Παράρτημα ΣΤ

Πρόγραμμα μητρωικής μελέτης κατασκευής *gril3d*

Στο παράρτημα αυτό περιλαμβάνεται ο πηγαίος κώδικας προγράμματος επίλυσης κατασκευών στην ελαστική περιοχή, με τη μητρωική μέθοδο (μέθοδο των μετατοπίσεων). Περιλαμβάνονται επίσης παραδείγματα αρχείων εισόδου και αντίστοιχων αποτελεσμάτων το πρόγραμμα.

```
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C             Linear Elastic Analysis of a 3-D Grillage
C
C                   using the Matrix Stiffness Method
C
C
C      This program was developed by Dr. P.A. Caridis
C      Department of Naval Architecture and Marine
C      Engineering, NTUA, Athens
C      April 1991.
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C    - Program gril3d.f can be used to analyse the response
C      of a 3-D structure with up to 100 joints. This
C      corresponds to a rectangular grillage having 8 beams
C      in each direction. Array sizes have to be modified
C      should larger structures be considered.
C    - The properties of each beam are allowed to vary
C      although the geometrical and material particulars are
C      assumed constant between joints.
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . C
       REAL IZ,IY,K,KMEM,KSS,KS,K11,K12,K21,K22,L,LX,LY,LZ,
      &LENXZ,LEN1,NU
       COMMON/real/R(3,3),T(100,12,12),TT(100,12,12),K(12,12
      &),P(600),D(600),PJNT(100,6),PMG(100,12),KSS(600,600),
      &KS(600,600),XJC(100),YJC(100),X1(100),Y1(100),X2(100)
      &,Y2(100),K11(6,6),K12(6,6),K21(6,6),K22(6,6),ANG(100)
      &,L(100),PROD(12,12),RED(600,600),PML(100,12),BMOM(100
      &,5),SX(100,5),X(5),DMG(100,12),DML(100,12),KMEM(100,1
      &2,12),RML(100,12),RMG(100,12),PP(600),PJP(100,6),PFIX
      &(100,6),RS(600),DMEM(100,5),Z1(100),Z2(100),ZJC(100),
      &LX(100),LY(100),LZ(100),LENXZ(100),LEN1(100)
       COMMON/ints/MNO,JNO,ISSNO,ICLNO,LC,NUMSEC,ID(100,6),J
      &N(100),J1(100),J2(100),ISEC(100),IDLIN(600),JNUM(2),N
      &DOF,IOUT(100),IYLD(100,5),ISPAC
      COMMON/mats/ E(100),NU(100),B(100),TP(100),HW(100),TW(
      &100),BF(100),TF(100),AREA(100),IZ(100),IY(100),TORC(1
      &00),YZZ(100),RADZ(100),ZZ(100),YYY(100),RADY(100),ZY(
      &100),UDL(100),YS(100)
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
      OPEN(5, FILE='gril3.dat')
      OPEN(6, FILE='gril3.out')
C
C     Initialisations
C
```

```
      CALL INITL
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C
C                          L e g e n d
C
C
C      ISPAC      :   =1 for space frame, =0 for plane frame
C      MNO        :   No. of members in structure
C      JNO        :   No. of joints in structure
C      NUMSEC     :   No. of section types
C      NLOAD      :   No. of applied external loads
C      ISUPNO     :   No. of suppressed degrees of freedom
C      IOUT       :   No. of members for which output is
C                     produced
C      XJC(I)     :   X coord of Ith joint
C      YJC(I)     :   Y coord of Ith joint
C      ZJC(I)     :   Z coord of Ith joint
C      E(I),NU(I) :   Young's Modulus, Poisson's ratio
C      YS(I)      :   Yield stress for Ith member
C      B(I),TP(I) :   Effective breadth and thickness of
C                     plating
C      HW(I),TW(I):   Height and thickness of web
C      BF(I),TF(I):   Breadth and thickness of flange
C      ISEC(I)    :   Section type for Ith stiffener
C      J1(I)      :   Joint number at end 1 for Ith member
C      J2(I)      :   Joint number at end 2 for Ith member
C      PJNT(I)    :   Load acting on Ith joint with J d.o.f.
C      P(I,J)     :   Load vector for Ith member and Jth
                       d.o.f.
C      D(I,J)     :   Displacement vector for Ith joint with
                       J d.o.f
C      ID(I,J)    :   Identification array for active &
C                     suppressed d.o.f.
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C
      READ(5,*) ISPAC,MNO,JNO,NUMSEC
      READ(5,*) NLOAD,NUDL,ISUPNO,IOMAX
      READ(5,*) (I,XJC(I1),YJC(I1),ZJC(I1),I1=1,JNO)
      READ(5,*) (I,E(I1),NU(I1),YS(I1),B(I1),TP(I1),HW(I1),
     &TW(I1),BF(I1),TF(I1),I1=1,NUMSEC)
       READ(5,*) (IMEM,ISEC(I1),J1(I1),J2(I1),
     &ANG(I1),I1=1,MNO)
       IF(NLOAD.NE.0) READ(5,*) (JNT,IDOF,PJNT(JNT,
     &IDOF),LOD=1,NLOAD)
       IF(NUDL.NE.0)  READ(5,*) (IMEM,UDL(IMEM),LOD=1,NUDL)
       DO 1 ISUP=1,ISUPNO
       READ(5,*) JNT,IDOF
```

```
   1   ID(JNT,IDOF)=1
       DO 2 I=1,IOMAX
       READ(5,*) J
   2   IOUT(J)=1
       DO 3  IMEM=1,MNO
       DO 3 I=1,5
   3   IYLD(IMEM,I)=0
C
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C
C     Length and orientation angle of each member
C
      DO 4 I=1,MNO
      J=J1(I)
      X1(I)=XJC(J)
      Y1(I)=YJC(J)
      Z1(I)=ZJC(J)
      J=J2(I)
      X2(I)=XJC(J)
      Y2(I)=YJC(J)
   4  Z2(I)=ZJC(J)
C
      DO 5 I=1,MNO
      LX(I)=X2(I)-X1(I)
      LY(I)=Y2(I)-Y1(I)
      LZ(I)=Z2(I)-Z1(I)
      XX=LX(I)*LX(I)+LY(I)*LY(I)
      XY=XX+LZ(I)*LZ(I)
      L(I)=SQRT(XY)
      XZ=LX(I)*LX(I)+LZ(I)*LZ(I)
      LENXZ(I)=SQRT(XZ)
      LEN1(I)=L(I)*LENXZ(I)
      IF(ISPAC.EQ.1) GO TO 5
      RATIO=LY(I)/L(I)
      ANG(I)=ASIN(RATIO)
   5  CONTINUE
      NDOF=6*JNO
C
C     Geometrical properties of all sections
C
      CALL PROPS
C
C     Echo of input data
C
      CALL INPUT
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
```

```
C     Assemble member stiffness matrices
C
      DO 20 IMEM=1,MNO
      CALL STIF(IMEM,ISEC,E,NU,AREA,L,IZ,IY,TORC,K)
C
C     WRITE(6,39) IMEM
C     WRITE(6,42) (K(I,J),I=1,12),J=1,12)
C
C     Member transformation matrix
C
      CALL ROT(IMEM,ISPAC,ANG,R,T,TT,L,LX,LY,LZ,LEN1,LENXZ)
C
C     ------------------------------------------------------
C
C     Obtain particular solution with clamped joints for
C     distributed loads
C
      IF(UDL(IMEM).EQ.0.0) GO TO 10
      CALL PARSOL(IMEM)
C
C     Reactions at joints due to UDLs
C
C     a) from member local to member global coords
C
      DO 7 J=1,12
      PMG(IMEM,J)=0.0
      DO 7 I=1,12
    7 PMG(IMEM,J)=PMG(IMEM,J)+TT(IMEM,I,J)*PML(IMEM,I)
C
C     b) from member global to joint global loads due to
C     UDLs
C
      JNT1=J1(IMEM)
      JNT2=J2(IMEM)
      DO 8 IDOF=1,6
      PJP(JNT1,IDOF)=PJP(JNT1,IDOF)+PMG(IMEM,IDOF)
      PJP(JNT2,IDOF)=PJP(JNT2,IDOF)+PMG(IMEM,IDOF+6)
      IF(ID(JNT1,IDOF).EQ.1) PFIX(JNT1,IDOF)=PJP(JNT1,IDOF)
      IF(ID(JNT2,IDOF).EQ.1) PFIX(JNT2,IDOF)=PJP(JNT2,IDOF)
      IF(ID(JNT1,IDOF).EQ.1) PJP(JNT1,IDOF)=0.0
    8 IF(ID(JNT2,IDOF).EQ.1) PJP(JNT2,IDOF)=0.0
C
C     Column load vector containing reactions to UD loads
C
      DO 9 JNT=1,JNO
      DO 9 IDOF=1,6
      INOD=6*(JNT-1)
    9 PP(INOD+IDOF)=PJP(JNT,IDOF)
   10 CONTINUE
C
C     ------------------------------------------------------
```

```
C
C     Structural load vector including member UDL loads and
C     joint loads
C
      DO 11 JNT=1,JNO
      DO 11 IDOF=1,6
      INOD=6*(JNT-1)
  11  P(INOD+IDOF)=PJNT(JNT,IDOF)-PP(INOD+IDOF)
C
      DO 12 J=1,12
      DO 12 I=1,12
  12  KMEM(IMEM,I,J)=K(I,J)
C
C     Transform member stiffness matrix to global coordinate
C     system
C
      CALL STIFGL(IMEM)
C
      DO 13 JNT=1,JNO
      DO 13 IDOF=1,6
      INOD=6*(JNT-1)
  13  IDLIN(INOD+IDOF)=ID(JNT,IDOF)
C
C     Partition member stiffness matrix
C
      DO 14 J=1,6
      DO 14 I=1,6
      K11(I,J)=K(I,J)
      K12(I,J)=K(I+6,J)
      K21(I,J)=K(I,J+6)
  14  K22(I,J)=K(I+6,J+6)
C
C     WRITE(*,101)IMEM
C     PRINT *,IMEM
      IF(IOUT(IMEM).EQ.0) GO TO 15
C     WRITE(6,45) IMEM
C     WRITE(6,44) ((R(I,J),I=1,3),J=1,3)
C     WRITE(6,46) IMEM
C     WRITE(6,42) ((T(IMEM,I,J),I=1,12),J=1,12)
C     WRITE(6,47) IMEM
C     WRITE(6,42) ((TT(IMEM,I,J),I=1,12),J=1,12)
C     WRITE(6,34) IMEM
C     WRITE(6,42) ((K(I,J),I=1,12),J=1,12)
C     WRITE(6,35) IMEM
C     WRITE(6,43) ((K11(I,J),I=1,6),J=1,6)
C     WRITE(6,36) IMEM
C     WRITE(6,43) ((K12(I,J),I=1,6),J=1,6)
C     WRITE(6,37) IMEM
C     WRITE(6,43) ((K21(I,J),I=1,6),J=1,6)
C     WRITE(6,38) IMEM
```

```
C     WRITE(6,43) ((K22(I,J),I=1,6),J=1,6)
C
C     Form structural stiffness matrix {K} using the direct
C     stiffness method
C
  15  JNUM1=J1(IMEM)
      JF=1+6*(JNUM1-1)
      IF=JF
      DO 16 J=JF,JF+5
      JEL=J-JF+1
      DO 16 I=IF,IF+5
      IEL=I-IF+1
  16  KS(I,J)=KS(I,J)+K11(IEL,JEL)
C
      JNUM2=J2(IMEM)
      IF=1+6*(JNUM2-1)
      JF=1+6*(JNUM1-1)
      DO 17 J=1,6
      DO 17 I=1,6
      IS=IF-1+I
      JS=JF-1+J
  17  KS(IS,JS)=KS(IS,JS)+K12(I,J)
C
      IF=1+6*(JNUM1-1)
      JF=1+6*(JNUM2-1)
      DO 18 J=1,6
      DO 18 I=1,6
      IS=IF-1+I
      JS=JF-1+J
  18  KS(IS,JS)=KS(IS,JS)+K21(I,J)
C
      IF=1+6*(JNUM2-1)
      JF=1+6*(JNUM2-1)
      DO 19 J=1,6
      DO 19 I=1,6
      IS=IF-1+I
      JS=JF-1+J
  19  KS(IS,JS)=KS(IS,JS)+K22(I,J)
  20  CONTINUE
      WRITE(*,102)
C
      DO 21 J=1,NDOF
      DO 21 I=1,NDOF
  21  KSS(I,J)=KS(I,J)
C
C     Obtain reduced structural stiffness matrix
C
      DO 22 J=1,NDOF
      DO 22 I=1,NDOF
      IF((IDLIN(J).EQ.1).AND.(I.EQ.J)) KS(I,J)=1.0
  22  IF((IDLIN(J).EQ.1).AND.(I.NE.J)) KS(I,J)=0.0
```

```
C
      DO 23 I=1,NDOF
      DO 23 J=1,NDOF
  23  IF((KS(I,I).EQ.0.0).AND.(J.NE.I)) KS(I,J)=0.0
      DO 24 J=1,NDOF
      DO 24 I=1,NDOF
  24  RED(I,J)=KS(I,J)
      DO 25 I=1,NDOF
  25  IF(IDLIN(J).EQ.1) P(J)=0.0
      WRITE(*,103)
C
C     Solve reduced system of equations using Gauss
C     elimination
C
C     a) Stiffness matrix
C
      DO 26 I=1,NDOF-1
      DO 26 J=1,NDOF
      IF(J.LE.I) GO TO 26
      IF(KS(I,J).EQ.0.0) GO TO 26
      FACTOR=KS(I,J)/KS(I,I)
      KS(I,J)=KS(I,J)-FACTOR*KS(I,I)
      DO 27 M=I+1,NDOF
  27  KS(M,J)=KS(M,J)-FACTOR*KS(M,I)
C
C     b) Load vector
C
      P(J)=P(J)-FACTOR*P(I)
  26  CONTINUE
      WRITE(*,104)
C
C     Check ill-conditioning of stiffness matrix
C
      ICONT=1
      DO 28 I=1,NDOF
      IF(KS(I,I).LT.0.0) WRITE(*,40) I,KS(I,I)
      IF(KS(I,I).EQ.0.0) ICONT=0
  28  IF(KS(I,I).EQ.0.0) WRITE(*,41) I
      IF(ICONT.EQ.0) GO TO 100
C
C     Back-substitute to obtain displacement vector
C
      DO 29 JJ=1,NDOF
      J=NDOF-JJ+1
      SUM=0.0
      DO 30 I=J+1,NDOF
  30  SUM=SUM+KS(I,J)*D(I)
  29  D(J)=(P(J)-SUM)/KS(J,J)
      WRITE(*,105)
C
```

```
C     Structure load vector {R} (incl. reactions at
C     supports)
C
      DO 31 J=1,NDOF
      RS(J)=0.0
      DO 31 I=1,NDOF
  31  RS(J)=RS(J)+KSS(I,J)*D(I)
C
C     Structure load vector {R} including reactions due to
C     UD loads
C
      DO 32 I=1,NDOF
  32  RS(I)=RS(I)+PP(I)
      DO 33 JNT=1,JNO
      DO 33 IDOF=1,6
      INOD=6*(JNT-1)
  33  RS(INOD+IDOF)=RS(INOD+IDOF)+PFIX(JNT,IDOF)
C
C     Obtain member load vectors in local coordinates
C
      CALL LOAD
C
C     Output
C
      WRITE(*,106)
      CALL OUT
      GO TO 34
  44   FORMAT(3(10X,G12.5))
  39   FORMAT(//,25X,'Member No. ',I2,' - Stiffness matrix
     &{k}',//)
  45   FORMAT(//,25X,'Member No. ',I2,' - Rotation matrix
     &{Ro}',//)
  46   FORMAT(//,25X,'Member No. ',I2,' - Transformation
     &matrix {T}',//)
  47   FORMAT(//,25X,'Member No. ',I2,' - Transpose of
     &Transformation matrix {Tt}',//)
C 34   FORMAT(//,25X,'Member No. ',I2,
C     &' - Stiffness matrix {k',1H','} (={Tt}{k}{T})',//)
  35   FORMAT(//,25X,'Member No. ',I2,
     &' - Partitioned stiffness matrix {k11',1H','}',//)
  36   FORMAT(//,25X,'Member No. ',I2,
     &' - Partitioned stiffness matrix {k12',1H','}',//)
  37   FORMAT(//,25X,'Member No. ',I2,
     &' - Partitioned stiffness matrix {k21',1H','}',//)
  38   FORMAT(//,25X,'Member No. ',I2,
     &' - Partitioned stiffness matrix {k22',1H','}',//)
  40   FORMAT(//,5X,'*** WARNING - Negative Diagonal Term –
     &I = ',I2,5X,'KS = ',G12.5,//)
  41   FORMAT(//,5X,'*** WARNING - I = ',I2,5X,
     &'Stiffness Matrix Found Indeterminate',
     &//,25X,'PROGRAM ABORTS',/)
```

```
 42   FORMAT(12(2X,G9.2))
 43   FORMAT(6(5X,G12.5))
101   FORMAT(/,5X,'Member stiffness matrix {k} no',I3,'
     &partitioned',/)
102   FORMAT(/,5X,'Structural stiffness matrix {K}
     &assembled',//)
103   FORMAT(5X,'Reduction of {K} accomplished',//)
104   FORMAT(5X,'Upper triangular form of {K} obtained',//)
105   FORMAT(5X,'Displacements at all joints obtained',//)
106   FORMAT(5X,'Reactions at all joints obtained –
     &Solution completed',//)
200   FORMAT(5X,'I = ',I2,2X,'M = ',I2,2X,'J = ',I2,2X,'KS
     &',G12.5)
203   FORMAT(' I = ',I2,' IDOF= ',I2,' RS = ',G12.5)
100   CALL OUT
 34   CONTINUE
      STOP
      END
```

```
      SUBROUTINE INITL
C
      COMMON/real/ VAR1(1138146)
      COMMON/mats/ VAR3(2000)
      COMMON/ints/ IVAR(2210)
C
C     Initialisations
C
      DO 1 I=1,1138146
   1  VAR1(I)=0.0
      DO 3 I=1,2000
   3  VAR3(I)=0.0
      DO 5 I=1,2210
   5  IVAR(I)=0
      RETURN
      END
      SUBROUTINE PROPS
C
      REAL IY,IZ,NU
      COMMON/mats/ E(100),NU(100),B(100),TP(100),HW(100),
      &TW(100),BF(100),TF(100),AREA(100),IZ(100),IY(100),TOR
      &C(100),YZZ(100),RADZ(100),ZZ(100),YYY(100),RADY(100),
      &ZY(100),UDL(100),YS(100)
       COMMON/ints/MNO,JNO,ISSNO,ICLNO,LC,NUMSEC,ID(100,6),J
      &N(100),J1(100),J2(100),ISEC(100),IDLIN(600),JNUM(2),N
      &DOF,IOUT(100),IYLD(100,5),ISPAC
C
C     Section properties for flat plating and attached
C     stiffeners
C
      DO 1 I=1,NUMSEC
      AP=B(I)*TP(I)
      AW=HW(I)*TW(I)
      AF=BF(I)*TF(I)
      AREA(I)=AP+AW+AF
      IF(BF(I).EQ.0.0) TF(I)=0.0
      IF(TF(I).EQ.0.0) BF(I)=0.0
C
C     Flatbar stiffener
C
      IF(BF(I).NE.0.0) GO TO 2
      YZZ(I)=(AP*TP(I)/2.0+AW*(TP(I)+HW(I)/2.0))/AREA(I)
      CGF1=B(I)/2.0
      YYY(I)=(AP+AW)*CGF1/AREA(I)
C
       IZ(I)=(B(I)*(TP(I)**3.0)+(HW(I)**3.0)*TW(I))/12.0+
     &(HW(I)/2.0+TP(I)-YZZ(I))**2.0+HW(I)*TW(I)+
     &B(I)*TP(I)*(YZZ(I)-TP(I)/2.0)**2.0
C
       IY(I)=(AP*B(I)*B(I)+AW*TW(I)*TW(I))/12.0+
     &AREA(I)*(B(I)-YYY(I))*(B(I)-YYY(I))
```

```
      GO TO 3
C
C     b) T-bar or angle-bar stiffener
C
   2  YZZ(I)=(B(I)*(TP(I)**2.0)/2.0+(HW(I)**2.0)*
     &TW(I)/2.0+HW(I)*TP(I)*TW(I)+BF(I)*TF(I)*HW(I)+
     &BF(I)*TF(I)*TP(I)+BF(I)*(TF(I)**2.0)/2.
     &0)/(B(I)*TP(I)+HW(I)*TW(I)+BF(I)*TF(I))
      CGF1=B(I)/2.0
      CGF2=CGF1
      IANG=0
      IF(IANG.EQ.1) CGF2=CGF2+BF(I)/2.0
      YYY(I)=((AP+AW)*CGF1+AF*CGF2)/AREA(I)
C
      IZ(I)=(B(I)*(TP(I)**3.0)+TW(I)*(HW(I)**3.0)+
     &BF(I)*(TF(I)**3.0))/12.0+B(I)*TP(I)*(YZZ(I)-
     &TP(I)/2.0)**2.0+HW(I)*TW(I)*(HW(I)/2.0+TP(I)
     &-YZZ(I))**2.0+BF(I)*TF(I)*(HW(I)+TP(I)+TF(I)/2.0
     &-YZZ(I))**2.
C
      IY(I)=(AP*B(I)*B(I)+AW*TW(I)*TW(I)+AF*BF(I)*
     &BF(I))/12.0+AREA(I)*(B(I)-YYY(I))*(B(I)-YYY(I))
C
   3  CONTINUE
C
      RADZ(I)=SQRT(IZ(I)/AREA(I))
      RADY(I)=SQRT(IY(I)/AREA(I))
      ZZ(I)=IZ(I)/YZZ(I)
      ZY(I)=IY(I)/YYY(I)
      TORC1=0.0
      TORC(I)=AP*TP(I)*TP(I)+AW*TW(I)*TW(I)+AF*TF(I)*TF(I)
      TORC(I)=TORC(I)/3.0
   1  CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE PARSOL(IMEM)
C
       REAL IZ,IY,K,KMEM,KSS,KS,K11,K12,K21,K22,L,LX,LY,LZ,
      &LENXZ,LEN1,NU

       COMMON/real/R(3,3),T(100,12,12),TT(100,12,12),K(12,12
      &),P(600),D(600),PJNT(100,6),PMG(100,12),KSS(600,600),
      &KS(600,600),XJC(100),YJC(100),X1(100),Y1(100),X2(100)
      &,Y2(100),K11(6,6),K12(6,6),K21(6,6),K22(6,6),ANG(100)
      &,L(100),PROD(12,12),RED(600,600),PML(100,12),BMOM(100
      &,5),SX(100,5),X(5),DMG(100,12),DML(100,12),KMEM(100,1
      &2,12),RML(100,12),RMG(100,12),PP(600),PJP(100,6),PFIX
      &(100,6),RS(600),DMEM(100,5),Z1(100),Z2(100),ZJC(100),
      &LX(100),LY(100),LZ(100),LENXZ(100),LEN1(100)
       COMMON/ints/MNO,JNO,ISSNO,ICLNO,LC,NUMSEC,ID(100,6),J
      &N(100),J1(100),J2(100),ISEC(100),IDLIN(600),JNUM(2),N
      &DOF,IOUT(100),IYLD(100,5),ISPAC
       COMMON/mats/ E(100),NU(100),B(100),TP(100),HW(100),TW
      &(100),BF(100),TF(100),AREA(100),IZ(100),IY(100),TORC(
      &100),YZZ(100),RADZ(100),ZZ(100),YYY(100),RADY(100),ZY
      &(100),UDL(100),YS(100)

C
C     Reactions pz1, pz2 and moments mz1, mz2 on loaded mem
C     ber
C
      PML(IMEM,3)=-UDL(IMEM)*L(IMEM)/2.0
      PML(IMEM,9)=PML(IMEM,3)
C
      PML(IMEM,5)=-UDL(IMEM)*L(IMEM)*L(IMEM)/12.0
      PML(IMEM,11)=-PML(IMEM,5)
      RETURN
      END
```

```
       SUBROUTINE STIF(I,ISEC,E,NU,AREA,L,IZ,IY,TORC,K)
C
       REAL IY,IZ,K,L,L1,L2,L3,NU
       DIMENSION E(100),NU(100),AREA(100),IZ(100),IY(100),
      &TORC(100),K(12,12),L(100),ISEC(100)
C
      IS=ISEC(I)
      L1=1.0/L(IS)
      L2=1.0/(L(IS)*L(IS))
      L3=1.0/(L(IS)*L(IS)*L(IS))
      TORQ=TORC(IS)/(2.0*L(IS)*(1.0+NU(IS)))
C
      DO 1 J1=1,12
      DO 1 J2=1,12
   1  K(J1,J2)=0.0
C
C     Non-zero upper triangular elements of member stiffness
C     matrix {k}
C
      K(1,1)  =+AREA(IS)*L1
      K(7,1)  =-AREA(IS)*L1
      K(2,2)  =+12.0*IZ(IS)*L3
      K(6,2)  =+6.0*IZ(IS)*L2
      K(8,2)  =-12.0*IZ(IS)*L3
      K(12,2) =+6.0*IZ(IS)*L2
      K(3,3)  =+12.0*IY(IS)*L3
      K(5,3)  =-6.0*IY(IS)*L2
      K(9,3)  =-12.0*IY(IS)*L3
      K(11,3) =-6.0*IY(IS)*L2
      K(4,4)  =+TORQ
      K(10,4) =-TORQ
      K(5,5)  =+4.0*IY(IS)*L1
      K(9,5)  =+6.0*IY(IS)*L2
      K(11,5) =+2.0*IY(IS)*L1
      K(6,6)  =+4.0*IZ(IS)*L1
      K(8,6)  =-6.0*IZ(IS)*L2
      K(12,6) =+2.0*IZ(IS)*L1
      K(7,7)  =+AREA(IS)*L1
      K(8,8)  =+12.0*IZ(IS)*L3
      K(12,8) =-6.0*IZ(IS)*L2
      K(9,9)  =+12.0*IY(IS)*L3
      K(11,9) =+6.0*IY(IS)*L2
      K(10,10)=+TORQ
      K(11,11)=+4.0*IY(IS)*L1
      K(12,12)=+4.0*IZ(IS)*L1
C
C     Symmetry about leading diagonal
C
      DO 2 J1=1,12
      DO 2 I1=1,12
```

```
      K(I1,J1)=E(IS)*K(I1,J1)
   2  IF(J1.GT.I1) K(I1,J1)=K(J1,I1)
      RETURN
      END
      SUBROUTINE ROT(I,ISPAC,ANG,R,T,TT,L,LX,LY,LZ,
C     LEN1,LENXZ)
C
      REAL L,LX,LY,LZ,LEN1,LENXZ
      DIMENSION T(100,12,12),TT(100,12,12),R(3,3),ANG(100),
     &L(100),LX(100),LY(100),LZ(100),LEN1(100),LENXZ(100)
C
C     Rotation Matrix {Ro}
C
      SS=SIN(ANG(I))
      CS=COS(ANG(I))
      ABSS=ABS(SS)
      ABCS=ABS(CS)
      IF(ABSS.LT.0.10E-02) SS=0.0
      IF(ABCS.LT.0.10E-02) CS=0.0
      IF(ISPAC.EQ.1) GO TO 1
C
      R(1,1)=CS
      R(2,1)=SS
      R(1,2)=-SS
      R(2,2)=CS
      R(3,3)=1.0
      GO TO 2
C
   1  R(1,1)=LX(I)/L(I)
      R(2,1)=LY(I)/L(I)
      R(3,1)=LZ(I)/L(I)
      R(1,2)=-LX(I)*LY(I)*CS
      R(1,2)=R(1,2)-L(I)*LZ(I)*SS
      R(1,2)=R(1,2)/LEN1(I)
      R(2,2)=LENXZ(I)*CS/L(I)
      R(3,2)=-LY(I)*LZ(I)*CS
      R(3,2)=R(3,2)+L(I)*LX(I)*SS
      R(3,2)=R(3,2)/LEN1(I)
      R(1,3)=LX(I)*LY(I)*SS
      R(1,3)=R(1,3)-L(I)*LZ(I)*CS
      R(1,3)=R(1,3)/LEN1(I)
      R(2,3)=-LENXZ(I)*SS/L(I)
      R(3,3)=LY(I)*LZ(I)*SS
      R(3,3)=R(3,3)+L(I)*LX(I)*CS
      R(3,3)=R(3,3)/LEN1(I)
C
      ISET=0
      IF((LX(I).EQ.0.0).AND.(LZ(I).EQ.0.0)) ISET=1
      IF(ISET.EQ.0) GO TO 3
      R(1,2)=1.0
      R(3,2)=1.0
```

```
      R(1,3)=1.0
      R(3,3)=1.0
   3  CONTINUE
C
C     Transformation Matrix {T}
C
   2  T(I,1,1)=R(1,1)
      T(I,2,1)=R(2,1)
      T(I,1,2)=R(1,2)
      T(I,2,2)=R(2,2)
      T(I,3,3)=R(3,3)
      T(I,4,4)=R(1,1)
      T(I,5,4)=R(2,1)
      T(I,4,5)=R(1,2)
      T(I,5,5)=R(2,2)
      T(I,6,6)=R(3,3)
      T(I,7,7)=R(1,1)
      T(I,8,7)=R(2,1)
      T(I,7,8)=R(1,2)
      T(I,8,8)=R(2,2)
      T(I,9,9)=R(3,3)
      T(I,10,10)=R(1,1)
      T(I,11,10)=R(2,1)
      T(I,10,11)=R(1,2)
      T(I,11,11)=R(2,2)
      T(I,12,12)=R(3,3)
C
C     Transpose of transformation matrix {Tt}
C
      DO 4 J=1,12
      DO 4 I1=1,12
   4  TT(I,I1,J)=T(I,J,I1)
      RETURN
      END
```

```
      SUBROUTINE STIFGL(IMEM)
C
       REAL K,KMEM,KSS,KS,K11,K12,K21,K22,L,LX,LY,LZ
C     &,LENXZ,LEN1

       COMMON/real/R(3,3),T(100,12,12),TT(100,12,12),K(12,12
      &),P(600),D(600),PJNT(100,6),PMG(100,12),KSS(600,600),
      &KS(600,600),XJC(100),YJC(100),X1(100),Y1(100),X2(100)
      &,Y2(100),K11(6,6),K12(6,6),K21(6,6),K22(6,6),ANG(100)
      &,L(100),PROD(12,12),RED(600,600),PML(100,12),BMOM(100
      &,5),SX(100,5),X(5),DMG(100,12),DML(100,12),KMEM(100,1
      &2,12),RML(100,12),RMG(100,12),PP(600),PJP(100,6),PFIX
      &(100,6),RS(600),DMEM(100,5),Z1(100),Z2(100),ZJC(100),
      &LX(100),LY(100),LZ(100),LENXZ(100),LEN1(100)
C
C     Member stiffness matrix in global coordinates {k'}
C
      DO 1 J=1,12
      DO 1 I=1,12
      SUM=0.0
      DO 2 M=1,12
   2  SUM=SUM+K(M,J)*T(IMEM,I,M)
      PROD(I,J)=SUM
   1  CONTINUE
C
      DO 3 J=1,12
      DO 3 I=1,12
      SUM=0.0
      DO 4 M=1,12
   4  SUM=SUM+TT(IMEM,M,J)*PROD(I,M)
      K(I,J)=SUM
   3  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LOAD
C
       REAL IZ,IY,K,KMEM,KSS,KS,K11,K12,K21,K22,L,LX,LY,LZ,
      &LENXZ,LEN1,NU

       COMMON/real/R(3,3),T(100,12,12),TT(100,12,12),K(12,12
      &),P(600),D(600),PJNT(100,6),PMG(100,12),KSS(600,600),
      &KS(600,600),XJC(100),YJC(100),X1(100),Y1(100),X2(100)
      &,Y2(100),K11(6,6),K12(6,6),K21(6,6),K22(6,6),ANG(100)
      &,L(100),PROD(12,12),RED(600,600),PML(100,12),BMOM(100
      &,5),SX(100,5),X(5),DMG(100,12),DML(100,12),KMEM(100,1
      &2,12),RML(100,12),RMG(100,12),PP(600),PJP(100,6),PFIX
      &(100,6),RS(600),DMEM(100,5),Z1(100),Z2(100),ZJC(100),
      &LX(100),LY(100),LZ(100),LENXZ(100),LEN1(100)
       COMMON/ints/MNO,JNO,ISSNO,ICLNO,LC,NUMSEC,ID(100,6),J
      &N(100),J1(100),J2(100),ISEC(100),IDLIN(600),JNUM(2),N
      &DOF,IOUT(100),IYLD(100,5),ISPAC
       COMMON/mats/ E(100),NU(100),B(100),TP(100),HW(100),TW
      &(100),BF(100),TF(100),AREA(100),IZ(100),IY(100),TORC(
      &100),YZZ(100),RADZ(100),ZZ(100),YYY(100),RADY(100),ZY
      &(100),UDL(100),YS(100)
C
C     DMG - member displacement vectors in global co-ordi-
C     nates {r'}
C
      DO 8 IMEM=1,MNO
      X(1)=0.0
      X(2)=L(IMEM)*0.25
      X(3)=L(IMEM)*0.50
      X(4)=L(IMEM)*0.75
      X(5)=L(IMEM)
C
      JN1=J1(IMEM)
      JN2=J2(IMEM)
      DO 1 IDOF=1,6
      JF=6*(JN1-1)
   1  DMG(IMEM,IDOF)=D(JF+IDOF)
      DO 2 IDOF=1,6
      JF=6*(JN2-1)
   2  DMG(IMEM,6+IDOF)=D(JF+IDOF)
C
C     DML - member displacement vectors in local coordinates
C     {r}={T}{r'}
C
      DO 3 J=1,12
      DML(IMEM,J)=0.0
      DO 3 I=1,12
   3  DML(IMEM,J)=DML(IMEM,J)+T(IMEM,I,J)*DMG(IMEM,I)
C
      ISNO=ISEC(IMEM)
```

```
      DENOM=24.0*E(ISNO)*IZ(ISNO)
      UDL1=UDL(IMEM)/DENOM
      DO 4 I=1,5
      C1=X(I)-L(IMEM)
      C2=C1*C1
      C3=X(I)*X(I)
      DMEM(IMEM,I)=UDL1*C3*C2
      C4=X(I)/L(IMEM)
      C5=DML(IMEM,9)-DML(IMEM,3)
      C6=DML(IMEM,3)+C5*C4
   4  DMEM(IMEM,I)=DMEM(IMEM,I)+C6
C
C     RML - member load vectors in local coords {p}={k}{r}
C
      DO 5 J=1,12
      RML(IMEM,J)=0.0
      DO 5 I=1,12
   5  RML(IMEM,J)=RML(IMEM,J)+KMEM(IMEM,I,J)*DML(IMEM,I)
C
C     RMG - member load vectors in global coords
C     {p'}={Tt}{p}
C
      DO 9 J=1,12
      RMG(IMEM,J)=0.0
      DO 9 I=1,12
   9  RMG(IMEM,J)=RMG(IMEM,J)+TT(IMEM,I,J)*RML(IMEM,J)
C
C     Bending Moments in each member (at x=0, L/4, L/2,
C     3L/4, L)
C
      L6   =L(IMEM)/6.0
      UDL2=UDL(IMEM)/12.0
C
      DO 6 I=1,5
      C1=X(I)-L(IMEM)
      C2=6.0*X(I)*C1
      C3=L(IMEM)*L(IMEM)
      C4=C2+C3
      BMOM(IMEM,I)=UDL2*C4
      C5=RML(IMEM,11)+RML(IMEM,5)
      C6=X(I)/L(IMEM)
      C7=C5*C6
      C8=RML(IMEM,5)-C7
      BMOM(IMEM,I)=BMOM(IMEM,I)+C8
   6  WRITE(*,13) IMEM,I,BMOM,C8,C7,C6,C5,C4,C3,C2,C1
C  13 FORMAT(2(2X,I2),9(2X,G10.3))
C
C     Maximum total stresses in each member at each location
C
      YMAX=HW(ISNO)+TP(ISNO)+TF(ISNO)-YZZ(ISNO)
      IF(YMAX.LT.YZZ(ISNO)) YMAX=YZZ(ISNO)
```

```
      ZMIN=IZ(ISNO)/YMAX
      SXC=PML(IMEM,1)/AREA(ISNO)
      DO 7 I=1,5
      SXB=BMOM(IMEM,I)/ZMIN
      SX(IMEM,I)=SXB+SXC
C
C     Design criterion
C
      STRES=1.2*ABS(SX(IMEM,I))
      IF(STRES.GT.YS(ISNO)) IYLD(IMEM,I)=1
   7  CONTINUE
   8  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INPUT
C
       REAL IZ,IY,K,KMEM,KSS,KS,K11,K12,K21,K22,L,LX,LY,LZ,
C     &LENXZ,LEN1,NU

       COMMON/real/R(3,3),T(100,12,12),TT(100,12,12),K(12,12
      &),P(600),D(600),PJNT(100,6),PMG(100,12),KSS(600,600),
      &KS(600,600),XJC(100),YJC(100),X1(100),Y1(100),X2(100)
      &,Y2(100),K11(6,6),K12(6,6),K21(6,6),K22(6,6),ANG(100)
      &,L(100),PROD(12,12),RED(600,600),PML(100,12),BMOM(100
      &,5),SX(100,5),X(5),DMG(100,12),DML(100,12),KMEM(100,1
      &2,12),RML(100,12),RMG(100,12),PP(600),PJP(100,6),PFIX
      &(100,6),RS(600),DMEM(100,5),Z1(100),Z2(100),ZJC(100),
      &LX(100),LY(100),LZ(100),LENXZ(100),LEN1(100)
       COMMON/ints/MNO,JNO,ISSNO,ICLNO,LC,NUMSEC,ID(100,6),J
      &N(100),J1(100),J2(100),ISEC(100),IDLIN(600),JNUM(2),N
      &DOF,IOUT(100),IYLD(100,5),ISPAC
       COMMON/mats/ E(100),NU(100),B(100),TP(100),HW(100),TW
      &(100),BF(100),TF(100),AREA(100),IZ(100),IY(100),TORC(
      &100),YZZ(100),RADZ(100),ZZ(100),YYY(100),RADY(100),ZY
C     &(100),UDL(100),YS(100)
  90  FORMAT(20(/),30X,25('. '))
  91  FORMAT(30X,
     &'.   Linear Elastic Analysis of a 3-D Grillage   .')
  92  FORMAT(30X,
     &'.       using the Matrix Stiffness Method       .')
  93  FORMAT(30X,25('. '),35(/))
  94  FORMAT(30X,'.',47X,'.')
  10  FORMAT(6X,'No of members',2X,I3,6X,'No of
     &joints',2X,I3,6X,'No of section types',2X,I2,//)
  11  FORMAT(/,6X,'Joint No.',16X,'X Coord',20X,'Y Co
     &ord',20X,'Z Coord',//)
  12  FORMAT(/,6X,'Member No',4X,'Jnt No. at end 1',4X,
     &'Jnt No at end 2',4X,'Section No.',5X,'Length',14X,
     &'Angle',//)
  88  FORMAT(//,2X,'Sect No',6X,'E',10X,'nu',12X,'ys',12X,
     &'bp',13X,'tp',12X,'hw',12X,'tw',12X,'bf',12X,'tf'
     &,/,13X,'N/mm2',20X,'N/mm2',12X,'mm',12X,'mm',
     &12X,'mm',12X,'mm',12X,'mm',12X,'mm2',//)
  95  FORMAT(//,2X,'Sect No',4X,'Yzz',11X,'Izz',11X,'Zzz',
     &11X,'Rzz',11X,'Yyy',12X,'Iyy',10X,'Zyy',11X,'Ryy',
     &15X,'J',/,12X,' mm',12X,'mm4',11X,'mm3',11X,'mm',12X,
     &'mm',12X,'mm4',10X,'mm3',11X,'mm',14X,'mm4',//)
  89  FORMAT(5X,I2,10(2X,G12.5))
  14  FORMAT(25X,'Joint Loads',//,15X,'Joint
     &No.',10X,'Degree of freedom no. ',10X,'Load',//)
  37  FORMAT(//,25X,'Uniformly Distributed Loads',
     &//,15X,'Member No.',21X,'Load',//)
  15  FORMAT(34X,'Suppressions',//28X,'Joint No.',10X,
     &'d.o.f.',//)
  23  FORMAT(//,6X,'Member No.    d.o.f.     Applied
```

```
     &load',//)
  21  FORMAT(/,22X,'Member No ',I2,12X,'Uniformly distrib
     &uted load = ',G12.5,'  N/mm')
C
  26  FORMAT(8X,I3,19X,F8.2,19X,F8.2,19X,F8.2)
  27  FORMAT(10X,I2,15X,I2,18X,I2,14X,I2,9X,F9.3,9X,F9.3)
  28  FORMAT(5X,I3,6(4X,G12.5))
  29  FORMAT(//)
  35  FORMAT(32X,I2,15X,I2)
  36  FORMAT(18X,I2,23X,I2,18X,G12.5)
  38  FORMAT(18X,I2,23X,F8.3)
C
C     Screen output
C
C     WRITE(*,29)
C     WRITE(*,10) MNO,JNO,NUMSEC
C     WRITE(*,11)
C     WRITE(*,26) (I,XJC(I),YJC(I),ZJC(I),I=1,JNO)
C     WRITE(*,29)
C     WRITE(*,12)
C     WRITE(*,27) (I,J1(I),J2(I),ISEC(I),L(I),
C    &ANG(I),I=1,MNO)
C     WRITE(*,29)
C     WRITE(*,88)
C     DO 5 I=1,NUMSEC
C  5  WRITE(*,89) I,E(I),NU(I),YS(I),B(I),TP(I),
C    &HW(I),TW(I),BF(I),TF(I)
C     WRITE(*,95)
C     DO 6 I=1,NUMSEC
C  6  WRITE(*,89) I,YZZ(I),IZ(I),ZZ(I),RADZ(I),YYY(I),IY(I),
     &ZY(I),RADY(I),TORC(I)
C     WRITE(*,29)
C     WRITE(*,14)
C     DO 1 JNT=1,JNO
C     DO 1 IDOF=1,6
C  1  IF(PJNT(JNT,IDOF).NE.0.0) WRITE(*,36) JNT,IDOF,
     &PJNT(JNT,IDOF)
C     WRITE(*,29)
C     WRITE(*,15)
C     DO 2 JNT=1,JNO
C     DO 2 IDOF=1,6
C  2  IF(ID(JNT,IDOF).EQ.1) WRITE(*,35) JNT,IDOF
C
C     File output
C
      WRITE(6,90)
      WRITE(6,94)
      WRITE(6,94)
      WRITE(6,91)
      WRITE(6,94)
```

```
      WRITE(6,94)
      WRITE(6,92)
      WRITE(6,94)
      WRITE(6,94)
      WRITE(6,93)
      WRITE(6,29)
      WRITE(6,10) MNO,JNO,NUMSEC
      WRITE(6,11)
      WRITE(6,26) (I,XJC(I),YJC(I),ZJC(I),I=1,JNO)
      WRITE(6,29)
      WRITE(6,12)
      WRITE(6,27) (I,J1(I),J2(I),ISEC(I),L(I),
     &ANG(I),I=1,MNO)
      WRITE(6,29)
      WRITE(6,88)
      DO 8 I=1,NUMSEC
    8 WRITE(6,89) I,E(I),NU(I),YS(I),B(I),TP(I),HW(I),TW(I),
     &BF(I),TF(I)
      WRITE(6,95)
      DO 9 I=1,NUMSEC
    9 WRITE(6,89) I,YZZ(I),IZ(I),ZZ(I),RADZ(I),YYY(I),IY(I),
     &ZY(I),RADY(I),TORC(I)
      WRITE(6,29)
      WRITE(6,14)
      DO 3 JNT=1,JNO
      DO 3 IDOF=1,6
    3 IF(PJNT(JNT,IDOF).NE.0.0) WRITE(6,36) JNT,IDOF,
     &PJNT(JNT,IDOF)
      WRITE(6,37)
      DO 7 IMEM=1,MNO
    7 IF(UDL(IMEM).NE.0.0) WRITE(6,38) IMEM,UDL(IMEM)
      WRITE(6,29)
      WRITE(6,15)
      DO 4 JNT=1,JNO
      DO 4 IDOF=1,6
    4 IF(ID(JNT,IDOF).EQ.1) WRITE(6,35) JNT,IDOF
      RETURN
      END
```

```
      SUBROUTINE OUT
C
      REAL IZ,IY,K,KMEM,KSS,KS,K11,K12,K21,K22,L,LX,LY,LZ,
     &LENXZ,LEN1,NU
      COMMON/real/R(3,3),T(100,12,12),TT(100,12,12),K(12,12
      &),P(600),D(600),PJNT(100,6),PMG(100,12),KSS(600,600),
      &KS(600,600),XJC(100),YJC(100),X1(100),Y1(100),X2(100)
      &,Y2(100),K11(6,6),K12(6,6),K21(6,6),K22(6,6),ANG(100)
      &,L(100),PROD(12,12),RED(600,600),PML(100,12),BMOM(100
      &,5),SX(100,5),X(5),DMG(100,12),DML(100,12),KMEM(100,1
      &2,12),RML(100,12),RMG(100,12),PP(600),PJP(100,6),PFIX
      &(100,6),RS(600),DMEM(100,5),Z1(100),Z2(100),ZJC(100),
      &LX(100),LY(100),LZ(100),LENXZ(100),LEN1(100)
       COMMON/ints/MNO,JNO,ISSNO,ICLNO,LC,NUMSEC,ID(100,6),J
      &N(100),J1(100),J2(100),ISEC(100),IDLIN(600),JNUM(2),N
      &DOF,IOUT(100),IYLD(100,5),ISPAC
       COMMON/mats/ E(100),NU(100),B(100),TP(100),HW(100),TW
      &(100),BF(100),TF(100),AREA(100),IZ(100),IY(100),TORC(
      &100),YZZ(100),RADZ(100),ZZ(100),YYY(100),RADY(100),ZY&
      (100),UDL(100),YS(100)
C
   10 FORMAT(//,25X,'Structural stiffness matrix {K}
     &(={Tt}{k}{T}',//)
   11 FORMAT(//,25X,'Reduced structural stiffness matrix
     &{K}',//)
  12  FORMAT(//,25X,'Upper triangular form of the reduced'
     &'structural stiffness matrix {K}',//)
  20  FORMAT(/,5X,'Member No.',I2,6X,'{r',2H'},11X,
     &'{p',2H'},11X,'{r}',11X,'{p}',/)
  13  FORMAT(//,18X,'Displacements and support reactions
     &(global coords)',//,33X,'{D}',12X,'{P}',//)
  16  FORMAT(//,8X,'Member No. ',25X,'Bending Mo
     &ments',//,23X,'O',13X,'L/4',12X,'L/2',12X,
     &'3L/4',11X,'L',/)
  17  FORMAT(//,8X,'Member No.',20X,'Max total stresses
     &{s}',//,23X,'0',12X,'L/4',12X,'L/2',12X,
     &'3L/4',11X,'L',/)
  23  FORMAT(//,8X,'Member No.',20X,'Deflections along mem
     &ber length',//,23X,'0',12X,'L/4',12X,'L/2',12X,
     &'3L/4',11X,'L',/)
  24  FORMAT(//,8X,'Member No.',20X,'Member loads
     &{p',2H'},//,23X,'3',13X,'4',13X,'5',13X,'9',
     &13X,'10',12X,'11',/)
  21  FORMAT(/,11X,'Yielding is occurring in the following
     &members',//,10X,'No.',10X,'Member No.',11X,
     &'Longl bend. stress',/)
  14  FORMAT(7(2X,G12.5))
  15  FORMAT(15X,I3,10X,2(2X,G12.5))
  16  FORMAT(12(2X,G9.2))
  19  FORMAT(15X,I2,5(2X,G12.5))
```

```
  24  FORMAT(9X,I3,13X,I3,14X,G12.5)
  26  FORMAT(15X,I2,6(2X,G12.5))
C
C     WRITE(6,10)
C     WRITE(6,16) ((KSS(I,J),I=1,12),J=1,NDOF)
C     WRITE(6,10)
C     WRITE(6,16) ((KSS(I,J),I=13,24),J=1,NDOF)
C     WRITE(6,11)
C     WRITE(6,16) ((RED(I,J),I=1,12),J=1,NDOF)
C     WRITE(6,11)
C     WRITE(6,16) ((RED(I,J),I=13,24),J=1,NDOF)
C     WRITE(6,12)
C     WRITE(6,16) ((KS(I,J),I=1,12),J=1,NDOF)
C     WRITE(6,12)
C     WRITE(6,16) ((KS(I,J),I=13,24),J=1,NDOF)
C
C     WRITE(6,13)
C     DO 3 J=1,NDOF
C  3  WRITE(6,15) J,D(J),RS(J)
      DO 4 IMEM=1,MNO
      IF(IOUT(IMEM).EQ.0) GO TO 4
      WRITE(6,20) IMEM
      DO 5 IDOF=1,12
      WRITE(6,19) IDOF,DMG(IMEM,IDOF),RMG(IMEM,IDOF),
     &DML(IMEM,IDOF),RML(IMEM,IDOF)
   4  CONTINUE
      WRITE(6,17)
      DO 7 IMEM=1,MNO
   7  WRITE(6,19) IMEM,(BMOM(IMEM,I),I=1,5)
      WRITE(6,18)
      DO 6 IMEM=1,MNO
   6  WRITE(6,19) IMEM,(SX(IMEM,I),I=1,5)
      WRITE(6,23)
      DO 8 IMEM=1,MNO
   8  WRITE(6,19) IMEM,(DMEM(IMEM,I),I=1,5)
      WRITE(6,25)
      DO 21 IMEM=1,MNO
  22  WRITE(6,26) IMEM,RMG(IMEM,3),RMG(IMEM,4),
     &RMG(IMEM,5),RMG(IMEM,9),RMG(IMEM,10),RMG(IMEM,11)
      KOUNT=0
      WRITE(6,22)
      DO 9 IMEM=1,MNO
      DO 9 I=1,5
      IF(IYLD(IMEM,I).EQ.1) KOUNT=KOUNT+1
   9  IF(IYLD(IMEM,I).EQ.1) WRITE(6,24) KOUNT,IMEM,
     &SX(IMEM,I)
      RETURN
      END
```