YOUNG/(1.0-POISS)/((1.0+POISS)*(1.0-2.0*POISS))
      DMATX(1,1)=CONST
      DMATX(2,2)=CONST
      DMATX(1,2)=CONST*POISS/(1.0-POISS)
      DMATX(2,1)=CONST*POISS/(1.0-POISS)
      DMATX(3,3)=CONST*(1.0-2.0*POISS)/(2.0*(1.0-POISS))
   30 CONTINUE
      RETURN
      END
      SUBROUTINE SFR2(S,T)
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,NGAUS,NPRO
     1P,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEM,IPROB,NPROB
      COMMON/LODAT4/COORD(350,2),PROPS(15,5),PRESC(175,2),ASDIS(700),
     1ELOAD(525,16),STRIN(3,4725),NOFIX(175),IFPRE(175,2),LNODS(525,8)
     2,MATNO(525)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(3,3),CARTD(2,8),
     1DBMAT(3,16),BMATX(3,16),SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9)
     2,NEROR(50)
C
C*** CALCULATES SHAPE FUNCTIONS AND THEIR
C    DERIVATIVES FOR 2D ELEMENTS
C
      S2=S*2.0
      T2=T*2.0
      SS=S*S
      TT=T*T
      ST=S*T
      SST=S*S*T
      STT=S*T*T
      ST2=S*T*2.0
C
C*** SHAPE FUNCTIONS
C
      SHAPE(1)=(-1.0+ST+SS+TT-SST-STT)/4.0
      SHAPE(2)=(1.0-T-SS+SST)/2.0
      SHAPE(3)=(-1.0-ST+SS+TT-SST+STT)/4.0
      SHAPE(4)=(1.0+S-TT-STT)/2.0
      SHAPE(5)=(-1.0+ST+SS+TT+SST+STT)/4.0
      SHAPE(6)=(1.0+T-SS-SST)/2.0
      SHAPE(7)=(-1.0-ST+SS+TT+SST-STT)/4.0
      SHAPE(8)=(1.0-S-TT+STT)/2.0
C
C*** SHAPE FUNCTION DERIVATIVES
C
      DERIV(1,1)=(T+S2-ST2-TT)/4.0
      DERIV(1,2)=-S+ST
      DERIV(1,3)=(-T+S2-ST2+TT)/4.0
      DERIV(1,4)=(1.0-TT)/2.0
      DERIV(1,5)=(T+S2+ST2+TT)/4.0
      DERIV(1,6)=-S-ST
      DERIV(1,7)=(-T+S2+ST2-TT)/4.0
      DERIV(1,8)=(-1.0+TT)/2.0
      DERIV(2,1)=(S+T2-SS-ST2)/4.0
      DERIV(2,2)=(-1.0+SS)/2.0
      DERIV(2,3)=(-S+T2-SS+ST2)/4.0
```

```
      DERIV(2,4)=-(1-S)
      DERIV(2,5)=(S-(2.0*S-S [illegible]
      DERIV(2,6)=(1.0-S2)/[illegible]
      DERIV(2,7)=(-[illegible]
      DERIV(2,8)=-(1+ST
      RETURN
      END
      SUBROUTINE JACOB2(I[illegible],DJACB,KGASP)
      DIMENSION XJACM(2,2),XJACI(2,2)
      COMMON/CONTRO/NPOIN,NELEM,NNODE,[illegible]
     1P,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IP[illegible]DE,NFNOE
      COMMON/LGDATA/COORD(3500,2),PROPS(1[illegible],5),PRESC(175[illegible]
     1ELOAD(525,16),STRIN(3,4725),NOFIX(175[illegible]),IFPRE(175[illegible]
     2,MATNO(525)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(3,3)[illegible]
     1DBMAT(3,16),BMATX(3,16),SMATX(3,16,9),POSGP(3),WEIGP(3),[illegible]
     2,NERSP(50)
C
C*** CALCULATES COORDINATES OF GAUSS POINTS
C     AND THE JACOBIAN MATRIX AND ITS DETERMINANT
C     AND THE INVERSE FOR 2D ELEMENTS
C
C
C*** CALCULATE COORDINATES OF SAMPLING POINT
C
      DO 10 IDIME=1,NDIME
      GPCOD(IDIME,KGASP)=0.0
      DO 10 INODE=1,NNODE
      GPCOD(IDIME,KGASP)=GPCOD(IDIME,KGASP)+
     1ELCOD(IDIME,INODE)*SHAPE(INODE)
   10 CONTINUE
C
C*** CREATE JACOBIAN MATRIX XJACM
C
      DO 20 IDIME=1,NDIME
      DO 20 JDIME=1,NDIME
      XJACM(IDIME,JDIME)=0.0
      DO 20 INODE=1,NNODE
      XJACM(IDIME,JDIME)=XJACM(IDIME,JDIME)+
     1DERIV(IDIME,INODE)*ELCOD(JDIME,INODE)
   20 CONTINUE
C
C*** CALCULATE DETERMINANT AND INVERSE OF
C     JACOBIAN MATRIX
C
      DJACB=XJACM(1,1)*XJACM(2,2)-XJACM(1,2)*
     1XJACM(2,1)
      IF(DJACB.GT.0.0) GO TO 30
      WRITE(7,900) IELEM
      STOP
   30 XJACI(1,1)=XJACM(2,2)/DJACB
      XJACI(2,2)=XJACM(1,1)/DJACB
      XJACI(1,2)=-XJACM(1,2)/DJACB
      XJACI(2,1)=-XJACM(2,1)/DJACB
C
```

```
C*** CALCULATE CARTESIAN DERIVATIVES
C
      DO 40 IDIME=1,NDIME
      DO 40 INODE=1,NNODE
      CARTD(IDIME,INODE)=0.0
      DO 40 JDIME=1,NDIME
      CARTD(IDIME,INODE)=CARTD(IDIME,INODE)+
     1XJACI(IDIME,JDIME)*DERIV(JDIME,INODE)
   40 CONTINUE
  900 FORMAT(//,24HPROGRAM HALTED IN JACOB2,
     1/,11X,22H ZERO OR NEGATIVE AREA,/,
     110X,16H ELEMENT NUMBER ,I5)
      RETURN
      END
      SUBROUTINE BMATPS
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME [illegible]
     1P,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,NPROB
      COMMON/LGDATA/COORD(350,2),PROPS(15,5),PRESC(175,2) [illegible]
     1ELOAD(525,16),STRIN(3,4725),NOFIX(175),IFPRE(175) [illegible]
     2,MATNO(525)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8) [illegible]
     1DBMAT(3,16),BMATX(3,16),BMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9
     2,NEROR(5)
      NGASH=0
      DO 10 INODE=1,NNODE
      MGASH=NGASH+1
      NGASH=MGASH+1
      BMATX(1,MGASH)=CARTD(1,INODE)
      BMATX(1,NGASH)=0.
      BMATX(2,MGASH)=0.0
      BMATX(2,NGASH)=CARTD(2,INODE)
      BMATX(3,MGASH)=CARTD(2,INODE)
      BMATX(3,NGASH)=CARTD(1,INODE)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE DBE
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE [illegible]
     1P,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,NPROB
      COMMON/LGDATA/COORD(350,2),PROPS(15,5),PRESC(175,2) [illegible]
     1ELOAD(525,16),STRIN(3,4725),NOFIX(175),IFPRE(175) [illegible]
     2,MATNO(525)
      COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8) [illegible]
     1DBMAT(3,16),BMATX(3,16),BMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD [illegible]
     2,NEROR(5)
C
C*** CALCULATES D X B
C
      DO 10 ISTRE=1,NSTRE
      DO 10 IEVAB=1,NEVAB
      DBMAT(ISTRE,IEVAB)=0.0
      DO 10 JSTRE=1,NSTRE
      DBMAT(ISTRE,IEVAB)=DBMAT(ISTRE,IEVAB)+
     1DMATX(ISTRE,JSTRE)*BMATX(JSTRE,IEVAB)
   10 CONTINUE
```

```
      RETURN
      END
      SUBROUTINE LOADPS
       CHARACTER *4 TITLE
      DIMENSION TITLE(12),POINT(2),PRESS(3,2),[illegible]
     1,STRAN(3),STRES(3),NOPRS(3)
       COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,ISTAC,NTYPE,NGAUS,NFRO
     1F,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IFROE,NPROB
       COMMON/LSDATA/COORD(3500,2),PROPS(15,5),PRESC(1750,2),[illegible]
     1ELOAD(525,16),STRIN(3,4725),NOFIX(1750),IFPRE(1750,2),LNODE(525,8)
     2,MATNO(525)
       COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),[illegible]
     1DBMAT(3,16),BMATX(3,16),SMATX(3,16,9),POSGP(3),WEIGP(3),[illegible]
     2,NEROF(50)
       DO 10 IELEM=1,NELEM
       DO 10 IEVAB=1,NEVAB
   10 ELOAD(IELEM,IEVAB)=0.0
      READ(5,900) TITLE
  900 FORMAT(12A6)
      WRITE(7,905) TITLE,ICASE
  905 FORMAT(1H0,12A6,5X,12H LOAD CASE =,I3)
C
C***  READ DATA CONTROLLING LOADING TYPES
C     TO BE INPUT
C
      READ(5,910) IPLOD,IGRAV,IEDGE,ITEMP
      WRITE(7,910) IPLOD,IGRAV,IEDGE,ITEMP
  910 FORMAT(4I5)
C
C***  READ NODAL POINT LOADS
C
      IF(IPLOD.EQ.0) GO TO 500
   20 READ(5,915) LODPT,(POINT(IDOFN),IDOFN=
     11,NDOFN)
      WRITE(7,915) LODPT,(POINT(IDOFN),IDOFN=1,NDOFN)
  915 FORMAT(I5,2F10.3)
C
C***  ASSOCIATE THE NODAL POINT LOADS WITH
C     AN ELEMENT
C
      DO 30 IELEM=1,NELEM
      DO 30 INODE=1,NNODE
      NLOCA=LNODE(IELEM,INODE)
      IF(LODPT.EQ.NLOCA) GO TO 40
   30 CONTINUE
   40 DO 50 IDOFN=1,NDOFN
      NGASH=(INODE-1)*NDOFN+IDOFN
   50 ELOAD(IELEM,NGASH)=POINT(IDOFN)
      IF(LODPT.LT.NPOIN) GO TO 20
  500 CONTINUE
      IF(IGRAV.EQ.0) GO TO [illegible]
C
C***  GRAVITY LOADING SECTION
C
C
```

```
C***  READ GRAVITY ANGLE AND GRAVITATIONAL
C     CONSTANT
C
      READ(5,920) THETA,GRAV
  920 FORMAT(2F10.3)
      WRITE(7,925) THETA,GRAV
  925 FORMAT(1HC,16H GRAVITY ANGLE =,F10.3,
     119H GRAVITY CONSTANT =,F10.3)
      THETA=THETA/57.29577951
C
C***  LOOP OVER ELEMENTS
C
      DO 90 IELEM=1,NELEM
C
C***  SET UP PRELIMINARY CONSTANTS
C
      LPROP=MATNO(IELEM)
      THICK=PROPS(LPROP,3)
      DENSE=PROPS(LPROP,4)
      IF(DENSE.EQ.0.0) GO TO 90
      GXCOM=DENSE*GRAVITY*SIN(THETA)
      GYCOM=-DENSE*GRAVITY*COS(THETA)
C
C***  COMPUTE COORDINATES OF THE ELEMENT
C     NODAL POINTS
C
      DO 60 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      DO 60 IDIME=1,NDIME
   60 ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
C
C***  ENTER LOOPS FOR AREA NUMERICAL INTEGRATION
C
      DO 80 IGAUS=1,NGAUS
      DO 80 JGAUS=1,NGAUS
      EXISP=POSGP(IGAUS)
      ETASP=POSGP(JGAUS)
C***  COMPUTE THE SHAPE FUNCTIONS AT THE SAMPLING
C     POINTS AND ELEMENTAL VOLUME
C
      CALL SFR2(EXISP,ETASP)
      KGASP=1
      CALL JACOBC(IELEM,DJACB,KGASP)
      DVOLU=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
      IF(THICK.NE.0.0) DVOLU=DVOLU*THICK
C
C***  CALCULATE LOADS AND ASSOCIATE WITH
C     ELEMENT NODAL POINTS
C
      DO 70 INODE=1,NNODE
      NGASH=(INODE-1)*NDOFN+1
      MGASH=(INODE-1)*NDOFN+2
      ELOAD(IELEM,NGASH)=ELOAD(IELEM,NGASH)+GXCOM*SHAPE(INODE)*DVOLU
   70 ELOAD(IELEM,MGASH)=ELOAD(IELEM,MGASH)+
     1GYCOM*SHAPE(INODE)*DVOLU
```

```
   5" CONTINUE
   9; CONTINUE
  6" CONTINUE
      IF(IEDGE.EQ.3' GO TO 7;
C
C***  DISTRIBUTED EDGE LOADS SECTION
C
      READ(5,930' NEDGE
  930 FORMAT(I5'
      WRITE(7,935' NEDGE
  935 FORMAT(1H0,5X,21HNO. OF LOADED EDGES = ,I5
      WRITE(7,940'
  940 FORMAT(1H0,5X,
     1 38HLIST OF LOADED EDGES AND APPLIED LOADS'
      NODEG=3
C
C***  LOOP OVER EACH LOADED EDGE
C
      DO 16' IEDGE=1,NEDGE
C
C***  READ DATA LOCATING THE LOADED EDGE AND
C     APPLIED LOAD
C
      READ(5,945' NEASS,(NOPRS(IODEG),IODEG=
     1I,NODEG)
  945 FORMAT(4I5)
      WRITE(7,950' NEASS,(NOPRS(IODEG),IODEG=
     1I,NODEG)
  950 FORMAT('I10,5X,7I5'
      READ(5,955' ((PRES-'IODEG,IDOFN),IODEG=
     1I,NODEG),IDOFN=1,NDOFN'
      WRITE(7,955' ((PRESC(IODEG,IDOFN),IODEG=
     1I,NODEG),IDOFN=1,NDOFN'
  955 FORMAT(6F10.3'
      ETASP=-1.0
C
C***  CALCULATE THE COORDINATES OF THE NODES
C     OF THE ELEMENT EDGE
C
      DO 100 IODEG=1,NODEG
      LNODE=NOPRS(IODEG'
      DO 100 IDIME=1,NDIME
  100 ELCOD(IDIME,IODEG)=COORD(LNODE,IDIME'
C
C***  ENTER LOOP FOR(INPAR NUMERICAL INTEGRATION
C
      DO 150 IGAUS=1,NGAUS
      EXISP=POSGP(IGAUS'
C
C***  EVALUATE THE SHAPE FUNCTIONS AT THE
C     SAMPLING POINT
C
      CALL SFR2(E'ISF,ETASF'
C
C***  CALCULATE COORDINATES OF THE EQUIVALENT
```

```
C     NODAL LOADS
C
      DO 110 IDOFN=1,NDOFN
      PGASH(IDOFN)=0.0
      DGASH(IDOFN)=0.
      DO 110 IODEG=1,NODEG
      PGASH(IDOFN)=PGASH(IDOFN)+PRESS(IODES,
     1IDOFN)*SHAPE(IODEG)
  110 DGASH(IDOFN)=DGASH(IDOFN)+ELCOD(IDOFN,
     1IODEG)*DERIV(1,IODEG)
      DVOLU=WEIGP*JGAUS
      PXCOM=DGASH(1)*PGASH(2)-DGASH(2)*PGASH(1)
      PYCOM=DGASH(1)*PGASH(1)+DGASH(2)*PGASH(2)
C
C***  ASSOCIATE THE EQUIVALENT NODAL FORCE
C     LOADS WITH AN ELEMENT
C
      DO 120 INODE=1,NNODE
      NLOCA=LNODS(NEASS,INODE)
      IF(NLOCA.EQ.NOPPS(1)) GO TO 130
  120 CONTINUE
  130 JNODE=INODE+NODEG-1
      KOUNT=0
      DO 140 KNODE=INODE,JNODE
      KOUNT=KOUNT+1
      NGASH=(KNODE-1)*NDOFN+1
      MGASH=(KNODE-1)*NDOFN+2.
      IF(KNODE.GT.NNODE) NGASH=1
      IF(KNODE.GT.NNODE) MGASH=2
      ELOAD(NEASS,NGASH)=ELOAD(NEASS,NGASH)+
     1SHAPE(KOUNT)*PXCOM*DVOLU
  140 ELOAD(NEASS,MGASH)=ELOAD(NEASS,MGASH)+
     1SHAPE(KOUNT)*PYCOM*DVOLU
  150 CONTINUE
  160 CONTINUE
  700 CONTINUE
      IF(ITEMP.EQ.0) GO TO 20.
C
C***  THERMAL LOADING SECTION
C
C
C***  INITIALIZE AND INPUT THE NODAL TEMPERATURES
C
      DO 17 IPOIN=1,NPOIN
   17 TEMPE(IPOIN)=0.0
      WRITE(7,960)
  960 FORMAT(1H0,5(
     1/5HPRESCRIBED NODAL TEMPERATURES)
   18 READ(5,965) NODPT,TEMPE(NODPT)
      WRITE(7,965) NODPT,TEMPE(NODPT)
  965 FORMAT(I5,F10.2)
      IF(NODPT.LT.NPOIN) GO TO 18
      NDIME=NDIME+1
      KEAST=0
C
```

```
C***  LOOP OVER EACH ELEMENT
C
      DO 250 IELEM=1,NELEM
      LPROP=MATNO(IELEM)
      DO 200 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
C
C***  IDENTIFY THE COORDINATES AND TEMPERATURE
C     OF EACH ELEMENT NODE POINT
C
      DO 190 IDIME=1,NDIME
  190 ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
  200 ELCOD(NDIME,INODE)=TEMPE(LNODE)
C
C***  SET UP MATERIAL PROPERTIES
C
      CALL MODPS(LPROP)
      YOUNG=PROPS(LPROP,1)
      POISS=PROPS(LPROP,2)
      THICK=PROPS(LPROP,3)
      ALPHA=PROPS(LPROP,5)
C
C***  ENTER LOOPS FOR AREA NUMERICAL INTEGRATION
C
      DO 270 IGAUS=1,NGAUS
      DO 270 JGAUS=1,NGAUS
      IGAST=(IGAUS-1)*4
      EXISP=POSGP(IGAUS)
      ETASP=POSGP(JGAUS)
C
C***  EVALUATE THE SHAPE FUNCTIONS AND TEMPERATURE
C     AT THE SAMPLING POINTS,ELEMENTAL VOLUME
C     AND CARTESIAN DERIVATIVES
C
      CALL SHAPE2(EXISP,ETASP,
      NGASP=1
      CALL JACOB2(IELEM,DJACB,NGASP)
      THERM=0.0
      DO 210 INODE=1,NNODE
  210 THERM=THERM+ELCOD(NDIME,INODE)*SHAPE(INODE,
      DVOLU=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
      IF(THICK.NE.0.0) DVOLU=DVOLU*THICK
C
C***  EVALUATE THE INITIAL THERMAL STRAINS
C
      EIGEN=THERM*ALPHA
      IF(NTYPE.EQ.2) GO TO 220
      STRAN(1)=-EIGEN
      STRAN(2)=-EIGEN
      STRAN(3)=0.0
      GO TO 230
  220 STRAN(1)=-(1.0+POISS)*EIGEN
      STRAN(2)=-(1.0+POISS)*EIGEN
      STRAN(3)=0.0
C
```

```
C***  AND THE CORRESPONDING INITIAL STRESSES
C
  230 DO 250 ISTRE=1,NSTRE
      STRES(ISTRE)=0.0
      DO 240 JSTRE=1,NSTRE
  240 STRES(ISTRE)=STRES(ISTRE)+DMATX(ISTRE,
     1JSTRE)*STRAN(JSTRE)
  250 STRIN(ISTRE,NGAST)=STRES(ISTRE)
      IF(NTYPE.EQ.2) STRIN(4,NGAST)=-YOUNG*EIGEN
      IF(NTYPE.EQ.1) STRIN(4,NGAST)=0.0
C
C***  CALCULATE THE EQUIVALENT NODAL FORCES AND
C     ASSOCIATE WITH THE ELEMENT NODES
C
      DO 260 INODE=1,NNODE
      NGASH=(INODE-1)*NDOFN+1
      MGASH=(INODE-1)*NDOFN+2
      ELOAD(IELEM,NGASH)=ELOAD(IELEM,NGASH,
     1-(CARTD(1,INODE)*STRES(1)+CARTD(2,INODE)*
     ISTRES(3))*DVOLU
  260 ELOAD(IELEM,MGASH)=ELOAD(IELEM,MGASH)
     1-(CARTD(1,INODE)*STRES(3)+CARTD(2,INODE)*
     ISTRES(2))*DVOLU
  270 CONTINUE
  280 CONTINUE
  800 CONTINUE
      WRITE(7,970)
  970 FORMAT(1H0,5X,
     136H TOTAL NODAL FORCES FOR EACH ELEMENT)
      DO 290 IELEM=1,NELEM
  290 WRITE(7,975) IELEM,(ELOAD(IELEM,IEVAB),
     1IEVAB=1,NEVAB)
  975 FORMAT(1X,I4,5X,3E12.4/(9X,3E12.4))
      XFORCE=0.0
      YFORCE=0.0
      DO 400 IELEM=1,NELEM
      DO 300 IEVAB=2,NEVAB,2
      ITY=NELEM-IELEM+1
      ICOUNT=IEVAB-1
      XFORCE=ELOAD(ITY,ICOUNT)+XFORCE
  300 YFORCE=ELOAD(ITY,IEVAB)+YFORCE
  400 WRITE(7,6001)ITY,XFORCE,YFORCE
 6001 FORMAT(1H ,5X,12HELEMENT NO.=,I6,5X,7HX-FORCE=,F12.3
     1,5X,7HYFORCE=,F12.3)
      RETURN
      END
$DEBUG
$LARGE
$NOFLOATCALLS
      SUBROUTINE STRESS
      DIMENSION STRSS(3),ELDIS(2,8),STRSS(3)
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,NGAUS,NPRO
     1P,NMATS,NVFIX,NE/AB,NCASE,NCASE,ITEMP,IPROB,NPROB
      COMMON/LSDATA/COORD(1500,2),PROPS(15,5),PRESC(1750,2),ASDIS(7000),
     1ELOAD(525,16),STRIN(3,4713),NOFIX(1750),IFPRE(1750,2),LNODS(525,8)
```

```
     2,MATNO(525)
      COMMON/NOSH/ELCOD(3,9),SHAPE(8),DERIV(2,9),BMAT(3,16),SAMPD(2,5),
     1DPMAT(3,16),BMAT1(3,16),SMAT((3,16,9),POSSP(3),WEIGP(3),GPCOD(3,3)
     2,NEROP(50)
      DIMENSION LYREL(2,40),PHI(2,40),NOELYR(2),COUNT(2)
      DIMENSION STRESS(9,4),ELSTR(2,40,9)
      READ(5,5001)NOLYR
 5001 FORMAT(16I5)
      DO 100 I=1,NOLYR
      READ(5,5001)NOELS
      READ(5,5001)(LYREL(I,J),J=1,NOELS)
      KOUNT(I)=0
      NOELYR(I)=NOELS
  100 CONTINUE
      NSTR1=NSTRE+1
      WRITE(7,900)
      WRITE(7,905)
  905 FORMAT(1H0,4HG.P.,2X,8HX-COORD.,2X,
     18HY-COORD.,3X,8HX-STRESS,4X,8HY-STRESS,
     13X,9HXY-STRESS,3X,8HZ-STRESS,4X,
     38HMAX P.S.,4X,8HMIN P.S.,6X,5HANGLE)
      KGAST=0
C
C***  LOOP OVER EACH ELEMENT
C
      DO 60 IELEM=1,NELEM
      LPROP=MATNO(IELEM)
      POISS=PROPS(LPROP,2)
C
C***  READ THE STRESS MATRIX , SAMPLING POINT
C     COORDINATES FOR THE ELEMENT
C
      READ(10) BMAT,SPCOD
      WRITE(7,910) IELEM
C
C***  IDENTIFY THE DISPLACEMENTS OF THE
C     ELEMENT NODAL POINTS
C
      DO 10 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      NPOSN=(LNODE-1)*NDOFN
      DO 10 IDOFN=1,NDOFN
      NPOSN=NPOSN+1
      ELDIS(IDOFN,INODE)=ASDIS(NPOSN)
   10 CONTINUE
      IGASP=0
C
C***  ENTER LOOPS OVER EACH SAMPLING POINT
C
      DO 50 IGAUS=1,NGAUS
      DO 50 JGAUS=1,NGAUS
      KGAST=KGAST+1
      IGASP=IGASP+1
C
C***  COMPUTE THE CARTESIAN STRESS COMPONENTS
```

```
C     AT THE SAMPLING POINTS
C
      DO 20 ISTRE=1,NSTRE
      STRSG(ISTRE)=0.0
      KGASH=0
      DO 20 INODE=1,NNODE
      DO 20 IDOFN=1,NDOFN
      KGASH=KGASH+1
      STRSG(ISTRE)=STRSG(ISTRE)+SHAPE(ISTRE,
     1KGASH,KGASP)*ELDIS(IDOFN,INODE)
      STRESS(IGASP,ISTRE)=STRSG(ISTRE)
   20 CONTINUE
C
C***  COMPUTE THE OUT OF PLANE WORKING STRESS
C     COMPONENT
C
      IF(NTYPE.EQ.2) STRSG(4)=POISS*(STRSG(1)+
     1STRSG(2))
      IF(NTYPE.EQ.1) STRSG(4)=0.0
C
C***  FOR THERMAL LOADING ADD ON THE INITIAL
C     THERMAL STRESSES
C
      IF(ITEMP.EQ.0) GO TO 40
      DO 30 ISTR1=1,NSTR1
      STRSG(ISTR1)=STRSG(ISTR1)+STRIN(ISTR1,NGAUS)
   30 CONTINUE
C
C***  COMPUTE THE PRINCIPAL STRESSES
C
   40 XGASH=(STRSG(1)+STRSG(2))*0.5
      XGISH=(STRSG(1)-STRSG(2))*0.5
      XGESH=STRSG(3)
      XGOSH=SQRT(XGISH*XGISH+XGESH*XGESH)
      STRSP(1)=XGASH+XGOSH
      STRSP(2)=XGASH-XGOSH
      IF(XGISH.EQ.0.0) XGISH=0.1E-20
      STRSP(3)=ATAN(XGESH/XGISH)*28.64788975
C
C***  OUTPUT THE STRESSES
C
 7001 FORMAT(2I5,2F10.4/6E12.5/F10.4)
      WRITE(7,915) NGASP,(GPCOD(IDIME,NGASP),
     1IDIME=1,NDIME),(STRSG(ISTR1),ISTR1=1,NSTR1),
     1(STRSP(ISTRE),ISTRE=1,NSTRE)
   50 CONTINUE
      DO 300 I=1,NOLYR
      NOEL=NOELYR(I)
      DO 300 J=1,NOEL
      IF(IELEM.NE.LYREL(I,J))GO TO 300
      NOUNT(I)=NOUNT(I)+1
      JSTR=NOUNT(I)
      ELSTR(I,JSTR,1)=IELEM
      SIGX=(STRESS(3,1)+STRESS(7,1)+STRESS(9,1))/3.0
      SIGY=(STRESS(3,2)+STRESS(7,2)+STRESS(9,2))/3.0
```

```
      TAUXY=(STRESS(I,3)+STRESS(I,2)+STRESS(I,3))/3.0
      ICORD1=LNODS(IELEM,6)
      ICORD2=LNODS(IELEM,7)
      DIFX=COORD(ICORD1,1)-COORD(ICORD2,1)
      DIFY=COORD(ICORD1,2)-COORD(ICORD2,2)
      DIFF=SQRT((DIFX*DIFX)+(DIF*DIFY))
      ELSTR(I,JSTR)=DIFF
      THETA=ATAN(DIF/DIFY)
      ALPHA=1.570796-THETA
      SIG=SIGX*COS(ALPHA)*COS(ALPHA)+SIGY*SIN(ALPHA)*SIN(ALPHA
     1-2.0*TAUXY*SIN(ALPHA)*COS(ALPHA)
      TAU=TAUX*(SIN(ALPHA)*SIN(ALPHA)-TAUX*COS(ALPHA)*COS(ALPHA)
     1+(SIGY-SIGX)*SIN(ALPHA)*COS(ALPHA))
      ELSTR(I,JSTR,3)=SIGX
      ELSTR(I,JSTR,4)=SIGY
      ELSTR(I,JSTR,5)=TAUXY
      ELSTR(I,JSTP,6)=SIG
      ELSTR(I,JSTF,7)=TAU
      ELSTR(I,JSTR,8)=SIG*DIFF
      ELSTR(I,JSTR,9)=TAU*DIFF
  300 CONTINUE
   60 CONTINUE
      WRITE(7,4001)
 4001 FORMAT(1H1,20X,80(1H*))
      WRITE(7,4002)
 4002 FORMAT(1H0,25X,'RESULTS OF EQUILIBRIUM CHECKS AND FACTORS
     1,'OF SAFETY')
      WRITE(7,4003)
 4003 FORMAT(1H0,20X,80(1H*))
      DO 500 I=1,NOLYR
      WRITE(7,4007)I
 4007 FORMAT(1H0,5X,'TABLE OF STRESSES FOR LAYER',I2,
      WRITE(7,4004)
 4004 FORMAT(1H0,5X,100(1H-))
      WRITE(7,4005)
 4005 FORMAT(1H0,5X,'ELEMENT',2X,'ELEMENT',2X,'SIGMAX',2X,
     1'SIGMAY',2X,'TAUXY',2X,'SIG',2X,'TAU',5X,'Y',5X,'X',5X,
     2'NO',5X,'WIDTH')
      WRITE(7,4004)
      NOEL=NOELYR(I)
      WRITE(7,4006)((ELSTR,I,J,I),K=1,9),J=1,NOEL)
 4006 FORMAT(1H0,5X,9 F10.3,2X))
      WRITE(7,4011)
 4011 FORMAT(//////////,5X,100(1H-))
  500 CONTINUE
  900 FORMAT(/,10X,8HSTRESSES/)
  910 FORMAT(/,5X,12HELEMENT NO.=,I5)
  915 FORMAT(I5,2F10.4,6E12.5,F10.4)
      RETURN
      END
      SUBROUTINE CHECK1
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,NGAUS,NPRO
     1P,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,NPROB
      COMMON/LGDATA/COORD(3500,2),PROPS(15,5),PRESC(1750,2),ASDIS(7000),
     1ELOAD(525,16),STRIN(3,4725),NOFIX(1750),IFPRE(1750,2),LNODS(525,8)
```

```
     2,MATNO(525)
      COMMON/WORK/ELCOD(3,8),SHAPE(8),DERIV(2,8),DMATX(3,3),CARTD(2,8),
     1DBMAT(3,16),BMATX(3,16),SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(3,9)
     2,NEROR(50)
C
C*** TO CRITICIZE THE DATA CONTROL AND TO
C     PRINT ANY DIAGNOSTICS
C
        DO 10 IEROR=1,24
   10 NEROR(IEROR)=0
C
C*** CREATE THE DIAGNOSTIC MESSAGES
C
      IF(NPOIN.LE.0) NEROR(1)=1
      IF(NELEM*NNODE.LT.NPOIN) NEROR(2)=1
      IF(NVFIX.LT.1.OR.NVFIX.GT.NPOIN) NEROR(3)=1
      IF(NCASE.LE.0) NEROR(4)=1
      IF(NTYPE.LT.1.OR.NTYPE.GT.3) NEROR(5)=1
      IF(NNODE.LT.3.OR.NNODE.GT.8) NEROR(6)=1
      IF(NDOFN.LT.2.OR.NDOFN.GT.3) NEROR(7)=1
      IF(NMATS.LE.0.OR.NMATS.GT.NELEM) NEROR(8)=1
      IF(NPROP.LT.3.OR.NPROP.GT.5) NEROR(9)=1
      IF(NGAUS.LT.2.OR.NGAUS.GT.3) NEROR(10)=1
      IF(NDIME.LT.1.OR.NDIME.GT.3) NEROR(11)=1
      IF(NSTRE.LT.2.OR.NSTRE.GT.5) NEROR(12)=1
C
C*** EITHER RETURN,OR ELSE PRINT THE ERRORS
C     DIAGNOSED
C
      KEROR=0
      DO 20 IEROR=1,12
      IF(NEROR(IEROR).EQ.0) GO TO 20
      KEROR=1
      WRITE(7,900) IEROR
  900 FORMAT(//23H *** DIAGNOSIS BY CHECK1,
     16H ERROR,I3)
   20 CONTINUE
      IF(KEROR.EQ.0) RETURN
C
C*** OTHERWISE ECHO ALL THE REMAINING DATA
C     WITHOUT FURTHER COMMENT
C
      CALL ECHO
      END
      SUBROUTINE ECHO
      DIMENSION NTITL(80)
      COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,NGAUS,NPRO
     1P,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,NPROB
      COMMON/LGDATA/COORD(3500,2),PROPS(15,5),PRESC(1750,2),ASDIS(7000),
     1ELOAD(525,16),STRIN(3,4725),NOFIX(1750),IFPRE(1750,2),LNODS(525,8)
     2,MATNO(525)
      COMMON/WORK/ELCOD(3,8),SHAPE(8),DERIV(2,8),DMATX(3,3),CARTD(2,8),
     1DBMAT(3,16),BMATX(3,16),SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(3,9)
     2,NEROR(50)
      WRITE(7,900)
```

```
900 FORMAT(//15H NOW FOLLOWS A LISTING OF,
   125H POST-DISASTER DATA CARDS/)
 10 READ(5,905,END=9990) NTITL
905 FORMAT(90A1)
    WRITE(7,910) NTITL
910 FORMAT(20X,90A1)
    GO TO 10
9990 CONTINUE
    RETURN
    END
    SUBROUTINE CHECK2
    DIMENSION NDFRO(525)
    COMMON/CONTRO/NPOIN,NELEM,NNODE,NVFIX,NDIME,NSTRE,NTYPE,NEVAB,NPRO
   1P,NMATS,NVFIX,NSVAB,ICASE,NCASE,ITEMP,IPROB,NPROB
    COMMON/LGDATA/COORD(3500,2),PROPS(15,5),PRESC(750,2),RLOAD(7000),
   1ELOAD(525,16),STRIN(3,4725),NOFIX(1750),IFPRE(750,2),LNODS(525,8)
   2,MATNO(525)
    COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DJACM(2,2),CARTD(2,8),
   1DBMAT(3,16),BMATX(3,16),DMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(3,9)
   2,NEROR(50)
C
C*** TO CRITICIZE THE DATA FROM SUBROUTINE INPUT
C
    NFRON=125
C
C*** CHECK AGAINST TWO IDENTICAL NONZERO
C    NODAL COORDINATES
C
    DO 10 IELEM=1,NELEM
 10 NDFRO(IELEM)=0
    DO 40 IPOIN=2,NPOIN
    KPOIN=IPOIN-1
    DO 30 JPOIN=1,KPOIN
    DO 20 IDIME=1,NDIME
    IF(COORD(IPOIN,IDIME).NE.COORD(JPOIN,
   1IDIME)) GO TO 30
 20 CONTINUE
    NEROR(12)=NEROR(12)+1
    WRITE(7,70001) IPOIN,JPOIN
70001 FORMAT(1X,'*** IPOIN,JPOIN *** ',2I5)
 30 CONTINUE
 40 CONTINUE
C
C*** CHECK THE LIST OF ELEMENT PROPERTY NUMBERS
C
    DO 50 IELEM=1,NELEM
 50 IF(MATNO(IELEM).LE.0.OR.MATNO(IELEM).GT.
   1NMATS) NEROR(14)=NEROR(14)+1
C
C*** CHECK FOR IMPOSSIBLE NODE NUMBERS
C
    DO 70 IELEM=1,NELEM
    DO 60 INODE=1,NNODE
    IF(LNODS(IELEM,INODE).EQ.0) NEROR(15)=
   1NEROR(15)+1
```

```
      IF(LNODS(IELEM,INODE).LT.0.OR.LNODS(IELEM,
     1INODE).GT.NPOIN) THEN
           NEROR(16)=NEROR(16)+1
           WRITE(7,7000) IELEM,LNODS(IELEM,INODE)
7000   FORMAT(1X,'*** IELEM,LNODS ***',5X,2I10)
       ENDIF
   60  CONTINUE
   70 CONTINUE
C
C***  CHECK FOR ANY REPETITION OF A NODE
C     NUMBER WITHIN AN ELEMENT
C
      DO 140 IPOIN=1,NPOIN
      KSTAR=0
      DO 100 IELEM=1,NELEM
      KZERO=0
      DO 90 INODE=1,NNODE
      IF(LNODS(IELEM,INODE).NE.IPOIN) GO TO 90
      KZERO=KZERO+1
      IF(KZERO.GT.1) NEROR(12)=NEROR(12)+1
C
C***  SEEK FIRST, LAST AND INTERMEDIATE
C     APPEARANCES OF NODE IPOIN
C
      IF(KSTAR.NE.0) GO TO 80
      KSTAR=IELEM
C
C***  CALCULATE INCREASE OR DECREASE IN
C     FRONTWIDTH AT EACH ELEMENT STAGE
C
      NDFRO(IELEM)=NDFRO(IELEM)+NDOFN
   80 CONTINUE
C
C***  AND CHANGE THE SIGN OF THE LAST
C     APPEARANCE OF EACH NODE
C
      KLAST=IELEM
      NLAST=INODE
   90 CONTINUE
  100 CONTINUE
      IF(KSTAR.EQ.0) GO TO 110
      IF(KLAST.LT.NELEM) NDFRO(KLAST+1)=
     1NDFRO(KLAST+1)-NDOFN
      LNODS(KLAST,NLAST)=-IPOIN
      GO TO 140
C
C***  CHECK THAT COORDINATES FOR AN UNUSED
C     NODE HAVE NOT BEEN SPECIFIED
C
  110 WRITE(6,900) IPOIN
  900 FORMAT(/15H CHECK WHY NODE,I4,
     114H NEVER APPEARS)
      NEROR(8)=NEROR(8)+1
      SIGMA=0.0
      DO 120 IDIME=1,NDIME
```

```
  120 SIGMA=SIGMA+ABS(COORD(IPOIN,IDIME))
      IF(SIGMA.NE.0. .AND. NERCR(1).EQ.0)NERCR(1)=1
C
C***  CHECK THAT AN UNUSED NODE NUMBER IS NOT
C     A RESTRAINED NODE
C
      DO 130 IVFIX=1,NVFIX
  130 IF(NOFIX(IVFIX).EQ.IPOIN) NERCR(20)=
     1NERCR(20)+1
  140 CONTINUE
C
C***  CALCULATE THE LARGEST FRONTWIDTH
C
      NFRON=0
      KFRON=0
      DO 150 IELEM=1,NELEM
      NFRON=NFRON+NDFRO(IELEM)
  150 IF(NFRON.GT.KFRON) KFRON=NFRON
      WRITE(7,905) KFRON
  905 FORMAT(//28HMAX FRONTWIDTH ENCOUNTERED =,I5,
      IF(KFRON.GT.MFRON) NERCR(21)=1
C
C***  CONTINUE CHECKING THE DATA FOR THE
C     FIXED VALUES
C
      DO 170 IVFIX=1,NVFIX
      IF(NOFIX(IVFIX).LE.0.OR.NOFIX(IVFIX).
     1GT.NPOIN) NERCR(22)=NERCR(22)+1
      KOUNT=0
      DO 160 IDOFN=1,NDOFN
  160 IF(IFPRE(IVFIX,IDOFN).GT.0) KOUNT=1
      IF(KOUNT.EQ.0) NERCR(23)=NERCR(23)+1
      KVFIX=IVFIX-1
      DO 170 JVFIX = 1, KVFIX
  170 IF(IVFIX .NE. 1 .AND. NOFIX(IVFIX).EQ.
     1  NOFIX(JVFIX)) NERCR(24)=NERCR(24)+1
      KERCR = 0
      DO 180 IERCR = 13,24
         IF(NERCR(IERCR) .EQ. 0)GO TO 180
           KERCR = 1
           WRITE(7,910) IERCR,NERCR(IERCR)
  910      FORMAT(//' ***DIAGNOSIS BY CHECK2, ERROR',
     1     I3,6X,' ASSOCIATED NUMBER',I5)
  180   CONTINUE
        IF(KERCR.NE.0)GO TO 200
C
C***  RETURN ALL NODAL CONNECTION NUMBERS TO
C     POSITIVE VALUES
C
      DO 190 IELEM = 1,NELEM
         DO 190 INODE = 1,NNODE
            LNODS(IELEM,INODE)=IABS(LNODS(IELEM,INODE))
  190   CONTINUE
        RETURN
  200   CALL ECHO
```

(B) CHANGE IN D MATRIX

```
C
C***  D MATRIX FOR PLANE STRESS CASE
C
        IF(LFROF.EQ.1)THEN
      CONST=YOUNG/(1.0-POISS*POISS)
      DMATX(1,1)=CONST
      DMATX(2,2)=CONST
      DMATX(1,2)=CONST*POISS
      DMATX(2,1)=CONST*POISS
      DMATX(3,3)=(1.0-POISS)*CONST/2.0
      ELSE
      CONST=YOUNG
      DMATX(1,1)=CONST
      DMATX(2,2)=CONST/(2D*(1.0-POISS)*(2.0*POISS+..
      DMATX(3,3)=CONST/2*(POISS+1.0)
      ENDIF
      GO TO 30
```