Civil Engineering Study Structural Series 96-4

INRESB-3D-SUPII
Program Listing for Supercomputer

GENERAL PURPOSE PROGRAM FOR INELASTIC ANALYSIS OF RC AND STEEL BUILDING SYSTEMS FOR 3D STATIC AND DYNAMIC LOADS AND SEISMIC EXCITATIONS

by

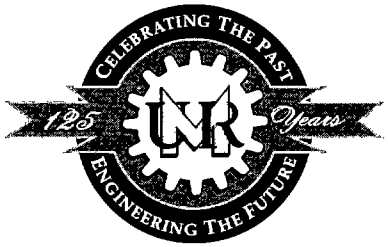
Franklin Y. Cheng
Curators' Professor

Jeng-Fuh Ger
Former Ph.D. Student and Post-Doctoral Fellow

Dan Li
Ph.D. Candidate

J.S. Yang
Former Ph.D. Student

Department of Civil Engineering
University of Missouri-Rolla
Rolla, MO 65409-0030



Report Series Prepared for the National Science Foundation
under Grants No. NSF BCS 9001494 and NSF MSS 9214664

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1.	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1996	Final

4. TITLE AND SUBTITLE INRESB-3D-SUPII Program Listing for Supercomputer: General Purpose Program for Inelastic Analysis of RC and Steel Building Systems for 3D Static and Dynamic Loads and Seismic Excitations

5. FUNDING NUMBERS
(G)
NSF BCS 9001494
NSF MSS 9214664

6. AUTHOR(S)
Franklin Y. Cheng, Jeng-Fuh Ger, Dan Li and J.S. Yang

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Civil Engineering
University of Missouri-Rolla
Rolla, MO 65409-0030

8. PERFORMING ORGANIZATION REPORT NUMBER
96-4

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Science Foundation
4201 Wilson Blvd.
Arlington, VA 22230

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

INRESB-3D-SUPII (supercomputer version) is written in FORTRAN and contains about 32,500 statements. It operates in the VM/SP CMS environment on an IBM 4381 computer or in the AIX/370 environment on an IBM 3090S supercomputer. INRESB-3D-SUPII is a general purpose computer program for analyzing elastic and inelastic building systems subject to static loads, dynamic forces, multicomponent earthquake motion, and pseudo-static cyclic loads. Also the program is capable of calculating inelastic post-buckling behavior of steel members and systems, natural frequency, and free vibration. A joint-based system is used to define the geometry of a structure. Structural members may be elastic 3D prismatic beams, nonlinear RC shear walls, nonlinear springs, inelastic 3D-beam-column elements, finite-segment elements, and nonlinear bracing elements. INRESB-3D-SUP II had five main functions. They are 1) elastic static analysis with multiple load cases based on control program SOL01; 2) elastic or inelastic analysis of a structure subject to 3D ground acceleration based on SOL02; 3) calculation of either natural frequencies and mode shape or buckling load and mode shape of an elastic structure based on SOL03; 4) calculation of nonlinear static cyclic response of a given loading pattern which consists of joint loads, imposed displacements, or element loads based on SOL04; and 5) calculation of maximum response of an elastic structure subject to pseudo-dynamic force obtained from the response spectrum based on SOL05. INRESB-3D-SUPII pertains to analysis of building systems above ground level. Any connection with basements is treated as a boundary condition fixed, hinged, or with a spring.

14. SUBJECT TERMS			15. NUMBER OF PAGES
Buckling	Ground acceleration	Natural frequency	114
Dynamic analysis	Inelastic analysis	Nonlinear material	16. PRICE CODE
Finite-element segment	Mode shape	Pseudo-dynamic force	
FORTRAN	Multicomponent earthquake motion	Response spectrum	

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

Civil Engineering Study Structural Series 96-4

INRESB-3D-SUPII
Program Listing for Supercomputer

GENERAL PURPOSE PROGRAM FOR INELASTIC ANALYSIS OF RC AND STEEL BUILDING SYSTEMS FOR 3D STATIC AND DYNAMIC LOADS AND SEISMIC EXCITATIONS

by

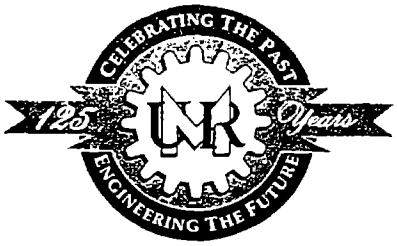
Franklin Y. Cheng
Curators' Professor

Jeng-Fuh Ger
Former Ph.D. Student and Post-Doctoral Fellow

Dan Li
Ph.D. Candidate

J.S. Yang
Former Ph.D. Student

Department of Civil Engineering
University of Missouri-Rolla
Rolla, MO 65409-0030



Report Series Prepared for the National Science Foundation
under Grants No. NSF BCS 9001494 and NSF MSS 9214664

ABSTRACT

INRESB-3D-SUPII (supercomputer version) is a general purpose program written in FORTRAN and containing about 32,500 statements. Neither the mathematics subroutine DG2CSP (an IMSL eigenvalue routine) nor subroutines SDUMP, CPUTIM, TIMEIT, TIME and DATE are included in this program. Mathematics library IMSL contains DG2CSP. Absence of the other five subroutines (SDUMP, CPUTIM, TIMEIT, TIME and DATE) does not affect operation of the program.

INRESB-3D-SUPII operates in the VM/SP CMS environment on an IBM 4381 computer or in the AIX/370 environment on an IBM 3090S supercomputer. Operation of this program in both the former and the latter consists of three steps. First, a macro library is generated that contains the common blocks. Second, the main program and each subroutine are compiled. Text files are stored in a disk on the user's account. Third, the program is executed. The first and second steps are done only once to install the program.

INRESB-3D-SUPII is suitable for analyzing elastic and inelastic building systems subject to static loads, dynamic forces, multi-component earthquake motion, and pseudo-static cyclic loads. Also, the program is capable of calculating inelastic post-buckling behavior of steel members and systems, natural frequency, and spectral response analysis. A joint-based system is used to define the geometry of a structure. Structural members may be elastic 3D prismatic beams, nonlinear reinforced concrete shear walls, nonlinear springs, inelastic 3D-beam-column elements, finite-segment elements, and nonlinear bracing elements.

INRESB-3D-SUPII has five main functions. They are 1) elastic static analysis with multiple load cases based on control program SOL01; 2) elastic or inelastic analysis of a structure subjected to multiple ground acceleration (one-, two-, or three-dimensional seismic input) based on SOL02; 3) calculation of either natural frequencies and mode shape or buckling load and mode shape for an elastic structure based on SOL03; 4) calculation of nonlinear static cyclic response for a given loading pattern which consists of joint loads, imposed displacements or element loads for based on SOL04; and 5) calculation of maximum response for an elastic

structure subject to pseudo-dynamic force obtained from the response spectrum based on SOL05.

INRESB-3D-SUPII pertains to analysis of building systems above ground level. Any connection with basements is treated as a boundary condition fixed, hinged, or with a spring.

Detail instruction of using the program is published in Civil Engineering Study Structural Series 96-3, INRESB-3D--SUPII, User's Manual.

ACKNOWLEDGMENTS

This is the second of two final reports on a research project sponsored by the National Science Foundation under grant numbers NSF BCS 9001494 and NSF MSS 9214664. This support is gratefully acknowledged. The authors gratefully acknowledge excellent service at University of Missouri Rolla computer center and the Cornell National Supercomputer Facility. The authors also gratefully acknowledge the advice and encouragement from Dr. S. C. Liu and Dr. K. P. Chong.

```
C------------ BEGIN COMMON ------------------------------------------- ZC000010
      INCLUDE (ZCOMN1)                                                  ZC000020
      INCLUDE (ZCOMN2)                                                  ZC000030
C------------ END COMMON --------------------------------------------- ZC000040

C------------ BEGIN BLOCK - ZCOMN1 ----------------------------------- ZC000010
      CHARACTER*112 TITLE(2)                                            ZC000020
      CHARACTER*40  STEPID                                              ZC000030
      LOGICAL BUG,MCOND,KGCOND,DOUBLE,NEWK,NEWKG,ELSTIC,ABORT          ZC000040
      INTEGER FMASS,FDAMP,CPUREM,CPUELA                                 ZC000050
      DOUBLE PRECISION SIE,SSE,PSE,EKE,EDD,EDUMMY                       ZC000060
      COMMON /ZDATAP/ TITLE ,BUG    ,IBUG  ,MAXEDF,MD(25),STEPID,GRAV   ZC000070
      COMMON /ZDATA/  IZ    ,IZREL ,CPUREM,CPUELA,ITLCPU,INCCPU,        ZC000080
     &  NEWK  ,KSAME ,NEWKG ,MCOND ,KGCOND,DOUBLE,ELSTIC,ABORT ,       ZC000090
     &  NNODE ,NDOF  ,NCOND ,NFREE ,NREST ,MAXNOD,NCOS  ,              ZC000100
     &  NMAT  ,NELMT ,NMASS ,                                          ZC000110
     &  NLOAD ,NSOLN ,                                                 ZC000120
     &  MAXELD,NELD  ,                                                 ZC000130
C                                                                      ZC000140
     &  IZID  ,IZIDOF,IZCORD,IZCOS ,IZCNST,IZJFLG,IZJCOS,              ZC000150
     &  IZMAT ,IZELE ,                                                 ZC000160
     &  IZIELD,IZELD ,                                                 ZC000170
     &  IZKE  ,IZLM  ,IZSTIF,IZMD  ,IZKEKG,IZLMKG,                     ZC000180
     &  IZMASS,IZMDMS,IZMDAT,IZDAMP,IZKGDT,IZKG  ,IZMDKG,              ZC000190
     &  IZLOAD,IZDISP,IZVEL ,IZACC ,                                   ZC000200
     &  IZDLOA,IZDDSP,IZDVEL,IZDACC,                                   ZC000210
     &  IZAGTX,IZAGTY,IZAGTZ,                                          ZC000220
     &  IZDUMP,IZFUB ,IZAN  ,IZAR  ,                                   ZC000230
C                                                                      ZC000240
     &  FMASS ,FDAMP,IUNIT,                                            ZC000250
     &  SIE,SSE,PSE,EKE,EDD,EDUMMY,                                    ZC000260
     &  SUMRCT(6),GROUND(9),INC,                                       ZC000270
     &  IDUMMY(500)                                                    ZC000280
C                                                                      ZC000290
C------------ END   BLOCK - ZCOMN1 ----------------------------------- ZC000300

C------------ BEGIN BLOCK - ZCOMN2 ----------C ZC000010
C                                            C ZC000020
      PARAMETER (MAXZ=500000, MAXDZ=MAXZ/2)    ZC000050
      COMMON /DATA/ Z(MAXZ)                    ZC000080
      DIMENSION NZ(MAXZ),DZ(MAXDZ)             ZC000090
      DOUBLE PRECISION DZ                      ZC000100
      EQUIVALENCE  (Z(1),NZ(1),DZ(1))          ZC000110
C                                            C ZC000120
C------------ END   BLOCK - ZCOMN2 ----------C ZC000130
.pf  :.ls normal
.sp 1 A
.us FILE: ZCOMN3 COPY
.sp 1 A
.bf p4  :.ls normal p5
C------------ BEGIN BLOCK - ZCOMN3 ----------------------------------- ZC000010
      DATA LZDATA/500/                                                  ZC000020
C------- VARIABLES IN COMMON                                           ZC000030
C TITLE  = USER INPUT TITLES DESCRIBING STRUCTURE AND SOLUTION         ZC000040
C IZ     = NEXT AVAILIABLE LOCATION IN LINEAR ARRAY Z                  ZC000050
C IZREL  = VALUE OF IZ IF STORAGE FOR A SOLUTION IS RELEASED           ZC000060
C BUG    = FLAG, IF TRUE, DETAILED OUTPUT IS PRINTED FOR DEBUGING      ZC000070
C MCOND  = FLAG, IF TRUE, MASS HAS 'NOT' BEEN CONDENSED OUT            ZC000080
C NNODE  = NUMBER OF NODES                                             ZC000090
C                                                                      ZC000100
C NDOF   = TOTAL NUMBER OF DEGREES OF FREEDOM                          ZC000110
C NCOND  = NUMBER OF DOF TO BE CONDENSED OUT                           ZC000120
C NFREE  = NUMBER OF FREE DOF                                          ZC000130
C NREST  = NUMBER OF RESTRAINED DOF                                    ZC000140
C MAXNOD = MAXIMUN NUMBER OF NODES INPUT                               ZC000150
C NMAT   = NUMBER OF MATERIAL PROPERTIES                               ZC000160
C NELMT  = NUMBER OF ELEMENTS                                          ZC000170
C                                                                      ZC000180
C NLCAD  = NUMBER OF LOADING CASES                                     ZC000190
C NSOLN  = SOLUTION NUMBER                                             ZC000200
C MAXEDF = MAXIMUN NUMBER OF ELEMENT DOF                               ZC000210
C MD     = MAIN DIAGONAL ADDRESSES FOR AN UPPER TRIANGULAR STIFFNESS   ZC000220
C NCOS   = NUMBER OF JOINT DIRECTION COSINES                           ZC000230
C MAXELD = MAXIMUN NUMBER OF ELEMENT LOADS                             ZC000240
C NELD   = NUMBER OF ELEMENT LOADS                                     ZC000250
C                                                                      ZC000260
C..... ALL VARIABLES BEGINING WITH IZ____ ARE THE ABSOLUTE ADDRESS     ZC000270
C..... IN THE LINEAR ARRAY FOR THE ____ DATA.                          ZC000280
C IZID   = EXTERNAL JOINT NUMBERS                                      ZC000290
C IZIDOF = DEGREE OF FREEDOM MATRIX FOR JOINT DOF'S                    ZC000300
C IZCORD = MATRIX OF JOINT COORD'S                                     ZC000310
C IZCOS  = MATRIX OF JOINT DIRECTION COSINES                           ZC000320
C IZCNST = MATIRX OF JOINT CONSTRAINT COIEFF                           ZC000330
C IZMAT  = MATERIAL DATA                                               ZC000340
C IZELE  = ELEMENT DATA                                                ZC000350
C                                                                      ZC000360
C IZJFLG = MATRIX OF JOINT DOF FLAGS (RELEASE CONSTRAIN CONDENSE ELIM.) ZC000370
C IZJCOS = MATRIX OF JOINT COSINE ID NUMBERS                           ZC000380
C IZKE   = ADDRESS OF ELEMENT STIFFNESS                                ZC000390
C IZLM   = ADDRESS OF ELEMENT GLOBAL TO LOCAL MAPPING MATRIX           ZC000400
C IZKEKG = ADDRESS OF ELEMENT GEOMETRIC STIFFNESS                      ZC000410
C IZLMKG = ADDRESS OF ELEMENT KG GLOBAL TO LOCAL MAPPING MATRIX        ZC000420
C IZSTIF = GLOBAL STIFFNESS                                            ZC000430
C IZMD   = ADDRESS OF MAIN DIAGONAL STIFFNESS TERMS                    ZC000440
C IZLOAD = LOAD MATRIX                                                 ZC000450
C IZDISP = DISPLACMENT                                                 ZC000460
C IZIELD = INTEGER ELEMENT LOAD DATA                                   ZC000470
C                                                                      ZC000480
C IZIELD = REAL ELEMENT LOAD DATA                                      ZC000490
C                                                                      ZC000500
C IZLOAD = TOTAL LOAD                                                  ZC000510
C IZDISP = TOTAL DISPLACEMENT                                          ZC000520
C IZVEL  = TOTAL VELOCTIY                                              ZC000530
C IZACC  = TOTAL ACCELERATION                                          ZC000540
C                                                                      ZC000550
C IZDLOA = INCRIMENTAL LOAD                                            ZC000560
C IZDDSP = INCRIMENTAL DISPLACEMENT                                    ZC000570
C IZDVEL = INCRIMENTAL VELOCTIY                                        ZC000580
C IZDACC = INCRIMENTAL ACCELERATION                                    ZC000590
C                                                                      ZC000600
C IZAGTX = TRANSLATIONAL GROUND ACCELERATION - X DIRECTION             ZC000610
C IZAGTY = TRANSLATIONAL GROUND ACCELERATION - Y DIRECTION             ZC000620
C IZAGTZ = TRANSLATIONAL GROUND ACCELERATION - Z DIRECTION             ZC000630
C                                                                      ZC000640
C IZFUB  = UNBALANCED FORCE VECTOR                                     ZC000650
C IZQ    = TEMPORARY VECTOR                                            ZC000660
C IZR    = TEMPORARY VECTOR                                            ZC000670
C                                                                      ZC000680
C FDAMP  = DAMPING TYPE                                                ZC000690
C FMASS  = MASS TYPE                                                   ZC000700
C IUNIT  = UNIT # FOR PRINTING DATA TO AN OUTPUT FILE                  ZC000710
C                                                                      ZC000720
C------------ END   COMMON BLOCK - ZCOMN ----------------------------- ZC000730

C=======================================================================CFEM00010
C                                                                      CFEM00020
C   PROGRAM FEM - ANALYSIS OF STRUCTURES                               CFEM00030
C                                                                      CFEM00040
C=======================================================================CFEM00050
C                                                                      CFEM00060
C   NOTE: USE IBM COMPILER OPTION AUTODBL(DBLPAD4)                     CFEM00070
C         TO COMPILE THIS PROGRAM IN DOUBLE PRECISION                  CFEM00080
C                                                                      CFEM00090
C   NOTE: USE IBM COMPILER OPTION OPT(2)                               CFEM00100
C         TO OPTIMIZE THE CODE FOR FASTER EXECUTION                    CFEM00110
C                                                                      CFEM00120
C=======================================================================CFEM00130
      CHARACTER*80 OPTION,INPUT,BLANK                                   FEM00140
      CHARACTER*8  TIME1,DATE1                                          FEM00150
      CHARACTER*6  PLACE                                                FEM00160
      CHARACTER*1  CHR(0:25), NAME                                      FEM00170
      LOGICAL TEST,BTEST,HEAD ,AXIAL                                    FEM00180
      INCLUDE (ZCOMN)                                                   FEM00190
      INCLUDE (ZCOMN3)                                                  FEM00200
      DATA CHR/'A','B','C','D','E','F','G','H','I','J','K','L','M',    FEM00210
     &         'N','O','P','Q','R','S','T','U','V','W','Z','Y','Z'/    FEM00220
C                                                                       FEM00230
C=======================================                                FEM00240
C== INITILIZE THE UMRBX TIMING SUBROUTINE ==                            FEM00250
C=======================================                                FEM00260
      CPUREM=0                                                          FEM00270
      CPUELA=0                                                          FEM00280
      CALL CPUTIM(CPUREM,CPUELA)                                        FEM00290
      ITLCPU=CPUREM                                                     FEM00300
C                                                                       FEM00310
C=============================                                         FEM00320
C== INITILIZE THE CPU CLOCK ===                                         FEM00330
C=============================                                         FEM00340
      CALL TIMEON                                                       FEM00350
      TCPU=0                                                            FEM00360
C                                                                       FEM00370
C=============================                                         FEM00380
C== ECHO THE INPUT.....   ==                                           FEM00390
C=============================                                         FEM00400
      I=55                                                              FEM00410
      L=0                                                               FEM00420
9995  READ (5,('(A)'),END=9996) INPUT                                   FEM00430
      IF (INDEX(INPUT,'NOECHO').NE.0) GO TO 9996                        FEM00440
      I=I+1                                                             FEM00450
      L=L+1                                                             FEM00460
      IF (  I.GT.55 ) THEN                                              FEM00470
         WRITE (6,9994) (KK,KK=1,8)                                     FEM00480
         I=0                                                            FEM00490
      ENDIF                                                             FEM00500
      WRITE (6,9997) L,INPUT                                            FEM00510
      IF (INDEX(INPUT,'STOP'  ).EQ.0) GO TO 9995                        FEM00520
9994  FORMAT ('1    ECHO OF INPUT DATA '//                              FEM00530
     &        1X,3X,'LINE  ',8('....|...',I1,'0'))                      FEM00540
9997  FORMAT (1X,I7,'| ',A)                                             FEM00550
9996  REWIND(5)                                                         FEM00560
C                                                                       FEM00570
C=============================                                         FEM00580
C== INITIALIZE DATA.....   ==                                           FEM00590
C=============================                                         FEM00600
      TITLE(1)=' '                                                      FEM00610
      TITLE(2)=' '                                                      FEM00620
      BLANK  =' '                                                       FEM00630
      BUG =.FALSE.                                                      FEM00640
      DOUBLE =.FALSE.                                                   FEM00650
      IBUG=0                                                            FEM00660
      MAXEDF=24                                                         FEM00670
      MD(1)=1                                                           FEM00680
      DO 1 I=2,(MAXEDF+1)                                               FEM00690
    1   MD(I)=MD(I-1)+(I-1)                                             FEM00700
      IZ=0                                                              FEM00710
C--------- INITILIZE MEMORY -----                                       FEM00720
C     DO 10 I=1,MAXZ                                                   FEM00730
C  10 Z(I)=0                                                            FEM00740
C-------------------------------                                       FEM00750
   20 READ(5,*,END=1000) OPTION                                         FEM00760
      RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                    FEM00770
C                                                                       FEM00780
C=============================                                         FEM00790
C== DEFINE JOB RUNNING PLACE ===                                        FEM00800
C=============================                                         FEM00810
      IF (INDEX(OPTION,'UMR').NE.0)  THEN                               FEM00820
          PLACE = 'UMRVMB'                                              FEM00830
          READ(5,*,END=1000) REQCPU                                     FEM00840
      ELSE IF (INDEX(OPTION,'COR').NE.0)  THEN                          FEM00850
          PLACE = 'CORNEL'                                              FEM00860
          READ(5,*,END=1000) REQCPU                                     FEM00870
C                                                                       FEM00880
C=============================                                         FEM00890
C== DEFINE THE STRUCTURE ===                                            FEM00900
C=============================                                         FEM00910
      ELSE IF (INDEX(OPTION,'STR').NE.0)  THEN                          FEM00920
          ABORT=.FALSE.                                                 FEM00930
          CALL  COMPT(1,LZDATA)                                         FEM00940
          GRAV=386.4                                                    FEM00950
          READ (5,*,END=1000) TITLE(1)                                  FEM00960
          CALL TIME(TIME1)                                              FEM00970
          CALL DATE(DATE1)                                              FEM00980
          WRITE (TITLE(1)(80:),45) TIME1,DATE1                          FEM00990
          RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                FEM01000
          WRITE (1,52)' BULIDING STRUCTURAL MODEL ----------*',FEM01010
     &                    IZ,RATIZ                                      FEM01020
          WRITE (1,51) TITLE(1)(1:70)                                   FEM01030
          ABORT=.FALSE.                                                 FEM01040
          CALL STRUCT                                                   FEM01050
          IZREL=IZ                                                      FEM01060
          RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                FEM01070
          CALL TIMEIT(CPU)                                              FEM01080
          TCPU=CPU+TCPU                                                 FEM01090
          WRITE (6,53) IZ,RATIZ,CPU,TCPU                                FEM01100
C                                                                       FEM01110
C=================================                                     FEM01120
C== DEFINE THE LOADING AND SOLVE ==                                     FEM01130
C=================================                                     FEM01140
      ELSE IF (INDEX(OPTION,'SOL').NE.0)  THEN                          FEM01150
          READ (5,*,END=1000) TITLE(2)                                  FEM01160
          CALL TIME(TIME1)                                              FEM01170
          CALL DATE(DATE1)                                              FEM01180
          WRITE (TITLE(2)(80:),45) TIME1,DATE1                          FEM01190
C         WRITE (6,50) '1',TITLE(1),TITLE(2)                            FEM01200
          WRITE (1,52)' SOLVING STRUCTURAL MODEL -----------*',FEM01210
     &                    IZ,RATIZ                                      FEM01220
          WRITE (1,51) TITLE(1)(1:70)                                   FEM01230
          WRITE (1,51) TITLE(2)(1:70)                                   FEM01240
          ABORT=.FALSE.                                                 FEM01250
C         CALL SOLN(OPTION)                                             FEM01260
          CALL SOLN(OPTION,PLACE,CPU,TCPU,REQCPU)                       FEM01270
          RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                FEM01280
          CALL TIMEIT(CPU)                                              FEM01290
          TCPU=CPU+TCPU                                                 FEM01300
          WRITE (6,53) IZ,RATIZ,CPU,TCPU                                FEM01310
          IF(ABORT) GO TO 1000                                          FEM01320
C                                                                       FEM01330
C=================================                                     FEM01340
C== DUMP THE MEMORARY...        ==                                     FEM01350
C=================================                                     FEM01360
      ELSE IF (INDEX(OPTION,'DUMP').NE.0)  THEN                         FEM01370
          WRITE (1,52)' DUMPING MEMORY ---------------------*',FEM01380
     &                    IZ,RATIZ                                      FEM01390
          WRITE (1,51) TITLE(1)(1:70)                                   FEM01400
          WRITE (1,51) TITLE(2)(1:70)                                   FEM01410
          CALL COMDMP(Z,NZ,DZ)                                          FEM01420
          RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                FEM01430
          CALL TIMEIT(CPU)                                              FEM01440
          TCPU=CPU+TCPU                                                 FEM01450
          WRITE (6,53) IZ,RATIZ,CPU,TCPU                                FEM01460
C                                                                       FEM01470
C=================================                                     FEM01480
C== STOP... END OF PROGRAM      ==                                     FEM01490
C=================================                                     FEM01500
      ELSE IF (INDEX(OPTION,'STOP').NE.0)  THEN                         FEM01510
          WRITE (1,52)' NORMAL STOP ------------------------*',FEM01520
     &                    IZ,RATIZ                                      FEM01530
          STOP                                                          FEM01540
C                                                                       FEM01550
C=================================                                     FEM01560
C==  READ RESULTS FROM INPUT FILE ==                                    FEM01570
C=================================                                     FEM01580
```

```
      ELSE IF (INDEX(OPTION,'READ').NE.0)  THEN                         FEM01610
  140   CALL GETINT(OPTION,'UNIT=',IUNIT,0,.TRUE.,.TRUE. )               FEM01620
        CALL GETINT(OPTION, 'INC=',INC  ,1,.TRUE.,.FALSE.)               FEM01630
        IF (IUNIT.GT.0) THEN                                             FEM01640
          WRITE (1,52)                                                   FEM01650
     &     ' READING DATA FROM FILE ----------------------*',IZ,RATIZ   FEM01660
          IOPT=4                                                         FEM01670
          CALL DMPDAT(IOPT,IWRITE,TO,DT)                                 FEM01680
          GO TO 140                                                      FEM01690
        ENDIF                                                            FEM01700
        RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                   FEM01710
        CALL TIMEIT(CPU)                                                 FEM01720
        TCPU=CPU+TCPU                                                    FEM01730
        WRITE (6,53) IZ,RATIZ,CPU,TCPU                                   FEM01740
C                                                                        FEM01750
C------------------------                                                FEM01760
C-- SET BUG OPTION     --                                                FEM01770
C------------------------                                                FEM01780
      ELSE IF (TEST(OPTION,'BUG=',.FALSE.))  THEN                        FEM01790
        CALL GETCHR(OPTION,'BUG=',INPUT,.TRUE.,.FALSE.)                  FEM01800
        IBUG=0                                                           FEM01810
        WRITE (OPTION,160) INPUT(1:70)                                   FEM01820
        DO 150 I=0,23                                                    FEM01830
          IF (TEST(INPUT,CHR(I),.FALSE.)) THEN                           FEM01840
            IBUG=IBSET(IBUG,I)                                           FEM01850
          ENDIF                                                          FEM01860
  150   CONTINUE                                                         FEM01870
  160   FORMAT ('BUG=',A)                                                FEM01880
        WRITE (1,52)' SET BUG OPTION: ----------------------------',FEM01890
     &              IZ,RATIZ                                            FEM01900
        WRITE (1,51)  OPTION(1:70)                                       FEM01910
        WRITE (6,51)  'SET BUG OPTION: '                                 FEM01920
        WRITE (6,51)  OPTION(1:70)                                       FEM01930
C                                                                        FEM01940
C------------------------                                                FEM01950
C-- SET BUG OPTION     --                                                FEM01960
C------------------------                                                FEM01970
      ELSE IF (INDEX(OPTION,'BUG').NE.0)  THEN                           FEM01980
        WRITE (1,51) ('SET BUG OPTION: '//OPTION(1:54) )                 FEM01990
        BUG=.TRUE.                                                       FEM02000
        IF (INDEX(OPTION,'NOBUG').NE.0) BUG=.FALSE.                      FEM02010
C                                                                        FEM02020
C---------------------------------------                                 FEM02030
C-- RELEASE MEMORY FROM LAST SOLUTION --                                 FEM02040
C---------------------------------------                                 FEM02050
      ELSE IF (INDEX(OPTION,'RELEASE').NE.0)  THEN                       FEM02060
        WRITE (6,52)'-----------------------------------------------'    FEM02070
        WRITE (1,52)' RELEASING MEMORY ----------------------------'     FEM02080
        WRITE (6,52)' RELEASING MEMORY ----------------------------'     FEM02090
        IF (INDEX(OPTION,'ELEMENT').NE.0)  THEN                          FEM02100
          WRITE (6,52)' RESETTING ELEMENT FORCES ----------------------' FEM02110
          WRITE (6,52)'-----------------------------------------------'  FEM02120
          DO 170 IELNO=1,NELMT                                           FEM02130
            LSTTYP=0                                                     FEM02140
            HEAD =.FALSE.                                                FEM02150
  170       CALL ELELIB                                                  FEM02160
     &       (13   ,LSTTYP ,.FALSE.,IREL ,HEAD  ,INPUT,EESE,EPSE,DAMAGE,FEM02170
     &        DUCFAG,                                                    FEM02180
     &        IELNO,IELDOF,KGDOF,PGEOM,Z ,AXIAL,IZDDSP,IZDLOA,MSTOR)     FEM02190
        ELSE                                                             FEM02200
          WRITE (6,52)'-----------------------------------------------'  FEM02210
        ENDIF                                                            FEM02220
        IZ=IZREL                                                         FEM02230
          IZACC =0                                                       FEM02240
          IZDACC=0                                                       FEM02250
          IZDDSP=0                                                       FEM02260
          IZDISP=0                                                       FEM02270
          IZDLOA=0                                                       FEM02280
          IZDVEL=0                                                       FEM02290
          IZFUB =0                                                       FEM02300
          IZLOAD=0                                                       FEM02310
          IZVEL =0                                                       FEM02320
          RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                 FEM02330
          CALL TIMEIT(CPU)                                               FEM02340
          TCPU=CPU+TCPU                                                  FEM02350
          WRITE (6,53) IZ,RATIZ,CPU,TCPU                                 FEM02360
C                                                                        FEM02370
C---------------------------------------                                 FEM02380
C-- SAVE MEMORY FOR A RESTART          --                                FEM02390
C---------------------------------------                                 FEM02400
      ELSE IF (INDEX(OPTION,'SAVE').NE.0)  THEN                          FEM02410
        WRITE (1,52)' SAVING DATA  ------------------------------',FEM02420
     &              IZ,RATIZ                                            FEM02430
        WRITE (1,51) TITLE(1)(1:70)                                      FEM02440
        WRITE (1,51) TITLE(2)(1:70)                                      FEM02450
        WRITE (6,200)                                                    FEM02460
  200   FORMAT(5X,'******** WRITING DATA TO UNIT FT09 ********')         FEM02470
        OPEN(UNIT=9,ACCESS='SEQUENTIAL',STATUS='UNKNOWN',                FEM02480
     &       FORM='UNFORMATTED')                                         FEM02490
        WRITE (9)   TITLE                                                FEM02500
        WRITE (9)   IZ    ,IZREL ,                                       FEM02510
     &    NDOF  ,NCOND ,NFREE ,NREST ,MAXNOD,NMAT  ,NELMT ,NNODE ,      FEM02520
     &    NLOAD ,NSOLN ,MAXEDF,MD,NCOS,MAXELD,NELD,                     FEM02530
     &    IZID  ,IZIDOF,IZCORD,IZCOS ,IZCNST,IZMAT ,IZELE ,             FEM02540
     &    IZJFLG,IZJCOS,                                                FEM02550
     &    IZKE  ,IZLM  ,IZSTIF,IZMD  ,IZLOAD,IZDISP,IZIELD,IZELD        FEM02560
        DO 210 J=1,IZ,256                                                FEM02570
          K=J+255                                                        FEM02580
          K=MIN(K,IZ)                                                    FEM02590
  210     WRITE (9) (Z(I),I=J,K)                                         FEM02600
        CLOSE (UNIT=9)                                                   FEM02610
        RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                   FEM02620
        CALL TIMEIT(CPU)                                                 FEM02630
        TCPU=CPU+TCPU                                                    FEM02640
        WRITE (6,53) IZ,RATIZ,CPU,TCPU                                   FEM02650
C                                                                        FEM02660
C---------------------------------------                                 FEM02670
C-- RESTART PROGRAM                    --                                FEM02680
C---------------------------------------                                 FEM02690
        WRITE (1,52)' RETREVING DATA ---------------------------',FEM02700
     &              IZ,RATIZ                                            FEM02710
      ELSE IF (INDEX(OPTION,'RESTART').NE.0)  THEN                       FEM02720
        WRITE (6,300)                                                    FEM02730
  300   FORMAT(5X,'******** READING DATA FROM UNIT FT09 ********')       FEM02740
        OPEN(UNIT=9,ACCESS='SEQUENTIAL',STATUS='OLD',                    FEM02750
     &       FORM='UNFORMATTED')                                         FEM02760
        READ  (9)   TITLE                                                FEM02770
        READ  (9)   IZ    ,IZREL ,                                       FEM02780
     &    NDOF  ,NCOND ,NFREE ,NREST ,MAXNOD,NMAT  ,NELMT ,NNODE ,      FEM02790
     &    NLOAD ,NSOLN ,MAXEDF,MD,NCOS,MAXELD,NELD,                     FEM02800
     &    IZID  ,IZIDOF,IZCORD,IZCOS ,IZCNST,IZMAT ,IZELE ,             FEM02810
     &    IZJFLG,IZJCOS,                                                FEM02820
     &    IZKE  ,IZLM  ,IZSTIF,IZMD  ,IZLOAD,IZDISP,IZIELD,IZELD        FEM02830
        DO 310 J=1,IZ,256                                                FEM02840
          K=J+255                                                        FEM02850
          K=MIN(K,IZ)                                                    FEM02860
  310     READ  (9) (Z(I),I=J,K)                                         FEM02870
        CLOSE (UNIT=9)                                                   FEM02880
C       WRITE (6,50) ' ',TITLE(1),TITLE(2)                               FEM02890
        WRITE (1,52)' DATA RETREIVED ---------------------------',FEM02900
     &              IZ,RATIZ                                            FEM02910
        WRITE (1,51) TITLE(1)(1:70)                                      FEM02920
        WRITE (1,51) TITLE(2)(1:70)                                      FEM02930
        RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                   FEM02940
        CALL TIMEIT(CPU)                                                 FEM02950
        TCPU=CPU+TCPU                                                    FEM02960
        WRITE (6,53) IZ,RATIZ,CPU,TCPU                                   FEM02970
C                                                                        FEM02980
C---------------------------------------                                 FEM02990
C-- DUMMY STATEMENTS DURING ABORT      --                                FEM03000
C---------------------------------------                                 FEM03010
      ELSE IF (ABORT) THEN                                               FEM03020
          GO TO  20                                                      FEM03030
C                                                                        FEM03040
C---------------------------------------                                 FEM03050
C-- INVALID OPTION                     --                                FEM03060
C---------------------------------------                                 FEM03070
      ELSE                                                               FEM03080
          IF (TEST(OPTION,'NOECHO',.FALSE.) ) GO TO 20                   FEM03090
          IF (OPTION.EQ.BLANK) GO TO 20                                  FEM03100
          WRITE (6,30) OPTION                                            FEM03110
          WRITE (1,51) ('INVALID OPTION: '//OPTION(1:54))                FEM03120
      ENDIF                                                              FEM03130
      GO TO  20                                                          FEM03140
C                                                                        FEM03150
C                                                                        FEM03160
   45 FORMAT(' TIME: ',A,', DATE: ',A)                                   FEM03170
C  45 FORMAT(' CREATED AT ',A,' ON ',A)                                  FEM03180
   50 FORMAT (A1,' STRUCTURE....: ',A,/                                  FEM03190
     &        1X,' SOLUTION.....: ',A,/)                                 FEM03200
   51 FORMAT (1X,A)                                                      FEM03210
   52 FORMAT (1X,'*---',A45,'*',1,' IZ=',I7,' MEM=',F6.3,'%')            FEM03220
   53 FORMAT  (//                                                        FEM03230
     &  1X,'*----',35('-'),'*'/                                          FEM03240
     &  1X,'*--- MEMORY UTILIZATION ..........  ----*'/                  FEM03250
     &  1X,'*--- IZ=', I7,',    MEM=',F6.3,'%    ----*'/                FEM03260
     &  1X,'*----',35('-'),'*'/                                          FEM03270
     &  1X,'*--- ELAPSED CPU TIME ',F8.2,' SEC ----*'/                   FEM03280
     &  1X,'*--- TOTAL CPU TIME   ',F8.2,' SEC ----*'/                   FEM03290
     &  1X,'*----',35('-'),'*'/ )                                        FEM03300
   30 FORMAT ('1 INVALID OPTION= ',A)                                    FEM03310
  110 FORMAT (' $MATRX(',I6,')=',I15,1P,G20.10,:,G20.10)                 FEM03320
 1000 STOP                                                               FEM03330
      END                                                                FEM03340
C***********************************************************************AVE00010
C       DEBUG UNIT(6),SUBCHK,SUBTRACE                                    AVE00020
C       END DEBUG                                                        AVE00030
      SUBROUTINE AVEACC(IOPT,BUG,DT,TO,TF,T,THETA,NEWKDT,                AVE00040
     &    L1,L2,L3,L4,L5,L6,IMDMAS,IMD,IMDS,IMDKG,NDOF,NFREE,            AVE00050
     &    KGLOAD,KGFORM,                                                 AVE00060
     &    A     ,V     ,D     ,P     ,                                  AVE00070
     &    DA    ,DV    ,DD    ,DP    ,WORK,                             AVE00080
     &    MASS  ,MDMASS,DAMP  ,STIFF ,MD    ,                          AVE00090
     &    KG    ,MDKG  ,ACC   ,GRAV  ,                                   AVE00100
     &    S     ,MDS   ,Q     ,R     ,PBAR  ,                          AVE00110
     &    FDAMP,FMASS,C0,C1,C2,C3,C4,C5,C6,ALPHA,BETA,C10,C11,ABORT)     AVE00120
      INTEGER FDAMP,FMASS                                                AVE00130
      REAL MASS,KG                                                       AVE00140
      LOGICAL BUG,NEWKDT,ABORT                                           AVE00150
      LOGICAL BTEST                                                      AVE00160
      DIMENSION A(NDOF),V(NDOF),D(NDOF),DA(NDOF),DV(NDOF),DD(NDOF)       AVE00170
      DIMENSION P(NDOF),DP(NDOF),WORK(2*NDOF)                            AVE00180
      DIMENSION PBAR(L3:L4),Q(L3:L4),R(L3:L4)                            AVE00190
      DIMENSION MDMASS(NDOF+1),MASS(IMDMAS)                              AVE00200
      DIMENSION MDKG(NDOF+1)  ,KG(IMDKG)                                 AVE00210
      DIMENSION MD(NDOF+1)    ,STIFF(IMD)                                AVE00220
      DIMENSION MDS(NDOF+1)   ,S(IMDS)     ,DAMP(3)                      AVE00230
C                                                                        AVE00240
C-----------------------------------------------------------------------AVE00250
C     DETERMINE THE RESPONSE AT TIME DT*2                                AVE00260
C                                                                        AVE00270
      TWODT=DT*2.0                                                       AVE00280
      THETA=2.0                                                          AVE00290
      CALL LINACC(IOPT,BUG,TWODT,TO,TF,T,THETA,NEWKDT,                   AVE00300
     &    L1,L2,L3,L4,L5,L6,IMDMAS,IMD,IMDS,IMDKG,NDOF,NFREE,            AVE00310
     &    KGLOAD,KGFORM,                                                 AVE00320
     &    A     ,V     ,D     ,P     ,                                  AVE00330
     &    DA    ,DV    ,DD    ,DP    ,WORK,                             AVE00340
     &    MASS  ,MDMASS,DAMP  ,STIFF ,MD    ,                          AVE00350
     &    KG    ,MDKG  ,ACC   ,GRAV  ,                                   AVE00360
     &    S     ,MDS   ,Q     ,R     ,PBAR  ,                          AVE00370
     &    FDAMP,FMASS,C0,C1,C2,C3,C4,C5,C6,ALPHA,BETA,C10,C11,ABORT)     AVE00380
C                                                                        AVE00390
      IF (IOPT.NE.2) RETURN                                              AVE00400
C                                                                        AVE00410
C------- DETERMINE ACCELERATION, VELOCITY AND DISPLACEMENT AT TIME DT    AVE00420
C        DA = INCRIMENTAL ACCELERATION BETWEEN TIMES T AND T+DT          AVE00430
C        DV = INCRIMENTAL VELOCTIY BETWEEN TIMES T AND T+DT              AVE00440
C        DD = INCRIMENTAL DISPLACMENT BETWEEN TIMES T AND T+DT           AVE00450
C                                                                        AVE00460
      DO 270 I=L3,L4                                                     AVE00470
        DSP2DT=DD(I)                                                     AVE00480
        VEL2DT=DV(I)                                                     AVE00490
        ACC2DT=DA(I)                                                     AVE00500
        IF (BUG) WRITE (6,271) I,DSP2DT,VEL2DT,ACC2DT                    AVE00510
        DA(I)= ACC2DT/2.00                                               AVE00520
        DV(I)= (4*A(I) + ACC2DT)*DT/4.                                   AVE00530
  270   DD(I)=V(I)*DT + (6.00*A(I) + ACC2DT)*DT*DT/12.                   AVE00540
  271   FORMAT ('  DOF#:',I6,' AT T+2DT,  DSP:',1P,G14.6,'  VEL:',G14.6, AVE00550
     &          '  ACC:',G14.6)                                          AVE00560
C                                                                        AVE00570
      RETURN                                                             AVE00580
C                                                                        AVE00590
      END                                                                AVE00600
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP                                   BML00010
C***********************************************************************BML00020
C       DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                         BML00030
C       END DEBUG                                                        BML00040
C-----------------------------------------------------------------------BML00050
      SUBROUTINE BMLOA(IELNO,FLAG,L,REL,MAXELD,NELD,NLOAD,               BML00060
     &    FEM,IELD,ELD)                                                  BML00070
C     INTEGER RT,RS                                                      BML00080
      REAL    L,MA,MB                                                    BML00090
      INTEGER REL                                                        BML00100
      LOGICAL B0,B1,B2,B3,B4,B5,BTEST                                    BML00110
      LOGICAL TEST,FLAG                                                  BML00120
      DIMENSION IELD(5,MAXELD),ELD(12,MAXELD)                            BML00130
      DIMENSION FEM(12,NLOAD),FORCE(12)                                  BML00140
      EQUIVALENCE (FORCE( 1),F1 ),(FORCE( 2),F2 ),(FORCE( 3),F3 )        BML00150
      EQUIVALENCE (FORCE( 4),F4 ),(FORCE( 5),F5 ),(FORCE( 6),F6 )        BML00160
      EQUIVALENCE (FORCE( 7),F7 ),(FORCE( 8),F8 ),(FORCE( 9),F9 )        BML00170
      EQUIVALENCE (FORCE(10),F10),(FORCE(11),F11),(FORCE(12),F12)        BML00180
      FLAG=.FALSE.                                                       BML00190
      L=L                                                                BML00200
      REL=REL                                                            BML00210
      B0=BTEST(REL,0)                                                    BML00220
      B1=BTEST(REL,1)                                                    BML00230
      B2=BTEST(REL,2)                                                    BML00240
      B3=BTEST(REL,3)                                                    BML00250
      B4=BTEST(REL,4)                                                    BML00260
      B5=BTEST(REL,5)                                                    BML00270
C                                                                        BML00280
      IF (MAXELD.LE.0) RETURN                                            BML00290
C                                                                        BML00300
C------- LOOP FOR EACH LOAD INPUT                                        BML00310
      DO 1000 N=1,NELD                                                   BML00320
C                                                                        BML00330
C....... DETERMINE IF MEMBER IS AFFECTED BY THIS LOAD                    BML00340
        TEST=.FALSE.                                                     BML00350
        IEL=IELD(2,N)                                                    BML00360
        IGEN=IELD(3,N)                                                   BML00370
        INC=IELD(4,N)                                                    BML00380
        IF (IEL.EQ.IELNO) TEST=.TRUE.                                    BML00390
        IF ((.NOT.TEST) .AND. IGEN.GT.0 .AND. INC.NE.0) THEN             BML00400
          IRMD=MOD(IELNO-IEL,INC)                                        BML00410
          JGEN=(IELNO-IEL)/INC                                           BML00420
          IF (IRMD.EQ.0 .AND. JGEN.LE.IGEN .AND. JGEN.GE.0)              BML00430
     &      TEST=.TRUE.                                                  BML00440
        ENDIF                                                            BML00450
        IF (.NOT. TEST) GO TO 1000                                       BML00460
C                                                                        BML00470
C.................. ZERO FEM                                            BML00480
        IF (.NOT. FLAG) THEN                                             BML00490
          DO 10 I=1,12                                                   BML00500
```

```
            DO 10 J=1,NLOAD                                          BMLO0510
   10         FEM(I,J)=0                                             BMLO0520
              FLAG=.TRUE.                                            BMLO0530
            ENDIF                                                    BMLO0540
C                                                                    BMLO0550
C.................. ZERO FORCE                                       BMLO0560
            DO 15 I=1,12                                             BMLO0570
   15         FORCE(I)=0                                             BMLO0580
C                                                                    BMLO0590
C....... DETERMINE LOAD CODES...                                     BMLO0600
            KODE=IELD(5,N)                                           BMLO0610
            KDOF=MOD(KODE,100)                                       BMLO0620
            KGRP=KODE-KDOF                                           BMLO0630
            LOAD=IELD(1,N)                                           BMLO0640
C                                                                    BMLO0650
C                                                                    BMLO0660
C....... CONCENTRATED LOADING                                        BMLO0670
          IF (KGRP.EQ.100) THEN                                      BMLO0680
            P=ELD(1,N)                                               BMLO0690
            R=ELD(2,N)                                               BMLO0700
            IF (KDOF.EQ.1) THEN                                      BMLO0710
              F1=-(1-R)*P                                            BMLO0720
              F7=-R*P                                                BMLO0730
            ELSE IF (KDOF.EQ.2) THEN                                 BMLO0740
              F6 =-P*R*L*(1-R)**2                                    BMLO0750
              F12= P*(1-R)*L*(R)**2                                  BMLO0760
              CALL FEMREL(F6 ,F12,B3 ,B0 ,0,IELNO)                   BMLO0770
              F8=(-F6-F12-P*R*L )/L                                  BMLO0780
              F2=-P-F8                                               BMLO0790
            ELSE IF (KDOF.EQ.3) THEN                                 BMLO0800
              F5 = P*R*L*(1-R)**2                                    BMLO0810
              F11=-P*(1-R)*L*(R)**2                                  BMLO0820
              CALL FEMREL(F5 ,F11,B4 ,B1 ,0,IELNO)                   BMLO0830
              F9 =(F5+F11-P*R*L)/L                                   BMLO0840
              F3 =-P-F9                                              BMLO0850
            ELSE IF (KDOF.EQ.4) THEN                                 BMLO0860
              F4 =-P*(1-R)                                           BMLO0870
              F10=-P*R                                               BMLO0880
              CALL FEMREL(F4 ,F10,B5 ,B2 ,1,IELNO)                   BMLO0890
            ELSE IF (KDOF.EQ.5) THEN                                 BMLO0900
              F5 =-P*(1 - 4*R + 3*R**2)                              BMLO0910
              F11= P*( 2*R - 3*R**2)                                 BMLO0920
              CALL FEMREL(F5 ,F11,B4 ,B1 ,0,IELNO)                   BMLO0930
              F9=( F5+F11+P)/L                                       BMLO0940
              F3=-F9                                                 BMLO0950
            ELSE IF (KDOF.EQ.6) THEN                                 BMLO0960
              F6 =-P*(1 - 4*R + 3*R**2)                              BMLO0970
              F12= P*( 2*R - 3*R**2)                                 BMLO0980
              CALL FEMREL(F6 ,F12,B3 ,B0 ,0,IELNO)                   BMLO0990
              F8 =-(F6+F12+P)/L                                      BMLO1000
              F2 = -F8                                               BMLO1010
            ENDIF                                                    BMLO1020
C                                                                    BMLO1030
C....... UNIFORM LOADING                                             BMLO1040
          ELSE IF (KGRP.EQ.200) THEN                                 BMLO1050
            W=ELD(1,N)                                               BMLO1060
            IF (KDOF.EQ.1) THEN                                      BMLO1070
              F1=-W*L/2                                              BMLO1080
              F7=-W*L/2                                              BMLO1090
            ELSE IF (KDOF.EQ.2) THEN                                 BMLO1100
              F6 =-W*L*L/12                                          BMLO1110
              F12= W*L*L/12                                          BMLO1120
              CALL FEMREL(F6 ,F12,B3 ,B0 ,0,IELNO)                   BMLO1130
              F8=(-F6-F12)/L  -W*L/2                                 BMLO1140
              F2=-W*L-F8                                             BMLO1150
            ELSE IF (KDOF.EQ.3) THEN                                 BMLO1160
              F5 = W*L*L/12                                          BMLO1170
              F11=-W*L*L/12                                          BMLO1180
              CALL FEMREL(F5 ,F11,B4 ,B1 ,0,IELNO)                   BMLO1190
              F9 =(F5+F11)/L  - W*L/2                                BMLO1200
              F3 =-W*L-F9                                            BMLO1210
            ELSE IF (KDOF.EQ.4) THEN                                 BMLO1220
              F4 =-W*L/2                                             BMLO1230
              F10=-W*L/2                                             BMLO1240
              CALL FEMREL(F4 ,F10,B5 ,B2 ,1,IELNO)                   BMLO1250
            ELSE                                                     BMLO1260
              WRITE (6,900) IELNO,' UNIFORM MT OR UNIFORM MZ'        BMLO1270
            ENDIF                                                    BMLO1280
C                                                                    BMLO1290
C....... USER INPUT FIXED END FORCE                                  BMLO1300
          ELSE IF (KGRP.EQ.400) THEN                                 BMLO1310
            DO 400 I=1,12                                            BMLO1320
  400       FORCE(I)= ELD(I,N)                                       BMLO1330
C                                                                    BMLO1340
C....... INVALID LOAD GROUP                                          BMLO1350
        ELSE                                                         BMLO1360
            WRITE (6,900) IELNO,'INVALID LOAD GROUP'                 BMLO1370
          ENDIF                                                      BMLO1380
C                                                                    BMLO1390
C....... TRANSFER BEAM LOADS TO FIXED END FORCES.....                BMLO1400
          DO 800 I=1,12                                              BMLO1410
  800     FEM(I,LOAD)=FEM(I,LOAD)-FORCE(I)                           BMLO1420
C                                                                    BMLO1430
C                                                                    BMLO1440
 1000 CONTINUE                                                       BMLO1450
C                                                                    BMLO1460
  900 FORMAT('  ER','ROR CALCULATING FIXED END FORCES FOR ELEMENT #', BMLO1470
     &  I5,' ',A,' IS NOT AVAILABLE')                                BMLO1480
      RETURN                                                         BMLO1490
      END                                                            BMLO1500
C*********************************************************************CKR00010
C          DEBUG UNIT(6),SUBCHK,SUBTRACE                             CKR00020
C          END DEBUG                                                 CKR00030
      SUBROUTINE CKRNG(V,A,B,NAME)                                   CKR00040
      CHARACTER*(*) NAME                                             CKR00050
      IF (A.LT.V .AND. V.LT.B) RETURN                                CKR00060
      WRITE (6,10) NAME,A,V,B                                        CKR00070
   10 FORMAT (' ====>',A,' IS OUT OF RANGE, SET ',1P,G13.5,' LE INPUT=',CKR00080
     &  G13.5,' LE ',G13.5)                                          CKR00090
      V=MIN(V,B)                                                     CKR00100
      V=MAX(V,A)                                                     CKR00110
      RETURN                                                         CKR00120
      END                                                            CKR00130
C*********************************************************************CKS00010
C          DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT                        CKS00020
C          END DEBUG                                                 CKS00030
      SUBROUTINE CKSTOR(ITEMP,IW)                                    CKS00040
      INCLUDE (ZCOMN)                                                CKS00050
C                                                                    CKS00060
C*******************************                                     CKS00070
C* CHECK STORAGE REQUIREMENTS ***                                    CKS00080
C*******************************                                     CKS00090
      N=MAX(IZ,IW)-ITEMP                                             CKS00100
      IF (N.LE.MAXZ) RETURN                                          CKS00110
C                                                                    CKS00120
      WRITE (6,10) N,MAXZ                                            CKS00130
   10 FORMAT (///1X, 49('*') /                                       CKS00140
     &  ' ***  AVAILABLE STORAGE IS EXCEEDED'           ,T48,'***'/  CKS00150
     &  ' ***  REQUIRED STORAGE =',I10                  ,T48,'***'/  CKS00160
     &  ' ***  AVAILABLE STORAGE =',I10                 ,T48,'***'/  CKS00170
     &  ' ***  EXECUTION IS TERMINATED.'                ,T48,'***'/  CKS00180
     &  ' ***  RECOMPILE PROGRAM WITH INCREASED MEMORY',T48,'***'/  CKS00190
     &  1X,49('*')  )                                                CKS00200
C                                                                    CKS00210
C------ INITIATE IBM ER_ROR TRACEBACK ROUTINE                        CKS00220
      CALL ERRTRA                                                    CKS00230
C                                                                    CKS00240
C------- ABORT RUN                                                   CKS00250
      STOP 'REQUIRED MEMORY > AVAILABLE MEMORY '                     CKS00260
      END                                                            CKS00270
@PROCESS SDUMP                                                       COM00010
C          DEBUG UNIT(6),SUBCHK,SUBTRACE                             COM00020
C          END DEBUG                                                 COM00030
      SUBROUTINE COMDMP(Z,NZ,DZ)                                     COM00040
      DIMENSION Z(1),NZ(1),DZ(1)                                     COM00050
      DOUBLE PRECISION DZ                                            COM00060
      LOGICAL BTEST                                                  COM00070
      INCLUDE (ZCOMN1)                                               COM00080
C                                                                    COM00090
      WRITE (6,50) TITLE(1),TITLE(2)                                 COM00100
      WRITE (6,51) 'DUMPING MEMORY'                                  COM00110
      CALL SDUMP                                                     COM00120
      IF (.NOT. BTEST(IBUG,13) ) RETURN                              COM00130
C                                                                    COM00140
            I=0                                                      COM00150
            K=0                                                      COM00160
            DO 100 J=1,IZ,2                                          COM00170
               I=I+1                                                 COM00180
               K=K+1                                                 COM00190
               IF (NZ(I).NE.0) WRITE(6,110) I,NZ(I),Z(I),DZ(K)       COM00200
               I=I+1                                                 COM00210
  100          IF (NZ(I).NE.0) WRITE(6,110) I,NZ(I),Z(I)            COM00220
C                                                                    COM00230
   50 FORMAT ('1 STRUCTURE....: ',A,/                                COM00240
     &        ' SOLUTION.....: ',A,/)                               COM00250
   51 FORMAT (1X,A)                                                  COM00260
  110 FORMAT (' ZMATRX(',I6,')=',I15,1P,G20.10,',',G20.10)           COM00270
      RETURN                                                         COM00280
      END                                                            COM00290
C*********************************************************************COM00010
CC        DEBUG UNIT(6),SUBCHK,SUBTRACE                              COM00020
CC        END DEBUG                                                  COM00030
      SUBROUTINE COMPT(IOPT,LZDATA)                                  COM00040
      CHARACTER*112 TITLE(2)                                         COM00050
      COMMON /ZDATAP/ TITLE                                          COM00060
      COMMON /ZDATA/  IZ  ,IDATA(1)                                  COM00070
C                                                                    COM00080
      IF (IOPT.EQ.1)  THEN                                           COM00090
         DO 10 I=1,LZDATA-1                                          COM00100
   10    IDATA(I)=0                                                  COM00110
         IZ=1                                                        COM00120
         TITLE(1)=' '                                                COM00130
         TITLE(2)=' '                                                COM00140
      ENDIF                                                          COM00150
C                                                                    COM00160
      END                                                            COM00170
C*********************************************************************CRO00010
C          DEBUG UNIT(6),SUBCHK,SUBTRACE                             CRO00020
C          END DEBUG                                                 CRO00030
      SUBROUTINE CROSS(VZ,VX,VY,NORM)                                CRO00040
      DIMENSION VX(3),VY(3),VZ(3)                                    CRO00050
      LOGICAL BTEST                                                  CRO00060
C                                                                    CRO00070
      CROSSI(A1,B1,C1,A2,B2,C2)=B1*C2-B2*C1                          CRO00080
      CROSSJ(A1,B1,C1,A2,B2,C2)=C1*A2-C2*A1                          CRO00090
      CROSSK(A1,B1,C1,A2,B2,C2)=A1*B2-A2*B1                          CRO00100
      VALUE (A1,A2,A3)=SQRT(A1**2+A2**2+A3**2)                       CRO00110
C                                                                    CRO00120
C ------- CALCULATE VZ, IF VY WAS CORRECTLY INPUT                    CRO00130
      VZ(1)=CROSSI(VX(1),VX(2),VX(3),VY(1),VY(2),VY(3))              CRO00140
      VZ(2)=CROSSJ(VX(1),VX(2),VX(3),VY(1),VY(2),VY(3))              CRO00150
      VZ(3)=CROSSK(VX(1),VX(2),VX(3),VY(1),VY(2),VY(3))              CRO00160
C                                                                    CRO00170
C ------- NORMALIZE VX,VY,VZ                                         CRO00180
      IF (NORM.NE.0) THEN                                            CRO00190
            AVX=VALUE(VX(1),VX(2),VX(3))                             CRO00200
            AVY=VALUE(VY(1),VY(2),VY(3))                             CRO00210
            AVZ=VALUE(VZ(1),VZ(2),VZ(3))                             CRO00220
            DO 10 I=1,3                                              CRO00230
               VX(I)=VX(I)/AVX                                       CRO00240
               VY(I)=VY(I)/AVY                                       CRO00250
   10          VZ(I)=VZ(I)/AVZ                                       CRO00260
C                                                                    CRO00270
      ENDIF                                                          CRO00280
      RETURN                                                         CRO00290
      END                                                            CRO00300
C - DIRECTION OF LOADING ..........................................  DIR00010
C          DEBUG UNIT(6),SUBCHK,SUBTRACE                             DIR00020
C          END DEBUG                                                 DIR00030
C     P       = CURRENT LOAD                                         DIR00040
C     PL      = LAST LOAD                                            DIR00050
C     D       = CURRENT DISPLACEMENT                                 DIR00060
C     DIR     = CURRENT DIRECTION                                    DIR00070
C               DIR=1, POSITIVE LOADING                              DIR00080
C               DIR=2, POSITIVE UNLOADING                            DIR00090
C               DIR=3, NEGATIVE LOADING                              DIR00100
C               DIR=4, NEGATIVE UNLOADING                            DIR00110
C     DIRL    = LAST DIRECTION                                       DIR00120
C     VEL     = PRODUCT OF LAST AND CURRENT VELOCITIES               DIR00130
      SUBROUTINE DIRECT(DIR,DIRL,P,PL,VEL)                           DIR00140
      IMPLICIT REAL(A-H,O-Z)                                         DIR00150
      IMPLICIT INTEGER(I-N)                                          DIR00160
      INTEGER DIR,DIRL                                               DIR00170
      LOGICAL BTEST                                                  DIR00180
C                                                                    DIR00190
      IF (P.GT.0) THEN                                               DIR00200
         IF (P.GT.PL .AND. PL.GT.0) THEN                             DIR00210
               DIRL=1                                                DIR00220
         ELSE IF (P.GT.PL .AND. PL.LT.0) THEN                        DIR00230
               DIRL=4                                                DIR00240
         ELSE IF (P.LT.PL) THEN                                      DIR00250
               DIRL=2                                                DIR00260
         ELSE                                                        DIR00270
               DIRL=DIR                                              DIR00280
         ENDIF                                                       DIR00290
      ELSE IF (P.LT.0) THEN                                          DIR00300
         IF (P.LT.PL .AND. PL.LT.0) THEN                             DIR00310
               DIRL=3                                                DIR00320
         ELSE IF (P.LT.PL .AND. PL.GT.0) THEN                        DIR00330
               DIRL=2                                                DIR00340
         ELSE IF (P.GT.PL) THEN                                      DIR00350
               DIRL=4                                                DIR00360
         ELSE                                                        DIR00370
               DIRL=DIR                                              DIR00380
         ENDIF                                                       DIR00390
      ELSE                                                           DIR00400
         IF (PL.LT.0) THEN                                           DIR00410
               DIRL=4                                                DIR00420
         ELSE IF (PL.GT.0) THEN                                      DIR00430
               DIRL=2                                                DIR00440
         ELSE                                                        DIR00450
               DIRL=DIR                                              DIR00460
         ENDIF                                                       DIR00470
      ENDIF                                                          DIR00480
C                                                                    DIR00490
      IF (DIRL.EQ.1 .OR.DIRL.EQ.0) THEN                              DIR00500
         IF (VEL.GT.0) THEN                                          DIR00510
               DIR=1                                                 DIR00520
         ELSE                                                        DIR00530
               DIR=2                                                 DIR00540
         ENDIF                                                       DIR00550
C                                                                    DIR00560
      ELSE IF (DIRL.EQ.2) THEN                                       DIR00570
         IF (VEL.GT.0) THEN                                          DIR00580
               IF (P.GT.0) THEN                                      DIR00590
                     DIR=2                                           DIR00600
               ELSE                                                  DIR00610
                     DIR=3                                           DIR00620
               ENDIF                                                 DIR00630
         ELSE                                                        DIR00640
               DIR=1                                                 DIR00650
         ENDIF                                                       DIR00660
C                                                                    DIR00670
                                                                     DIR00680
```

```
      ELSE IF (DIRL.EQ.3) THEN                                          DIR00690
         IF (VEL.GT.0) THEN                                             DIR00700
            DIR=3                                                       DIR00710
         ELSE                                                           DIR00720
            DIR=4                                                       DIR00730
         ENDIF                                                          DIR00740
                                                                        DIR00750
      ELSE IF (DIRL.EQ.4) THEN                                          DIR00760
         IF (VEL.GT.0) THEN                                             DIR00770
            IF (P.LT.0) THEN                                            DIR00780
               DIR=4                                                    DIR00790
            ELSE                                                        DIR00800
               DIR=1                                                    DIR00810
            ENDIF                                                       DIR00820
         ELSE                                                           DIR00830
            DIR=3                                                       DIR00840
         ENDIF                                                          DIR00850
      ELSE                                                              DIR00860
         WRITE (6,5)                                                    DIR00870
    5    FORMAT(5X,'ER','ROR IN SUBROUTINE DIRECT, DIRL.NE. 1,2,3,OR 4') DIR00880
         WRITE (6,10) DIR,DIRL,P,PL,VEL                                 DIR00890
      ENDIF                                                             DIR00900
      IF (DIR.EQ.0) DIR=1                                               DIR00910
                                                                        DIR00920
C     WRITE (6,10)      DIR,DIRL,P,PL,VEL                             DIR00930
   10  FORMAT (5X,'*** IN DIRECT ',1X,                                  DIR00940
     &  ' DIR=',I4,'  DIRL=',I4,'  P=',F8.3,' PL=',F8.3,'  VEL=',F10.3) DIR00950
      RETURN                                                            DIR00960
      END                                                               DIR00970
C*********************************************************************DMP00010
C        DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT                             DMP00020
C        END DEBUG                                                      DMP00030
C*********************************************************************DMP00040
      SUBROUTINE DMPDAT(IOPT,IWRITE,TO,DT)                              DMP00050
      LOGICAL TEST,OPEND,FALSE,AXIAL                                    DMP00060
      LOGICAL BTEST,FIRST                                               DMP00070
      CHARACTER*40 OPTION                                               DMP00080
      CHARACTER*80 NAME, DIR*2                                          DMP00090
      CHARACTER*112 TITL(2)                                             DMP00100
      CHARACTER*12 INAME                                                DMP00110
      DIMENSION RINPUT(100)                                             DMP00120
      DOUBLE PRECISION   TEI,TSE,DSE                                    DMP00130
      SAVE FIRST                                                        DMP00140
      INCLUDE (ZCOMN)                                                   DMP00150
C                                                                       DMP00160
      GO TO (100,200,300,400),IOPT                                      DMP00170
C                                                                       DMP00180
C*****************************************************                DMP00190
C= READ DATA ID TO BE SAVED, AND INITILIZE FILES  ===                   DMP00200
C*****************************************************                DMP00210
C                                                                       DMP00220
  100 FIRST = .TRUE.                                                    DMP00230
      IZDUMP=IZ                                                         DMP00240
      IZ=IZ+1                                                           DMP00250
      NWRITE=0                                                          DMP00260
      WRITE (6,109)                                                     DMP00270
  109 FORMAT (/,5X,' DATA WRITTEN TO FILES '/                           DMP00280
     &        5X,'========================'/)                          DMP00290
  101 READ (5,*) OPTION,IDOF,IUNIT,IGEN,IIDOF,IIUNIT                   DMP00300
  102 CONTINUE                                                          DMP00310
C                                                                       DMP00320
C-----SET UP DOF DATA                                                   DMP00330
      IF (TEST(OPTION,'DOF',.FALSE.)) THEN                              DMP00340
         WRITE (6,110) 'DEGREE OF FREEDOM',IDOF,IUNIT                   DMP00350
C                                                                       DMP00360
C        CHECK FILE STATUS                                              DMP00370
            INQUIRE (UNIT=IUNIT,OPENED=OPEND)                           DMP00380
            IF (OPEND) THEN                                             DMP00390
C              WRITE (6,105)                                            DMP00400
            ELSE                                                        DMP00410
               OPEN (UNIT=IUNIT)                                        DMP00420
            ENDIF                                                       DMP00430
C                                                                       DMP00440
C        INITILIZE DATA AND FILE                                        DMP00450
         NWRITE=NWRITE+1                                                DMP00460
         NZ(IZ)=1                                                       DMP00470
         NZ(IZ+1)=IDOF                                                  DMP00480
         NZ(IZ+2)=IUNIT                                                 DMP00490
         IZ=IZ+3                                                        DMP00500
         WRITE (IUNIT,120) TITLE(1)(1:80),TITLE(1)(81:)                 DMP00510
         WRITE (IUNIT,120) TITLE(2)(1:80),TITLE(2)(81:)                 DMP00520
         WRITE (IUNIT,130) 1,IDOF,IWRITE,TO,DT                          DMP00530
C------------ SET UP JOINT                                             DMP00540
      ELSE IF (TEST(OPTION,'JOINT',.FALSE.)) THEN                       DMP00550
         JOINT=IDOF                                                     DMP00560
         JOINTI=IQUICK(IDOF,NZ(IZID),NNODE)-1                           DMP00570
         IF (TEST(OPTION,'FX',.FALSE.)) THEN                            DMP00580
            IDOFJ=NZ(IZIDOF+JOINTI)                                     DMP00590
            DIR='FX'                                                    DMP00600
         ELSE IF (TEST(OPTION,'FY',.FALSE.)) THEN                       DMP00610
            IDOFJ=NZ(IZIDOF+JOINTI+MAXNOD*1)                            DMP00620
            DIR='FY'                                                    DMP00630
         ELSE IF (TEST(OPTION,'FZ',.FALSE.)) THEN                       DMP00640
            IDOFJ=NZ(IZIDOF+JOINTI+MAXNOD*2)                            DMP00650
            DIR='FZ'                                                    DMP00660
         ELSE IF (TEST(OPTION,'MX',.FALSE.)) THEN                       DMP00670
            IDOFJ=NZ(IZIDOF+JOINTI+MAXNOD*3)                            DMP00680
            DIR='MX'                                                    DMP00690
         ELSE IF (TEST(OPTION,'MY',.FALSE.)) THEN                       DMP00700
            IDOFJ=NZ(IZIDOF+JOINTI+MAXNOD*4)                            DMP00710
            DIR='MY'                                                    DMP00720
         ELSE IF (TEST(OPTION,'MZ',.FALSE.)) THEN                       DMP00730
            IDOFJ=NZ(IZIDOF+JOINTI+MAXNOD*5)                            DMP00740
            DIR='MZ'                                                    DMP00750
         ENDIF                                                          DMP00760
C                                                                       DMP00770
         WRITE (6,110) 'DEGREE OF FREEDOM',IDOFJ,IUNIT,JOINT,DIR        DMP00780
C                                                                       DMP00790
C        CHECK FILE STATUS                                              DMP00800
            INQUIRE (UNIT=IUNIT,OPENED=OPEND)                           DMP00810
            IF (OPEND) THEN                                             DMP00820
C              WRITE (6,105)                                            DMP00830
            ELSE                                                        DMP00840
               OPEN (UNIT=IUNIT)                                        DMP00850
            ENDIF                                                       DMP00860
C                                                                       DMP00870
C        INITILIZE DATA AND FILE                                        DMP00880
         NWRITE=NWRITE+1                                                DMP00890
         NZ(IZ)=1                                                       DMP00900
         NZ(IZ+1)=IDOFJ                                                 DMP00910
         NZ(IZ+2)=IUNIT                                                 DMP00920
         IZ=IZ+3                                                        DMP00930
         WRITE (IUNIT,120) TITLE(1)(1:80),TITLE(1)(81:)                 DMP00940
         WRITE (IUNIT,120) TITLE(2)(1:80),TITLE(2)(81:)                 DMP00950
         WRITE (IUNIT,130) 1,IDOFJ,IWRITE,TO,DT,JOINT,DIR               DMP00960
C                                                                       DMP00970
C-----SET UP ELEMENT                                                    DMP00980
      ELSE IF (TEST(OPTION,'ELE',.FALSE.)) THEN                         DMP00990
         WRITE (6,110) 'ELEMENT',IDOF,IUNIT                             DMP01000
C                                                                       DMP01010
C        CHECK FILE STATUS                                              DMP01020
            INQUIRE (UNIT=IUNIT,OPENED=OPEND)                           DMP01030
            IF (OPEND) THEN                                             DMP01040
C              WRITE (6,105)                                            DMP01050
            ELSE                                                        DMP01060
               OPEN (UNIT=IUNIT)                                        DMP01070
            ENDIF                                                       DMP01080
C                                                                       DMP01090
C        INITILIZE DATA AND FILE                                        DMP01100
         NWRITE=NWRITE+1                                                DMP01110
         NZ(IZ)=2                                                       DMP01120
         NZ(IZ+1)=IDOF                                                  DMP01130
         NZ(IZ+2)=IUNIT                                                 DMP01140
         IZ=IZ+3                                                        DMP01150
C                                                                       DMP01160
C        GET ELEMENT TYPE                                               DMP01170
         IC     = NZ(IZELE+IDOF )                                       DMP01180
         IELTYP = NZ(IC)                                                DMP01190
C                                                                       DMP01200
         WRITE (IUNIT,120) TITLE(1)(1:80),TITLE(1)(81:)                 DMP01210
         WRITE (IUNIT,120) TITLE(2)(1:80),TITLE(2)(81:)                 DMP01220
         WRITE (IUNIT,130) 2,IDOF,IWRITE,TO,DT,IELTYP,'ELEMENT'         DMP01230
C                                                                       DMP01240
C-----SET UP ENERGY                                                     DMP01250
      ELSE IF (TEST(OPTION,'ENERGY',.FALSE.)) THEN                      DMP01260
         WRITE (6,111) 'ENERGY BALANCE',IUNIT                           DMP01270
C                                                                       DMP01280
C        CHECK FILE STATUS                                              DMP01290
            INQUIRE (UNIT=IUNIT,OPENED=OPEND)                           DMP01300
            IF (OPEND) THEN                                             DMP01310
C              WRITE (6,105)                                            DMP01320
            ELSE                                                        DMP01330
               OPEN (UNIT=IUNIT)                                        DMP01340
            ENDIF                                                       DMP01350
C                                                                       DMP01360
C        INITILIZE DATA AND FILE                                        DMP01370
         NWRITE=NWRITE+1                                                DMP01380
         NZ(IZ)=3                                                       DMP01390
         NZ(IZ+1)=0                                                     DMP01400
         NZ(IZ+2)=IUNIT                                                 DMP01410
         IZ=IZ+3                                                        DMP01420
         WRITE (IUNIT,120) TITLE(1)(1:80),TITLE(1)(81:)                 DMP01430
         WRITE (IUNIT,120) TITLE(2)(1:80),TITLE(2)(81:)                 DMP01440
         WRITE (IUNIT,130) 3,IDOF,IWRITE,TO,DT,0,'ENERGY'               DMP01450
C                                                                       DMP01460
C-----SET UP SUM OF REACTIONS                                           DMP01470
      ELSE IF (TEST(OPTION,'SUMRCT',.FALSE.)) THEN                      DMP01480
         WRITE (6,111) 'SUMMATION OF REACTIONS',IUNIT                   DMP01490
C                                                                       DMP01500
C        CHECK FILE STATUS                                              DMP01510
            INQUIRE (UNIT=IUNIT,OPENED=OPEND)                           DMP01520
            IF (OPEND) THEN                                             DMP01530
C              WRITE (6,105)                                            DMP01540
            ELSE                                                        DMP01550
               OPEN (UNIT=IUNIT)                                        DMP01560
            ENDIF                                                       DMP01570
C                                                                       DMP01580
C        INITILIZE DATA AND FILE                                        DMP01590
         NWRITE=NWRITE+1                                                DMP01600
         NZ(IZ)=4                                                       DMP01610
         NZ(IZ+1)=0                                                     DMP01620
         NZ(IZ+2)=IUNIT                                                 DMP01630
         IZ=IZ+3                                                        DMP01640
         WRITE (IUNIT,120) TITLE(1)(1:80),TITLE(1)(81:)                 DMP01650
         WRITE (IUNIT,120) TITLE(2)(1:80),TITLE(2)(81:)                 DMP01660
         WRITE (IUNIT,130) 4,IDOF,IWRITE,TO,DT,0,'SUMRCT'               DMP01670
C                                                                       DMP01680
C-----SET UP SUM OF REACTIONS                                           DMP01690
      ELSE IF (TEST(OPTION,'GROUND',.FALSE.)) THEN                      DMP01700
         WRITE (6,111) 'GROUND MOTION',IUNIT                            DMP01710
C                                                                       DMP01720
C        CHECK FILE STATUS                                              DMP01730
            INQUIRE (UNIT=IUNIT,OPENED=OPEND)                           DMP01740
            IF (OPEND) THEN                                             DMP01750
C              WRITE (6,105)                                            DMP01760
            ELSE                                                        DMP01770
               OPEN (UNIT=IUNIT)                                        DMP01780
            ENDIF                                                       DMP01790
C                                                                       DMP01800
C        INITILIZE DATA AND FILE                                        DMP01810
         NWRITE=NWRITE+1                                                DMP01820
         NZ(IZ)=5                                                       DMP01830
         NZ(IZ+1)=0                                                     DMP01840
         NZ(IZ+2)=IUNIT                                                 DMP01850
         IZ=IZ+3                                                        DMP01860
         WRITE (IUNIT,120) TITLE(1)(1:80),TITLE(1)(81:)                 DMP01870
         WRITE (IUNIT,120) TITLE(2)(1:80),TITLE(2)(81:)                 DMP01880
         WRITE (IUNIT,130) 5,IDOF,IWRITE,TO,DT,0,'GROUND'               DMP01890
C                                                                       DMP01900
      ELSE IF (TEST(OPTION,'END',.FALSE.)) THEN                         DMP01910
         NZ(IZDUMP)=NWRITE                                              DMP01920
         RETURN                                                         DMP01930
      ELSE                                                              DMP01940
         WRITE (6,140) OPTION                                           DMP01950
         GO TO 101                                                      DMP01960
      ENDIF                                                             DMP01970
C                                                                       DMP01980
C........ GENERATE LOADS                                                DMP01990
      IF (IGEN.NE.0) THEN                                               DMP02000
         IGEN=IGEN-1                                                    DMP02010
         IDOF =IDOF  + IIDOF                                            DMP02020
         IUNIT=IUNIT + IIUNIT                                           DMP02030
         GO TO 102                                                      DMP02040
      ENDIF                                                             DMP02050
      GO TO 101                                                         DMP02060
C                                                                       DMP02070
  105 FORMAT (5X,'FILE IS ALREADY OPENED, WRITE IS ABORTED',10('*'))    DMP02080
  110 FORMAT (5X,A,' #',I5,' IS WRITTEN TO UNIT #',I5,:,'  JOINT:',I6,  DMP02090
     &        '   DIRECTION: ',A)                                       DMP02100
  111 FORMAT (5X,A,' IS WRITTEN TO UNIT #',I5 )                         DMP02110
  120 FORMAT (A)                                                        DMP02120
  130 FORMAT (3I10,1P,2G15.5,0P,1X,I6,A)                                DMP02130
  140 FORMAT (5X,'INVALID INPUT: ',A,', INPUT IS SKIPED...')            DMP02140
C                                                                       DMP02150
C*************************                                            DMP02160
C= WRITE DATA TO FILES ===                                              DMP02170
C*************************                                            DMP02180
C                                                                       DMP02190
  200 NUMB=NZ(IZDUMP)                                                   DMP02200
      DO 250 I=1,NUMB                                                   DMP02210
         II=IZDUMP+3*(I-1) +1                                           DMP02220
         ITYPE=NZ(II)                                                   DMP02230
         IDOF =NZ(II+1)                                                 DMP02240
         IUNIT=NZ(II+2)                                                 DMP02250
         IF (ITYPE.EQ.1) THEN                                           DMP02260
            J=IDOF-1                                                    DMP02270
            IF (IZVEL.EQ.0 .OR. IZACC.EQ.0) THEN                        DMP02280
               WRITE (IUNIT,210) Z(IZLOAD+J),Z(IZDISP+J),0,0            DMP02290
            ELSE                                                        DMP02300
               WRITE (IUNIT,210) Z(IZLOAD+J),Z(IZDISP+J),               DMP02310
     &                           Z(IZVEL+J),Z(IZACC+J)                  DMP02320
            ENDIF                                                       DMP02330
         ELSE IF (ITYPE.EQ.2) THEN                                      DMP02340
            I7=7                                                        DMP02350
            IELNO=IDOF                                                  DMP02360
            LASTTP=0                                                    DMP02370
            FALSE=.FALSE.                                               DMP02380
            CALL ELELIB                                                 DMP02390
     &        (I7    ,LSTTYP ,FIRST,IREL,FIRST, NAME ,ESSE,EPSE,DAMAGE,  DMP02400
     &        DUCFAG,                                                   DMP02410
     &        IELNO,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR) DMP02420
         ELSE IF (ITYPE.EQ.3) THEN                                      DMP02430
            DIFF=EIE-ESE-EKE-PSE-EDD                                    DMP02440
            IF (EIE.NE.0) THEN                                          DMP02450
              RE=DIFF/EIE * 100                                         DMP02460
            ELSE                                                        DMP02470
               IF (DIFF.NE.0) RE=100.00                                 DMP02480
               IF (DIFF.EQ.0) RE=  0.00                                 DMP02490
            ENDIF                                                       DMP02500
            WRITE (IUNIT,210) EIE, ESE,EKE,  PSE,EDD,  RE              DMP02510
         ELSE IF (ITYPE.EQ.4) THEN                                      DMP02520
            WRITE (IUNIT,210) SUMRCT                                    DMP02530
         ELSE IF (ITYPE.EQ.5) THEN                                      DMP02540
```

```
            WRITE (IUNIT,210) GROUND
          ENDIF
  250 CONTINUE
  210 FORMAT(1P,6G13.5)
      FIRST=.FALSE.
      RETURN
C
C**********************
C= CLOSE DATA FILES ===
C**********************
C
  300 NUMB=NZ(IZDUMP)
      DO 350 I=1,NUMB
        II=IZDUMP+3*(I-1) +1
C       ITYPE=NZ(II)
C       IDOF =NZ(II+1)
        IUNIT=NZ(II+2)
        CLOSE(UNIT=IUNIT)
  350 CONTINUE
      RETURN
C
C*****************************************
C= READ DATA FROM FILE AND WRITE TO FT06 ===
C*****************************************
C
  400 OPEN (UNIT=IUNIT)
      REWIND(UNIT=IUNIT)
      READ (IUNIT,120,END=499) TITL(1)(1:80),TITL(1)(81:)
      READ (IUNIT,120,END=499) TITL(2)(1:80),TITL(2)(81:)
      WRITE (6,405) TITL(1),TITL(2)
  405 FORMAT ('1 STRUCTURE.....: ',A,/
     &        ' SOLUTION.....: ',A,/)
      READ (IUNIT,131,END=499) ITYPE,IDOF,IWRITE,TO,DT,ITEMP,INAME
  131 FORMAT (3I10,0P,2G15.5,0P,1X,I6,A)
C
C-------- DOF DATA --------------
      IF (ITYPE.EQ.1) THEN
        WRITE (6,450) IDOF,IUNIT,ITEMP,INAME
        ISTEP=-IWRITE
  410   READ (IUNIT,*,END=440) RLOAD,DISP,VEL,ACC
        ISTEP=ISTEP+IWRITE
        T=TO+DT*ISTEP
        IF (MOD(ISTEP,INC).EQ.0)
     &  WRITE (6,460) ISTEP,T,RLOAD,DISP,VEL,ACC
        GO TO 410
      ELSE IF (ITYPE.EQ.2) THEN
        IELNO =IDOF
        IELTYP=ITEMP
        WRITE (6,445) IELNO,IUNIT

C       GET ELEMENT TYPE AND COMPARE WITH INPUT TYPE.....
        IF (IZELE.GT.0) THEN
            IC= NZ(IZELE+IELNO)
        ELSE
            IC=-1
        ENDIF
        IF (IC.GT.0 .AND. IC.LE.MAXZ) THEN
            IELTPO = NZ(IC)
        ELSE
            IELTPO = -100
        ENDIF
C       CREATE DUMMY ELEMENT DIRECTORY IF INPUT TYPE IS DIFF
        IZELET=IZELE
        IF (IELTYP.NE.IELTPO) THEN
           CALL CKSTOR(IELNO+2,0)
           IZELE = IZ+2
           NZ(IZELE+IELNO  )=IZELE+IELNO+1
           NZ(IZELE+IELNO+1)=IELTYP
        ENDIF

        I8=8
        LASTYP=0
        FALSE=.FALSE.
        CALL ELELIB
     &   (I8    ,LSTTYP ,FALSE,IREL,FALSE,  NAME ,EESE,EPSE,DAMAGE,
     &    DUCFAG,
     &    IELNO,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)

C       RESTORE OLD  ELEMENT DIRECTORY
        IZELE=IZELET
      ELSE IF (ITYPE.EQ.3) THEN
        TEI=0.00
        TSE=0.00
        DSE=0.00
        EIEM=0
        ESEM=0
        EKEM=0
        PSEM=0
        EDDM=0
        WRITE (6,455) IUNIT
        ISTEP=-IWRITE
  430   READ (IUNIT,*,END=431) EIE, ESE,EKE,  PSE,EDD,  RE
        TEI=TEI+EIE
        TSE=TSE+ESE+EKE+PSE+EDD
        DSE=DSE+ABS(EIE-ESE-EKE-PSE-EDD)
        EIEM=AMAX1(SNGL(EIE),EIEM)
        ESEM=AMAX1(SNGL(ESE),ESEM)
        EKEM=AMAX1(SNGL(EKE),EKEM)
        PSEM=AMAX1(SNGL(PSE),PSEM)
        EDDM=AMAX1(SNGL(EDD),EDDM)
        ISTEP=ISTEP+IWRITE
        T=TO+DT*ISTEP
        IF (MOD(ISTEP,INC).EQ.0)
     &  WRITE (6,459) ISTEP,T, EIE, ESE,EKE,  PSE,EDD,  RE
        GO TO 430
CC431   RE =100*(TEI-TSE)/TEI
  431   RE =200*DSE/(TEI+TSE)
        WRITE (6,458) EIEM,ESEM,EKEM,PSEM,EDDM,RE
        GO TO 440
      ELSE IF (ITYPE.EQ.4) THEN
        WRITE (6,456) IUNIT
        ISTEP=-IWRITE
  435   READ (IUNIT,*,END=440) SUMRCT
        ISTEP=ISTEP+IWRITE
        T=TO+DT*ISTEP
        IF (MOD(ISTEP,INC).EQ.0)
     &  WRITE (6,460) ISTEP,T, SUMRCT
        GO TO 435
      ELSE IF (ITYPE.EQ.5) THEN
        WRITE (6,457) IUNIT
        ISTEP=-IWRITE
  436   READ (IUNIT,*,END=440) GROUND
        ISTEP=ISTEP+IWRITE
        T=TO+DT*ISTEP
        IF (MOD(ISTEP,INC).EQ.0)
     &  WRITE (6,461) ISTEP,T, GROUND
        GO TO 436
      ENDIF
C
  440 CLOSE (UNIT=IUNIT)
      RETURN
C
  499 CLOSE (UNIT=IUNIT)
      WRITE (6,500) IUNIT
  500 FORMAT (///10X,'PREMATURE END OF FILE, UNIT=',I5/
     &        10X,'READ COMMAND IS ABORTED'/// )
      RETURN
C
  445 FORMAT (5X,'ELEMENT #',I5,' IS READ FROM UNIT #',I5)
  450 FORMAT (5X,'DEGREE OF FREEDOM #',I5,' IS READ FROM UNIT #',I5,
     &        5X,'JOINT #',I5,', DIRECTION= ',A2//
     & 6X,'STEP',       '      TIME       ','     LOAD       ',
     &' DISPLACEMENT ','    VELOCITY    ','  ACCELERATION  ')
  455 FORMAT (5X,'ENERGY DATA IS READ FROM UNIT #',I5//
     & 6X,'STEP',       '      TIME       ','     INPUT      ',
     &'ELASTIC STRAIN ','   KINETIC     ','PLASTIC STRAIN ',
     &' DAMPINED    ',' RELATIVE ERROR %')
  456 FORMAT (5X,'SUMMATION OF REACTIONS ARE READ FROM UNIT #',I5//
     & 6X,'STEP',       '      TIME       ','      FX        ',
     &'      FY        ','      FZ        ','      MX        ',
     &'      MY        ','      MZ        ' )
  457 FORMAT (5X,'GROUND MOTION IS READ FROM UNIT #',I5//
     & 2X,'   ',       '                 ',
     &' /---------- ACCELERATION ----------/',
     &' /------------ VELOCITY ------------/',
     &' /---------- DISPLACEMENT ----------/', /
C       1234567890AB1234567890AB1234567890AB
     & 2X,'STEP',       '      TIME       ',
     &' GLOBAL X  ',' GLOBAL Y  ',' GLOBAL Z  ',
     &' GLOBAL X  ',' GLOBAL Y  ',' GLOBAL Z  ',
     &' GLOBAL X  ',' GLOBAL Y  ',' GLOBAL Z  ')
C       1234567890AB  1234567890AB  1234567890AB
  458 FORMAT(25X,90('-')/10X,'  MAXIMUN        ',1P,5G15.5,/
     &       /25X,'GROSS RE:',0P,F15.4,'%')
C     &       /25X,'GROSS RE2:',0P,F15.4,'%')
  459 FORMAT(I10,1P,6G15.5,0P,F15.4)
  460 FORMAT(I10,1P,7G15.5)
  461 FORMAT(I6,1P,10G12.4)
C
      END
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP
C**********************************************************************
C       DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE
C       END DEBUG
C**********************************************************************
C  DUCTILITIES AND EXCURSION RATIOS
C
      SUBROUTINE DNE(IOPT,U,E,DT,D,ESE,PSE,PSEOLD,CSE)
      DIMENSION U(3),E(6),UC(3)
C       WRITE (6,*)  '     U=',U
C       WRITE (6,*)  '     E=',E
C       WRITE (6,*)  '    DT=',DT
C       WRITE (6,*)  '     D=',D
C       WRITE (6,*)  '   ESE=',ESE
C       WRITE (6,*)  '   PSE=',PSE
C       WRITE (6,*)  'PSEOLD=',PSEOLD
C       WRITE (6,*)  '   CSE=',CSE
C
C
C-------- VARIABLES ------------------------------------------------------
C     U(I)    = DUCTILITY   (I)
C     E(I)    = EXCURSION RATIO (I)
C     E(I+3)  = DUCTILITY (I) FOR THE CURRENT HALF CYCLE
C     DT      = YEILD DISPLACMENT
C     D       = CURRENT DISPL
C     TSE     = TOTAL STRAIN ENERGY
C     PSE     = PLASTIC STRAIN ENERGY
C     ESE     = ELASTIC STRAIN ENERGY
C     CSE     = CONSTANT STRAIN ENERGY AT YEILD.....
C     UC      = CURRENT DUCTILITY
C----------------------------------------------------------------------
C
      IF (IOPT.EQ.1) THEN
C           ..... CALCULATE CURRENT DUCTILITIES .....
            PSEO=PSE-PSEOLD
            IF (DT.NE.0)  UC(1)=D/DT
            IF (ESE.NE.0) UC(2)=(PSEO/ESE +1 )
            IF (CSE.NE.0) UC(3)=(PSEO/CSE +1 )
C           DO 10 I=1,3
              J=I+3
C             IF ( UC(I).GE.U(I) ) U(I)=UC(I)
C 10          E(J)=MAX( E(J),UC(I) )
            U(1)=MAX( U(1),UC(1) )
            U(2)=MAX( U(2),UC(2) )
            U(3)=MAX( U(3),UC(3) )
            E(4)=MAX( E(4),UC(1) )
            E(5)=MAX( E(5),UC(2) )
            E(6)=MAX( E(6),UC(3) )
      ELSE
C           ..... CALCULATE EXCURSION RATIO'S .....
            DO 20 I=1,3
              J=I+3
              IF (E(J).GT.1) E(I) = E(I) + E(J) -1
   20         E(J)=0
      ENDIF
C       WRITE (6,*)  '     U=',U
C       WRITE (6,*)  '     E=',E
C       WRITE (6,*)  '    DT=',DT
C       WRITE (6,*)  '     D=',D
C       WRITE (6,*)  '   ESE=',ESE
C       WRITE (6,*)  '   PSE=',PSE
C       WRITE (6,*)  'PSEO=',PSEO
C
      RETURN
      END
C@PROCESS SDUMP OPT(0) GOSTMT XREF
C**********************************************************************
C       DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT
C       END DEBUG
C**********************************************************************
      SUBROUTINE DYLOA(IOPT,BUG,N,NDOF,NAIN,T,TL,ACCEL2,TO,DT,GRAV,
     &    MAXNOD,NNODE,COSINE,IDOF,JTFLG,JTCOS,NCOS,
     &    TX,TY,TZ,MASS,MD,IMD,A,MC,P,GA,GAT)
      DIMENSION COSINE(3,3,NCOS),IDOF(MAXNOD,6)
      DIMENSION JTFLG(MAXNOD)   ,JTCOS(MAXNOD)
      DIMENSION TX(N),TY(N),TZ(N),A(NAIN),P(NDOF)
      DIMENSION V1(3),V2(3),V3(3),MD(NDOF+1),GA(3),GAT(3)
      REAL      MASS(IMD),MC(NDOF,3)
      CHARACTER*80 FORMAT(2),ATITLE(2)
      LOGICAL FMT,ECHO,BUG,PRINT,REWIND
      LOGICAL BTEST
      ACCEL2=0
C
      GO TO (100,200) IOPT
C
C*************************
C= INPUT ACCELERATIONS ===
C*************************
  100 CONTINUE
C------- ZERO GLOBAL BASE ACCELERATIONS
      DO 101 I=1,N
        TX(I)=0
        TY(I)=0
  101   TZ(I)=0
C
C-------- INPUT NUMBER OF ACCEL, AMP. FACTOR SCALE, AND TIME INCREMENT
      READ (5,*) NA,ASCALE,TO,DT,PRINT
C
C------- INPUT DIRECTION VECTORS V1 AND V2
      READ (5,*) V1,V2
      IF (BUG)  WRITE (6,29) 'INPUT- V1=',V1,' V2=',V2
C
C------- CALCULATE V3 = V1 X V2
      CALL CROSS(V3,V1,V2,0)
      IF (BUG) WRITE (6,29) 'V3=V1XV2 V1=',V1,' V2=',V2,' V3=',V3
C
C--------CALCULATE V2 = V3 X V1, AND NORMALIZE ALL VECTORS...
      CALL CROSS(V2,V3,V1,1)
      IF (BUG) WRITE (6,29) 'V2=V3XV1 V1=',V1,' V2=',V2,' V3=',V3
C
      IF (BUG)  WRITE (6,29) ' V1=',V1,' V2=',V2,' V3=',V3
   29 FORMAT(5X,3(1,A,SS,F9.5,' I',SP,F9.5,' J',F9.5,' K  '),SS)
C
C-------- INPUT INDIVIDUAL ACCELEROGRAMS
```

```
      DO 30 I=1,NA                                                      DYL00500
        READ (5,*) IN,NPTS,IDIR,FMT,ECHO ,REWIND                        DYL00510
        NPTS=MIN(NPTS,N)                                                DYL00520
        IF (REWIND .AND. IN.NE.5) REWIND (IN)                           DYL00530
C                                                                       DYL00540
        IF (FMT) THEN                                                   DYL00550
          READ (5,91)  FORMAT(1),FORMAT(2)                              DYL00560
          READ (IN,FORMAT(1)) ATITLE(1),ATITLE(2)                       DYL00570
        ELSE                                                            DYL00580
          READ (IN,91) ATITLE(1),ATITLE(2)                              DYL00590
        ENDIF                                                           DYL00600
C                                                                       DYL00610
        IF (IDIR.GE.1 .AND. IDIR.LE.3) THEN                             DYL00620
          WRITE (6,10) I,IN,ATITLE(1),ATITLE(2),NPTS,TO,DT              DYL00630
   10     FORMAT (/5X,' GROUND ACCELERATION RECORD '/                   DYL00640
     &            5X,'============================='//                 DYL00650
     &            5X,'INPUTING TRANSLATIONAL ACCELERATION RECORD #',    DYL00660
     &            I2,', FROM UNIT:',I4,/,5X,A/,5X,A/                    DYL00670
     &            5X,'THE RECORD CONTAINS ',I6,' POINTS, ',             DYL00680
     &               ' BEGINING AT TIME:',1P,G16.6,                     DYL00690
     &               ' WITH A TIME INCREMENT OF:',1P,G15.6)             DYL00700
        ELSE                                                            DYL00710
          WRITE (6,*) 'INVALID DIRECTION:,',IDIR,' SET 0 < IDIR < 4'    DYL00720
          GO TO 30                                                      DYL00730
        ENDIF                                                           DYL00740
C                                                                       DYL00750
        IF (IDIR.EQ.1) THEN                                             DYL00760
          CX=V1(1)                                                      DYL00770
          CY=V1(2)                                                      DYL00780
          CZ=V1(3)                                                      DYL00790
          WRITE (6,92) V1                                               DYL00800
        ELSE IF (IDIR.EQ.2) THEN                                        DYL00810
          CX=V2(1)                                                      DYL00820
          CY=V2(2)                                                      DYL00830
          CZ=V2(3)                                                      DYL00840
          WRITE (6,92) V2                                               DYL00850
        ELSE IF (IDIR.EQ.3) THEN                                        DYL00860
          CX=V3(1)                                                      DYL00870
          CY=V3(2)                                                      DYL00880
          CZ=V3(3)                                                      DYL00890
          WRITE (6,92) V3                                               DYL00900
        ENDIF                                                           DYL00910
C                                                                       DYL00920
        IF (FMT) THEN                                                   DYL00930
          READ (IN,FORMAT(2),END=25) (A(J),J=1,NPTS)                    DYL00940
        ELSE                                                            DYL00950
          READ (IN,*,END=25) (A(J),J=1,NPTS)                            DYL00960
        ENDIF                                                           DYL00970
C                                                                       DYL00980
        GO TO 27                                                        DYL00990
C---------- END OF FILE WAS ENCOUNTERED BEFORE ALL INPUT WAS READ       DYL01000
   25   NPTS=J                                                          DYL01010
        WRITE (6,26) J                                                  DYL01020
   26   FORMAT(5X,'ONLY',I6,                                            DYL01030
     &       ' POINTS WERE READ BEFORE END OF FILE WAS ENCOUNTERED')     DYL01040
C                                                                       DYL01050
   27   IF (ECHO) WRITE (6,94) (A(J),J=1,NPTS)                          DYL01060
        WRITE (6,9280)                                                  DYL01070
        CALL SMAX (NPTS,DT,A,YMX,YMN,T1,T2,NPTS,AVE,STDEV,RMS)          DYL01080
C       T1=T1-DT+TO                                                     DYL01090
C       T2=T2-DT+TO                                                     DYL01100
        T1=T1+TO                                                        DYL01110
        T2=T2+TO                                                        DYL01120
        WRITE (6,9290) 'INPUT ACCELERATION  ',                          DYL01130
     &                 YMX,T1,YMN,T2,AVE,STDEV,RMS                      DYL01140
C                                                                       DYL01150
        DO 20 J=1,NPTS                                                  DYL01160
          ACC=A(J)*ASCALE                                               DYL01170
          TX(J)=TX(J) + ACC*CX                                          DYL01180
          TY(J)=TY(J) + ACC*CY                                          DYL01190
          TZ(J)=TZ(J) + ACC*CZ                                          DYL01200
   20   CONTINUE                                                        DYL01210
C                                                                       DYL01220
C                                                                       DYL01230
   30 CONTINUE                                                          DYL01240
      IF (PRINT) THEN                                                   DYL01250
        WRITE (6,31) 'X'                                                DYL01260
        WRITE (6,93) (TX(J),J=1,N)                                      DYL01270
        WRITE (6,31) 'Y'                                                DYL01280
        WRITE (6,93) (TY(J),J=1,N)                                      DYL01290
        WRITE (6,31) 'Z'                                                DYL01300
        WRITE (6,93) (TZ(J),J=1,N)                                      DYL01310
   31   FORMAT(//5X,'GLOBAL BASE ACCELERATION IN THE ',A,               DYL01320
     &     ' DIRECTION (g''s)  ')                                       DYL01330
      ENDIF                                                             DYL01340
C                                                                       DYL01350
      WRITE (6,9280)                                                    DYL01360
      CALL SMAX (N,DT,TX,YMX,YMN,T1,T2,N,AVE,STDEV,RMS)                 DYL01370
C     T1=T1-DT+TO                                                       DYL01380
C     T2=T2-DT+TO                                                       DYL01390
      T1=T1+TO                                                          DYL01400
      T2=T2+TO                                                          DYL01410
      WRITE (6,9290) 'X-AXIS ACCELERATION ',YMX,T1,YMN,T2,AVE,STDEV,RMS DYL01420
      CALL SMAX (N,DT,TY,YMX,YMN,T1,T2,N,AVE,STDEV,RMS)                 DYL01430
C     T1=T1-DT+TO                                                       DYL01440
C     T2=T2-DT+TO                                                       DYL01450
      T1=T1+TO                                                          DYL01460
      T2=T2+TO                                                          DYL01470
      WRITE (6,9290) 'Y-AXIS ACCELERATION ',YMX,T1,YMN,T2,AVE,STDEV,RMS DYL01480
      CALL SMAX (N,DT,TZ,YMX,YMN,T1,T2,N,AVE,STDEV,RMS)                 DYL01490
C     T1=T1-DT+TO                                                       DYL01500
C     T2=T2-DT+TO                                                       DYL01510
      T1=T1+TO                                                          DYL01520
      T2=T2+TO                                                          DYL01530
      WRITE (6,9290) 'Z-AXIS ACCELERATION ',YMX,T1,YMN,T2,AVE,STDEV,RMS DYL01540
C                                                                       DYL01550
C---------- CALCULATE MASS * COSINE  MATRIX FOR EACH DOF...             DYL01560
C..... LOOP FOR DIRECTION - JDIR                                       DYL01570
      DO 90 JDIR=1,3                                                    DYL01580
C                                                                       DYL01590
C..... ZERO MC MATRIX AND TEMP COSINE  MATRIX FOR JDIR                  DYL01600
        DO 40 I=1,NDOF                                                  DYL01610
          MC(I,JDIR)=0                                                  DYL01620
   40     A(I)=0                                                        DYL01630
C                                                                       DYL01640
C..... CONSTRUCT COSINE  MATRIX FOR JDIR                                DYL01650
C       OMIT ALL DOF THAT ARE CONSTRAINED TO A MASTER DOF...            DYL01660
        DO 50 NODE=1,NNODE                                              DYL01670
          ICOS=JTCOS(NODE)                                              DYL01680
          DO 50 K=1,3                                                   DYL01690
            IF (.NOT. BTEST(JTFLG(NODE),K+11)) THEN                     DYL01700
              II=IDOF(NODE,K)                                           DYL01710
              A(II)=A(II)+COSINE(JDIR,K,ICOS)                           DYL01720
            ENDIF                                                       DYL01730
   50     CONTINUE                                                      DYL01740
C                                                                       DYL01750
C..... MULTIPLY MASS * COSINE  MATRIX FOR JDIR                          DYL01760
C             STORE IN MC(___,JDIR)                                     DYL01770
        DO 60 J=1,NDOF                                                  DYL01780
          MDJ=MD(J)+J                                                   DYL01790
          M  =J-(MD(J+1)-MD(J))+1                                       DYL01800
          DO 60 I=M,J                                                   DYL01810
   60       MC(I,JDIR)=MC(I,JDIR) + MASS(MDJ-I) * A(J)                  DYL01820
        DO 70 I=2,NDOF                                                  DYL01830
          MDI=MD(I)+I                                                   DYL01840
          M  =I-(MD(I+1)-MD(I))+1                                       DYL01850
          DO 70 J=M,I-1                                                 DYL01860
   70       MC(I,JDIR)=MC(I,JDIR) + MASS(MDI-J) * A(J)                  DYL01870
C                                                                       DYL01880
   90 CONTINUE                                                          DYL01890
C                                                                       DYL01900
      IF (BUG) CALL WMATRX(MC,NDOF,3,'MASS*COSINE')                     DYL01910
C                                                                       DYL01920
   91 FORMAT (A)                                                        DYL01930
   92 FORMAT(5X,'DIRECTION OF ACCELERATION:',                           DYL01940
     &     4X,SS,F9.5,' I',SP,F9.5,' J',F9.5,' K  ',SS)                 DYL01950
   93 FORMAT (1X,1P,10G12.4)                                            DYL01960
   94 FORMAT (/5X,'INPUT ACCELERATION:',                                DYL01970
     &                 (g''s) '                                         DYL01980
     &    /(1X,1P,10G12.4))                                             DYL01990
C                                                                       DYL02000
C=================================                                      DYL02010
C=   INITIAL GROUND ACCELERATION  ===                                   DYL02020
C=================================                                      DYL02030
      IF (T.GE.TO .AND. T.LE.N*DT ) THEN                                DYL02040
            I=(T-TO)/DT +1                                              DYL02050
            J=I+1                                                       DYL02060
            R=(I*DT+TO-T)/DT                                            DYL02070
            GAT(1)=TX(I)*R                                              DYL02080
            GAT(2)=TY(I)*R                                              DYL02090
            GAT(3)=TZ(I)*R                                              DYL02100
            IF (J.LE.N .AND. R.NE.1 ) THEN                              DYL02110
                 GAT(1)=GAT(1) + (1-R)*TX(J)                            DYL02120
                 GAT(2)=GAT(2) + (1-R)*TY(J)                            DYL02130
                 GAT(3)=GAT(3) + (1-R)*TZ(J)                            DYL02140
            ENDIF                                                       DYL02150
      ELSE                                                              DYL02160
            GAT(1)=0                                                    DYL02170
            GAT(2)=0                                                    DYL02180
            GAT(3)=0                                                    DYL02190
      ENDIF                                                             DYL02200
      GAT(1)=  GAT(1)*GRAV                                              DYL02210
      GAT(2)=  GAT(2)*GRAV                                              DYL02220
      GAT(3)=  GAT(3)*GRAV                                              DYL02230
      IF (BUG) WRITE (6,9279) T,TL,ACCEL%,GAT(1),GAT(2),GAT(3)          DYL02240
      RETURN                                                            DYL02250
C                                                                       DYL02260
C=======================================                                DYL02270
C= DETERMIN INCRIMENTAL ACCELERATIONS  ===                              DYL02280
C=======================================                                DYL02290
  200 CONTINUE                                                          DYL02300
C------ DETERMINE THE INCRIMENTAL ACCELERATION BETWEEN TIME T AND TL    DYL02310
      DO 201 I=1,3                                                      DYL02320
  201 GA(I)=0                                                           DYL02330
      T=T                                                               DYL02340
      DT=DT                                                             DYL02350
C                                                                       DYL02360
C........CURRENT ACCELERATION                                           DYL02370
      IF (T.GE.TO .AND. T.LE.N*DT ) THEN                                DYL02380
            I=(T-TO)/DT +1                                              DYL02390
            J=I+1                                                       DYL02400
            R=(I*DT+TO-T)/DT                                            DYL02410
            GA(1)=TX(I)*R                                               DYL02420
            GA(2)=TY(I)*R                                               DYL02430
            GA(3)=TZ(I)*R                                               DYL02440
            IF (J.LE.N .AND. R.NE.1 ) THEN                              DYL02450
                 GA(1)=GA(1) + (1-R)*TX(J)                              DYL02460
                 GA(2)=GA(2) + (1-R)*TY(J)                              DYL02470
                 GA(3)=GA(3) + (1-R)*TZ(J)                              DYL02480
            ENDIF                                                       DYL02490
            TXI=TX(I)                                                   DYL02500
            TXJ=TX(J)                                                   DYL02510
      ENDIF                                                             DYL02520
C                                                                       DYL02530
C........PREVIOUS ACCELERATION                                          DYL02540
      IF (TL.GE.TO .AND. TL.LE.N*DT ) THEN                              DYL02550
            I=(TL-TO)/DT +1                                             DYL02560
            J=I+1                                                       DYL02570
            R=(I*DT+TO-TL)/DT                                           DYL02580
            GA(1)=GA(1)-TX(I)*R                                         DYL02590
            GA(2)=GA(2)-TY(I)*R                                         DYL02600
            GA(3)=GA(3)-TZ(I)*R                                         DYL02610
            IF (J.LE.N .AND. R.NE.1 ) THEN                              DYL02620
                 GA(1)=GA(1) - (1-R)*TX(J)                              DYL02630
                 GA(2)=GA(2) - (1-R)*TY(J)                              DYL02640
                 GA(3)=GA(3) - (1-R)*TZ(J)                              DYL02650
            ENDIF                                                       DYL02660
            TXI=TX(I)                                                   DYL02670
            TXJ=TX(J)                                                   DYL02680
      ENDIF                                                             DYL02690
C                                                                       DYL02700
C------ MULTIPLY B BY GRAV.                                             DYL02710
      DO 215 K=1,3                                                      DYL02720
  215   GA(K)=  GA(K)*GRAV                                              DYL02730
C                                                                       DYL02740
C------ CALC. ACCELERATION VECTOR = -MASS * COSINE * GROUND ACCELERATIONDYL02750
      DO 220 I=1,NDOF                                                   DYL02760
      DO 220 K=1,3                                                      DYL02770
  220   P(I)=P(I) - MC(I,K)*GA(K)                                       DYL02780
C                                                                       DYL02790
      IF (BUG) WRITE (6,9279) T,TL,ACCEL%,GA(1),GA(2),GA(3)             DYL02800
 9279 FORMAT(' DYLOA:   T=',1P,G12.4,'   TL=',G12.4,' ACCEL%=',G12.4,/  DYL02810
     &       ' DYLOA:   X=',1P,G12.4,'    Y=',G12.4,'       Z=',G12.4) DYL02820
      IF (BUG) CALL WMATRX(P,NDOF,1,'INCRIMENTAL LOAD')                 DYL02830
      RETURN                                                            DYL02840
 9280 FORMAT(/3X,20X,'     MAXIMUM    AT     TIME     ',1X,             DYL02850
     &'     MINIMUM    AT     TIME          AVERAGE     STANDARD DEV.',1X,DYL02860
     &'     RMS  ')                                                    DYL02870
 9290 FORMAT (3X,A,1P,7G15.5)                                           DYL02880
C                                                                       DYL02890
      END                                                               DYL02900
C*************************************************************************ELE00010
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    ELE00020
C           END DEBUG                                                   ELE00030
C*************************************************************************ELE00040
      SUBROUTINE ELELIB                                                 ELE00050
     &  (IOPT ,LSTTYP ,PRINT ,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,          ELE00060
     &   DUCPAG,                                                        ELE00070
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IDISP,ILOAD,MSTOR)      ELE00080
      LOGICAL KSAME,ERR,FIRST,PRINT,HEAD,AXIAL,BTEST                    ELE00090
      CHARACTER*80 NAME                                                 ELE00100
      DIMENSION RINPUT(100)                                             ELE00110
C     DIMENSION ASAT(12,12),SAT(12,12),STIFF(78)                        ELE00120
C                                                                       ELE00130
      INCLUDE (%COMN)                                                   ELE00140
      IOPT=IOPT                                                         ELE00150
      IW    = I%                                                        ELE00160
      IC    = N%(I%ELE+IELNO)                                           ELE00170
      I%ELE=   I%ELE                                                    ELE00180
      I%ELNO= IELNO                                                     ELE00190
C     IF (.NOT.(IOPT.EQ.1 .OR. IOPT.EQ.9 )) IELTYP = N%(IC)             ELE00200
      IELTYP = N%(IC)                                                   ELE00210
      IF (LSTTYP.EQ.IELTYP) THEN                                        ELE00220
        FIRST=.FALSE.                                                   ELE00230
      ELSE                                                              ELE00240
        FIRST=.TRUE.                                                    ELE00250
        IF (HEAD) WRITE (6,305) TITLE(1),TITLE(2)                       ELE00260
  305   FORMAT ('1 STRUCTURE.....: ',A,/                                ELE00270
     &          '      SOLUTION.....: ',A,/)                           ELE00280
      ENDIF                                                             ELE00290
C                                                                       ELE00300
      MSTOR=0                                                           ELE00310
C                                                                       ELE00320
      IF (IELTYP.EQ.1)  THEN                                            ELE00330
       I%LM=IC+64                                                       ELE00340
       I%KE=IC+281+N%(IC+133)                                           ELE00350
       I%LMKG=IC+121                                                    ELE00360
       IF( MAXELD .LE. 0 ) MAXELD=1                                     ELE00370
       IF (IOPT.NE.3) CALL ELE01                                        ELE00380
     &  (IOPT     ,IUNIT     ,INC       ,FIRST     ,PRINT     ,        ELE00390
     &   NDOF     ,NLOAD     ,MAXNOD    ,NNODE     ,NCOS      ,        ELE00400
     &   N%(I%ID) ,N%(I%IDOF) , %(I%CORD) , %(I%COS ) ,N%(I%JCOS) ,     ELE00410
     &    %(I%CNST) , %(IDISP) , %(ILOAD) ,   IREL    ,BUG       ,     ELE00420
     &   IBUG     , %(I%KGDT) ,N%(I%KGDT+1),GRAV      ,PGEOM     ,     ELE00430
```

```
     &   MAXELD      ,NELD        ,N%(I%IELD) ,%(I%ELD)    ,NAME        ,
     &   STEPID      ,AXIAL       ,IELNO      ,RINPUT      ,%(I%FUB)    ,
     &   EESE        , EPSE       , DAMAGE    ,DUCFAG      ,
     &   N%(IC    ) ,N%(IC+  1) , %(IC+  2) , %(IC+  3) , %(IC+  4) ,
     &    %(IC+  5) , %(IC+  6) , %(IC+  7) , %(IC+  8) , %(IC+  9) ,
     &    %(IC+ 10) , %(IC+ 11) , %(IC+ 12) , %(IC+ 13) , %(IC+ 14) ,
     &    %(IC+ 15) , %(IC+ 16) , %(IC+ 28) , %(IC+ 40) , %(IC+ 52) ,
     &   N%(I%LM  ) , %(IC+ 76) , %(IC+ 85) , %(IC+ 94) , %(IC+103) ,
     &    %(IC+112) ,N%(IC+113) ,N%(IC+114) ,N%(IC+115) ,N%(IC+116) ,
     &   N%(IC+117) , %(IC+118) ,N%(IC+119) ,N%(IC+120) ,N%(I%LMKG) ,
     &   N%(IC+133) , %(IC+134) , %(IC+136) ,N%(IC+280) , %(IC+281) ,
     &    %(I%KE   ) ,
     &    %(IW   ) )
C
      MSTOR =N%(IC+133)
      IELDOF=N%(IC+1)
      I%LMKG=IC+121
      I%EKEG=I%KE+ N%(IC+120)
      KGDOF =N%(IC+119)
C
C
     ELSE IF (IELTYP.EQ.2)  THEN
      AXIAL=.FALSE.
      I%KE=IC+55+N%(IC+12)
      I%LM=IC+40
       CALL ELE02
     &  (IOPT        ,IUNIT       ,NEWK       ,FIRST       ,PRINT       ,
     &   NDOF        ,NLOAD       ,MAXNOD     ,NNODE       ,NCOS        ,
     &   N%(IZID)    ,N%(IZIDOF) ,%(IZCORD)  ,%(IZCOS)    ,N%(IZJCOS) ,
     &   %(IZCNST)  ,%(IDISP )  ,%(IZDVEL)  ,%(ILOAD )  ,%(IZFUB)   ,
     &   IREL        ,BUG         ,IBUG       ,ACCEL       ,N%(IZKGDT+1),
     &   GRAV        ,PGEOM       ,ELSTIC     ,NAME        ,STEPID      ,
     &   IELNO       ,RINPUT      ,EESE       ,EPSE        ,INC         ,
     &   DAMAGE      ,DUCFAG      ,
     &   N%(IC    ) ,N%(IC+  1) ,N%(IC+  2) ,N%(IC+  3) ,N%(IC+  4) ,
     &   N%(IC+  5) , %(IC+  6) , %(IC+  7) , %(IC+  8) , %(IC+  9) ,
     &    %(IC+ 10) , %(IC+ 11) ,N%(IC+ 12) , %(IC+ 13) , %(IC+ 14) ,
     &   N%(IC+ 15) , %(IC+ 16) ,N%(IC+ 40) ,N%(IC+ 52) , %(IC+ 53) ,
     &    %(IC+ 55) , %(I%KE   ) )
      IOPT=IOPT
       MSTOR =N%(IC+12)
       IELDOF=N%(IC+1)
       I%LMKG=0
       I%EKEG=0
       KGDOF =0
C
     ELSE IF (IELTYP.EQ.3)  THEN
      I%KE=IC+167+N%(IC+5)
       CALL ELE03
     &  (IOPT        ,IUNIT       ,NEWK       ,FIRST       ,PRINT       ,
     &   NDOF        ,NLOAD       ,MAXNOD     ,NNODE       ,NCOS        ,
     &   N%(IZID)    ,N%(IZIDOF) ,%(IZCORD)  ,%(IZCOS)    ,N%(IZJCOS) ,
     &   %(IZCNST)  ,%(IDISP )  ,%(IZDVEL)  ,%(ILOAD )  ,%(IZFUB)   ,
     &   IREL        ,BUG         ,IBUG       ,ACCEL       ,N%(IZKGDT+1),
     &   GRAV        ,PGEOM       ,ELSTIC     ,NAME        ,STEPID      ,
     &   IELNO       ,RINPUT      ,AXIAL      ,EESE        ,EPSE        ,
     &   INC         ,DAMAGE      ,DUCFAG     ,
     &   N%(IC    ) ,N%(IC+  1) ,N%(IC+  2) ,N%(IC+  3) ,N%(IC+  4) ,
     &   N%(IC+  5) ,N%(IC+  6) ,N%(IC+  7) ,N%(IC+  8) ,N%(IC+  9) ,
     &   N%(IC+ 10) ,N%(IC+ 11) ,N%(IC+ 12) , %(IC+ 13) , %(IC+ 14) ,
     &    %(IC+ 17) , %(IC+ 20) , %(IC+ 23) , %(IC+ 26) , %(IC+ 29) ,
     &    %(IC+ 32) ,N%(IC+ 33) ,N%(IC+ 34) ,N%(IC+ 58) , %(IC+ 82) ,
     &    %(IC+154) , %(IC+155) , %(IC+156) , %(IC+157) , %(IC+158) ,
     &   N%(IC+159) , %(IC+160) , %(IC+161) , %(IC+167) , %(I%KE   ) )
      MSTOR =N%(IC+5)
      IELDOF=N%(IC+1)
      I%LM=IC+34
      I%LMKG=IC+58
      I%EKEG=IC+167+N%(IC+5) + N%(IC+33)-1
      KGDOF =N%(IC+159)
C
     ELSE IF (IELTYP.EQ.8)  THEN
      AXIAL=.FALSE.
      I%KE=IC+55+N%(IC+12)
      I%LM=IC+40
       CALL ELE08
     &  (IOPT        ,IUNIT       ,NEWK       ,FIRST       ,PRINT       ,
     &   NDOF        ,NLOAD       ,MAXNOD     ,NNODE       ,NCOS        ,
     &   N%(IZID)    ,N%(IZIDOF) ,%(IZCORD)  ,%(IZCOS)    ,N%(IZJCOS) ,
     &   %(IZCNST)  ,%(IDISP )  ,%(IZDVEL)  ,%(ILOAD )  ,%(IZFUB)   ,
     &   IREL        ,BUG         ,IBUG       ,ACCEL       ,N%(IZKGDT+1),
     &   GRAV        ,PGEOM       ,ELSTIC     ,NAME        ,STEPID      ,
     &   IELNO       ,RINPUT      ,EESE       ,EPSE        ,INC         ,
     &   DAMAGE      ,DUCFAG      ,
     &   N%(IC    ) ,N%(IC+  1) ,N%(IC+  2) ,N%(IC+  3) ,N%(IC+  4) ,
     &   N%(IC+  5) , %(IC+  6) , %(IC+  7) , %(IC+  8) , %(IC+  9) ,
     &    %(IC+ 10) , %(IC+ 11) ,N%(IC+ 12) , %(IC+ 13) , %(IC+ 14) ,
     &   N%(IC+ 15) , %(IC+ 16) ,N%(IC+ 40) ,N%(IC+ 52) , %(IC+ 53) ,
     &    %(IC+ 55) , %(I%KE   ) )
      IOPT=IOPT
       MSTOR =N%(IC+12)
       IELDOF=N%(IC+1)
       I%LMKG=0
       I%EKEG=0
       KGDOF =0
C
     ELSE IF (IELTYP.EQ.9)  THEN
      I%LM=IC+73
      I%KE=IC+458+N%(IC+141)
      I%LMKG=IC+129
      IF( MAXELD .LE. 0 ) MAXELD=1
      CALL ELE09
&(IOPT,IUNIT,INC,NEWK,FIRST,PRINT,NDOF,NLOAD,MAXNOD,NNODE,NCOS,
& N%(IZID),ELSTIC,N%(IZIDOF),%(IZCORD),%(IZCOS )  ,N%(IZJCOS) ,
&    %(IZCNST) , %(IDISP)  , %(ILOAD)  ,   IREL    ,BUG         ,
&    IBUG      , ACCEL     ,N%(I%KGDT+1),GRAV      ,PGEOM       ,
&    MAXELD    ,NELD       ,N%(I%IELD) ,%(I%ELD)   ,NAME        ,
&    STEPID    ,AXIAL      ,IELNO      ,RINPUT     ,%(I%FUB)    ,
&    EESE , EPSE, DAMAGE,DUCFAG     ,N%(IC)     ,N%(IC+  1) ,
& N%(IC-2),N%(IC+3),N%(IC+4),N%(IC+5),N%(IC+6),N%(IC+7),%(IC+8),
& %(IC+9),%(IC+10),%(IC+11),%(IC+12),%(IC+13),%(IC+14),%(IC+15),
& %(IC+16),%(IC+17),N%(IC+18),N%(IC+19),N%(IC+20),N%(IC+21),
& N%(IC+22),N%(IC+23),%(IC+24),%(IC+25),%(IC+37),%(IC+49),%(IC+61),
&   N%(I%LM  ) , %(IC+ 85) , %(IC+ 94) , %(IC+103) , %(IC+112) ,
&   N%(IC+121) ,N%(IC+122) ,N%(IC+123) ,N%(IC+124) ,
&   N%(IC+125) , %(IC+126) ,N%(IC+127) ,N%(IC+128) ,N%(I%LMKG) ,
&   N%(IC+141) , %(IC+142) , %(IC+144) ,N%(IC+288) ,N%(IC+289) ,
&   N%(IC+290) ,N%(IC+291) ,N%(IC+292) ,N%(IC+293) ,N%(IC+294) ,
&   N%(IC+295),%(IC+296),%(IC+297),%(IC+298),%(IC+299),%(IC+300),
&    %(IC+301),%(IC+302),%(IC+303),%(IC+304),%(IC+305),%(IC+306),
& %(IC+307),%(IC+308),%(IC+309),%(IC+310),%(IC+311),%(IC+312),
& %(IC+313),%(IC+314),%(IC+458),%(I%KE),%(IW))
C
      MSTOR =N%(IC+141)
      IELDOF=N%(IC+1)
      I%LMKG=IC+129
      I%EKEG=I%KE+ N%(IC+128)
      KGDOF =N%(IC+127)
C
     ELSE IF (IELTYP.EQ.12)  THEN
      AXIAL=.FALSE.
      I%LM=IC+50
      I%KE=IC+540+N%(IC+104)
      IF( MAXELD .LE. 0 ) MAXELD=1
      CALL ELE12
&  (IOPT        ,IUNIT       ,INC,NEWK   ,FIRST       ,PRINT       ,
&   NDOF        ,NLOAD       ,MAXNOD     ,NNODE       ,NCOS        ,
& N%(IZID),ELSTIC,N%(IZIDOF),%(IZCORD) ,%(IZCOS )  ,N%(IZJCOS) ,
&    %(IZCNST) ,  %(IDISP)   ,  %(ILOAD)   ,   IREL    ,BUG         ,
&   IBUG        , ACCEL     ,N%(I%KGDT+1),GRAV      ,PGEOM       ,
&   MAXELD      ,NELD       ,N%(I%IELD) ,%(I%ELD)   ,NAME        ,
&   STEPID      ,IELNO      ,RINPUT     ,%(I%FUB)   ,
```

```
     &   EESE        , EPSE,DAMAGE,DUCFAG     ,N%(IC)      ,N%(IC+  1) ,
     &   %(IC+  2) , %(IC+ 14) , %(IC+ 26) , %(IC+ 38) ,
     &   N%(I%LM  ) , %(IC+ 62) , %(IC+ 71) , %(IC+ 80) , %(IC+ 89) ,
     &   N%(IC+ 98) ,N%(IC+ 99) ,N%(IC+100) ,N%(IC+101) ,
     &   N%(IC+102) , %(IC+103) ,
     &   N%(IC+104) , %(IC+105) , %(IC+107) , %(IC+251) ,%(IC+395)  ,
     &   N%(IC+539) , %(IC+540) , %(I%KE   )    )
C
       MSTOR =N%(IC+104)
       IELDOF=N%(IC+1)
       I%LMKG=0
       I%EKEG=0
       KGDOF =0
C
      ELSE IF (IELTYP.EQ.14)  THEN
       I%KE=IC+167+N%(IC+5)
        CALL ELE14
     &  (IOPT        ,IUNIT       ,NEWK       ,FIRST       ,PRINT       ,
     &   NDOF        ,NLOAD       ,MAXNOD     ,NNODE       ,NCOS        ,
     &   N%(IZID)    ,N%(IZIDOF) ,%(IZCORD)  ,%(IZCOS)    ,N%(IZJCOS) ,
     &   %(IZCNST)  ,%(IDISP )  ,%(IZDVEL)  ,%(ILOAD )  ,%(IZFUB)   ,
     &   IREL        ,BUG         ,IBUG       ,ACCEL       ,N%(IZKGDT+1),
     &   GRAV        ,PGEOM       ,ELSTIC     ,NAME        ,STEPID      ,
     &   IELNO       ,RINPUT      ,AXIAL      ,EESE        ,EPSE        ,
     &   INC         ,DAMAGE      ,DUCFAG     ,
     &   N%(IC    ) ,N%(IC+  1) ,N%(IC+  2) ,N%(IC+  3) ,N%(IC+  4) ,
     &   N%(IC+  5) ,N%(IC+  6) ,N%(IC+  7) ,N%(IC+  8) ,N%(IC+  9) ,
     &   N%(IC+ 10) ,N%(IC+ 11) ,N%(IC+ 12) , %(IC+ 13) , %(IC+ 14) ,
     &    %(IC+ 17) , %(IC+ 20) , %(IC+ 23) , %(IC+ 26) , %(IC+ 29) ,
     &    %(IC+ 32) ,N%(IC+ 33) ,N%(IC+ 34) ,N%(IC+ 58) , %(IC+ 82) ,
     &    %(IC+154) ,N%(IC+155) ,N%(IC+156) ,N%(IC+157) , %(IC+158) ,
     &   N%(IC+159) , %(IC+160) , %(IC+161) , %(IC+167) , %(I%KE   ) )
       MSTOR =N%(IC+5)
       IELDOF=N%(IC+1)
       I%LM=IC+34
       I%LMKG=IC+58
       I%EKEG=IC+167+N%(IC+5) + N%(IC+33)-1
       KGDOF =N%(IC+159)
      ELSE
         WRITE (6,10) IELNO
         ERR=.TRUE.
   10    FORMAT(' ELEMENT ',I6,' IS NOT AVAILABLE' )
      ENDIF
      LSTTYP=IELTYP
      IOPT=IOPT
C
      RETURN
      END
C*********************************************************************
C          DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE
C          END DEBUG
C*********************************************************************
      SUBROUTINE ELELOA(BUG,NF,NELMT,MAXELD,NELD,NDOF,NLOAD,FORCE,IELD,
     &   ELD,IDISP,ILOAD ,NAME,TITLE,HEAD)
C
C NOTE ON ARGUMENTS: THE LOAD MATRIX FORCE IS PASSED FROM THE CALLING
C  ROUTINE.  FORCE IS EQUIVALENCED TO THE % MATRIX BY THE CALLING ROUTINE
C  IE. %(ILOAD)=FORCE(1,1), WHERE ILOAD IS THE OFFSET IN THE % MATRIX.
C  BOTH THE CALLING ROUTINE AND SUBROUTINE ELE_LIB USE THE % MATRIX
C
      DIMENSION FORCE(NDOF,NLOAD),IELD(5,MAXELD),ELD(12,MAXELD)
      DIMENSION RINPUT(100)
      LOGICAL FIRST,HEAD,FALSE
      LOGICAL AXIAL,BUG
      LOGICAL BTEST
      CHARACTER*(*) TITLE(2)
      CHARACTER*(*) NAME
      CHARACTER*20 TYPE,DIRI
      CHARACTER*40 DIR
      CHARACTER*1 CX(6)
      FIRST=.TRUE.
    5 FORMAT ('1 STRUCTURE.....: ',A,/
     &        ' SOLUTION.....: ',A,/)
    6 FORMAT (/'    ELEMENT LOADS    ',//
     &         '  ================== ',//
     & 1X,' LOAD ELEM  GEN   INC   GROUP & DIRECTION    VALUES.... '/)
C              12345123451234512345XX12345678901234567890
C**********************
C* INPUT LOADING DATA **
C**********************
      N=0
   10 N=N+1
      IF (N.GT.MAXELD) GO TO 100
   20 READ (5,*) (IELD(I,N),I=1,4),TYPE,DIRI,ELD(1,N)
      DIR=TYPE//DIRI
      IF (INDEX(DIR,'END').NE.0) GO TO 100
      NELD=N
      IF (FIRST)  THEN
          IF (HEAD) WRITE (6,5) TITLE(1),TITLE(2)
          WRITE (6,6)
          FIRST=.FALSE.
      ENDIF
C
C------- PROCESS LOAD GROUPS AND READ ADDITIONAL INFO...
      IF (INDEX(DIR,'CONC').NE.0) THEN
         KODE=100
         J=2
         BACKSPACE(5)
         READ (5,*) (IELD(I,N),I=1,4),TYPE,DIRI,(ELD(I,N),I=1,J)
         DIR=TYPE//DIRI
         CALL CKRNG(ELD(2,N),0.,1.,'DISTANCE')
      ELSE IF (INDEX(DIR,'UNIF').NE.0) THEN
         J=1
         KODE=200
      ELSE IF (INDEX(DIR,'VARR').NE.0) THEN
         KODE=300
         J=4
         BACKSPACE(5)
         READ (5,*) (IELD(I,N),I=1,4),TYPE,DIRI,(ELD(I,N),I=1,J)
         DIR=TYPE//DIRI
         CALL CKRNG(ELD(3,N),0.      ,ELD(4,N),'DISTANCE')
         CALL CKRNG(ELD(4,N),ELD(3,N),1.      ,'DISTANCE')
      ELSE IF (INDEX(DIR,'FEM').NE.0) THEN
         KODE=400
         J=12
         BACKSPACE(5)
         READ (5,*) (IELD(I,N),I=1,4),TYPE,DIRI,(ELD(I,N),I=1,J)
         DIR=TYPE//DIRI
      ELSE
         WRITE(6,30) 'INVALID LOAD TYPE',
     &              (IELD(I,N),I=1,4),DIR,ELD(1,N)
         GO TO 20
      ENDIF
      IF (INDEX(DIR,'FX').NE.0) THEN
         KODE=KODE+1
      ELSE IF (INDEX(DIR,'FY').NE.0) THEN
         KODE=KODE+2
      ELSE IF (INDEX(DIR,'FZ').NE.0) THEN
         KODE=KODE+3
      ELSE IF (INDEX(DIR,'MX').NE.0) THEN
         KODE=KODE+4
      ELSE IF (INDEX(DIR,'MY').NE.0) THEN
         KODE=KODE+5
      ELSE IF (INDEX(DIR,'MZ').NE.0) THEN
         KODE=KODE+6
      ELSE IF (KODE.NE.400) THEN
         WRITE(6,30) 'INVALID LOAD DIRECTION',
     &              (IELD(I,N),I=1,4),DIR,(ELD(1,N),I=1,J)
         GO TO 20
      ENDIF
      IELD(5,N)=KODE
      WRITE(6,40) (IELD(I,N),I=1,4),DIR,(ELD(I,N),I=1,J)
```

```
      GO TO 10                                                          ELE00950
C                                                                       ELE00960
   30 FORMAT (' ',A, ' --- LOAD IS SKIPPED'/                            ELE00970
     &        1X,4I5,1X,A20,1P,6G14.6,:/(42X,6G14.6) )                  ELE00980
   40 FORMAT (1X,4I5,1X,A20,1P,6G14.6,:/(42X,6G14.6) )                  ELE00990
C                                                                       ELE01000
C------- RETURN IF NO ELEMENT LOADS WERE INPUT                          ELE01010
  100 IF (FIRST)  RETURN                                                ELE01020
C                                                                       ELE01030
C------- LOOP TO GENERATE MEMBER LOADS FOR EACH ELEMENT                 ELE01040
      IOPT=6                                                            ELE01050
      DO 101 IELNO=1,NELMT                                              ELE01060
        FALSE=.FALSE.                                                   ELE01070
  101   CALL ELELIB                                                     ELE01080
     &  (IOPT  ,LSTTYP ,FALSE ,IREL,FALSE, NAME ,EESE,EPSE,DAMAGE,      ELE01090
     &   DUCFAG,                                                        ELE01100
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IDISP,ILOAD,MSTOR)      ELE01110
C                                                                       ELE01120
C------- PRINT LOADS AT EACH DOF                                        ELE01130
      IF (BUG) THEN                                                     ELE01140
        N5=NDOF-5                                                       ELE01150
        DO 120 L=1,NLOAD                                                ELE01160
          IF (HEAD) WRITE (6,5) TITLE(1),TITLE(2)                       ELE01170
          WRITE (6,130) L                                               ELE01180
          DO 120 I=1,N5,5                                               ELE01190
          I5=MIN((I+4),NDOF)                                            ELE01200
          DO 110 J=I,I5                                                 ELE01210
            K=J-I+1                                                     ELE01220
            CX(K)=' '                                                   ELE01230
  110       IF (J.GT.NF) CX(K)='*'                                      ELE01240
            WRITE (6,140) (J,FORCE(J,L),CX(J-I+1),J=I,I5)               ELE01250
  120     CONTINUE                                                      ELE01260
      ENDIF                                                             ELE01270
C                                                                       ELE01280
  130 FORMAT (' GLOBAL JOINT FORCES DUE TO ELEMENT LOADS ',3X,          ELE01290
     &        '   LOADING #',I5,10X,                                    ELE01300
     &        ' NOTE: * DENOTES SUPPORT DISPLACEMENT'/                  ELE01310
     &        ' ------------------------------------------'//           ELE01320
     &        5X,5(' DOF',7X,'LOAD    ')/)                            ELE01330
  140   FORMAT (5X,5(0P,I5,1P,G14.6,A) )                                ELE01340
C                                                                       ELE01350
      RETURN                                                            ELE01360
      END                                                               ELE01370
$PROCESS SDUMP OPT(3) GOSTMT XREF                                       ELE00010
C=======================================================================ELE00020
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    ELE00030
C           END DEBUG                                                   ELE00040
C=======================================================================ELE00050
      SUBROUTINE ELE01                                                  ELE00060
     &  (IOPT       ,IUNIT      ,INC        ,FIRST      ,PRINT      ,    ELE00070
     &   NDOF       ,NLOAD      ,MAXNOD     ,NNODE      ,NCOS       ,    ELE00080
     &   ID         ,IDOF       ,COORD      ,COSINE     ,JTCOS      ,    ELE00090
     &   CONST      ,DISPL      ,FORCE      ,IREL       ,BUG        ,    ELE00100
     &   IBUG       ,ACCEL      ,KGDATA     ,GRAV       ,PGEOM      ,    ELE00110
     &   MAXELD     ,NELD       ,IELD       ,ELD        ,NAME       ,    ELE00120
     &   STEPID     ,AXIAL      ,IELNO      ,RINPUT     ,FUB        ,    ELE00130
     &   EESE       , EPSE      ,DAMAGE     ,DUCFAG     ,             ,  ELE00140
     &   IELTYP     ,IELDOF     ,PKG        ,H          ,Q          ,    ELE00150
     &   AI         ,AQ         ,C          ,BI         ,BJ         ,    ELE00160
     &   S22        ,S26        ,S212       ,S33        ,S35        ,    ELE00170
     &   S311       ,R          ,RT         ,F          ,FT         ,    ELE00180
     &   LM         ,CTS        ,CTE        ,BS         ,BE         ,    ELE00190
     &   D          ,NODEI      ,NODEJ      ,JOINTI     ,JOINTJ     ,    ELE00200
     &   REL        ,LENGTH     ,KGDOF      ,LMKG       ,LMKG       ,    ELE00210
     &   ISE        ,ENGDAT     ,FMAX       ,MAT        ,SE,             ELE00220
     &   STIFF      ,                                                    ELE00230
     &   FEM        )                                                    ELE00240
C                                                                       ELE00250
      IMPLICIT REAL(A-H,O-Z)                                            ELE00260
C                                                                       ELE00270
      COMMON /RSA/ICOMN,RSAF(12)                                        ELE00280
      REAL ICOMN                                                        ELE00290
C                                                                       ELE00300
      LOGICAL ERR,FIRST,PRINT,BUG,FEMFLG,FALSE,AXIAL,BTEST             ELE00310
      CHARACTER*80 NAME                                                 ELE00320
      CHARACTER*(*) STEPID                                              ELE00330
      CHARACTER*6  IRELC                                                ELE00340
      DIMENSION KGDATA(3)                                               ELE00350
      DIMENSION  IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS)        ELE00360
      DIMENSION CONST(MAXNOD,6),   ID(MAXNOD)  ,JTCOS(MAXNOD)           ELE00370
      DIMENSION ASAT(12,12),SAT(12,12),STIFF(156)                       ELE00380
C     DIMENSION A(12,12),  S(12,12),LM(12),LMT(12),LMKG(12),FMAX(12)    ELE00390
      DIMENSION A(12,12),  S(12,12),LM(12),LMT(12),LMKG(12),FMAX(144)   ELE00400
      DIMENSION CTS( 3, 3),CTE( 3, 3),COSLOC(3,3),VX(3),VY(3),VZ(3)     ELE00410
      DIMENSION BS( 3, 3),BE( 3, 3),SE(15),ENGDAT(2)                    ELE00420
      DIMENSION R(12),DISPL(NDOF,NLOAD),F(12),RT(12),FT(12),FUB(NDOF)   ELE00430
      DIMENSION FORCE(NDOF,NLOAD),IELD(5,MAXELD),ELD(12,MAXELD)         ELE00440
      DIMENSION ASA(12,12),RINPUT(100),FEM(12,NLOAD),GFEM(12)           ELE00450
      DIMENSION DUCT(3),EXCR(6)                                         ELE00460
      DIMENSION V(12),VL(12)                                            ELE00470
      REAL LENGTH                                                       ELE00480
      REAL IX,IY,IZ,KIIZ,KJJZ,KIJZ,KIIY,KJJY,KIJY,KTI,TYJ              ELE00490
      INTEGER REL,RELT                                                  ELE00500
      DIMENSION MD(13)                                                  ELE00510
      DATA MD/1,2,4,7,11,16,22,29,37,46,56,67,79/                       ELE00520
C                                                                       ELE00530
C-----------------------------------------------------------------------ELE00540
C VARIABLES:                                                            ELE00550
C-----------------------------------------------------------------------ELE00560
C---------- GLOBAL VARIABLES                                           ELE00570
C IOPT   = 1, INITIALIZE BEAM COLUMN ELEMENT                            ELE00580
C        = 2, CALCULATE GEOMETRIC STIFFNESS                             ELE00590
C        = 3, FORM STIFFNESS                                            ELE00600
C        = 4, CALCULATE INCREMENTAL FORCES                              ELE00610
C        = 5, CALCULATE TOTAL FORCES AND ENERGIES                       ELE00620
C IUNIT  = OUTPUT FILE UNIT # FOR PRINTING OUTPUT                       ELE00630
C ERR    = ERR OR FLAG, IF AN ERR OR HAS OCCURED, ERR=.TRUE.            ELE00640
C FIRST  = FLAG, FIRST=.TRUE., PRINT HEADERS                            ELE00650
C PRINT  = FLAG, PRINT=.TRUE., PRINT DATA                               ELE00660
C NDOF   = # OF GLOBAL DOF                                              ELE00670
C NLOAD  = # OF LOAD COMBINATIONS                                       ELE00680
C MAXNOD = # ROW DIMENSION OF NODE ARRAY                                ELE00690
C NNODE  = # OF NODES                                                   ELE00700
C ID     = ARRAY OF EXTERNAL NODE NUMBERS                               ELE00710
C IDOF   = ARRAY OF DEGREES OF FREEDOM                                  ELE00720
C COORD  = ARRAY OF COORDINATES                                         ELE00730
C COSINE = ARRAY OF DIRECTION COSINES OF NODES                          ELE00740
C CONST  = ARRAY OF CONSTRAINT TRANSFORMATIONS                          ELE00750
C DISPL  = GLOBAL DISPLACEMENT MATRIX                                   ELE00760
C IZMAT  = LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR           ELE00770
C NAME   = ELEMENT NAME                                                 ELE00780
C IELNO  = ELEMENT NUMBER                                               ELE00790
C RINPUT = INPUT DATA                                                   ELE00800
C STIFF  = OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)             ELE00810
C LM     = OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX              ELE00820
C---------- ELEMENT VARIABLES                                          ELE00830
C H,Q,AI,AJ,C,BI,BJ,S22,S26,S212,S33,S35,S311,D                         ELE00840
C        = INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX         ELE00850
C PKG    = INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX...         ELE00860
C R      = INCRIMENTAL DISPLACEMENTS                                    ELE00870
C RT     = TOTAL DISPLACEMENTS                                          ELE00880
C F      = INCRIMENTAL FORCES                                           ELE00890
C FT     = TOTAL FORCES                                                 ELE00900
C LM     = LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                     ELE00910
C CTS    = LOCAL TO GLOBAL ROTATION MATRIX START                        ELE00920
C CTE    = LOCAL TO GLOBAL ROTATION MATRIX END                          ELE00930
C BS     = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX STARTELE00940
C BE     = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END  ELE00950
C STIFF  = ELEMENT STIFFNESS                                            ELE00960
C IELTYP = ELEMENT TYPE = 10                                            ELE00970
C NODEI  = EXTERNAL JOINT NUMBER, END I                                 ELE00980
C NODEJ  = EXTERNAL JOINT NUMBER, END J                                 ELE00990
C JOINTI = INTERNAL JOINT NUMBER, END I                                 ELE01000
C JOINTJ = INTERNAL JOINT NUMBER, END J                                 ELE01010
C---------- TEMPRORY VARIABLES                                          ELE01020
C VX     = VECTOR DEFINING THE ELEMENT X AXIS                           ELE01030
C VY     = VECTOR DEFINING THE ELEMENT Y AXIS                           ELE01040
C COSLOC = LOCAL COSINE MATRIX                                          ELE01050
C A      = GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                  ELE01060
C S      = ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART   ELE01070
C SAT    = PRODUCT OF S*A                                               ELE01080
C ASAT   = PRODUCT OF A*SAT                                             ELE01090
C-----------------------------------------------------------------------ELE01100
C                                                                       ELE01110
C------- CHOOSE OPTION                                                  ELE01120
        IF (IOPT.LT.1 .OR.IOPT.GT.14)                                   ELE01130
     &  WRITE (6,*) 'INVALID OPTION IN ELE-01, IOPT=',IOPT              ELE01140
C                                                                       ELE01150
        GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,      ELE01160
     &                                          1300,1400),IOPT         ELE01170
C                                                                       ELE01180
  100 CONTINUE                                                          ELE01190
C                                                                       ELE01200
C------- ZERO STRAIN ENERGY                                             ELE01210
        ENGDAT(1) = 0                                                   ELE01220
        ENGDAT(2) = 0                                                   ELE01230
C                                                                       ELE01240
        IF ( BTEST(IBUG,7) ) THEN                                       ELE01250
           WRITE (6,*) NAME                                             ELE01260
           CALL WMATRX(RINPUT,10, 1,'RINPUT')                           ELE01270
        ENDIF                                                           ELE01280
C                                                                       ELE01290
C------- INTERPERTATE INPUT DATA                                        ELE01300
        IELTYP= 1                                                       ELE01310
        MAT   = RINPUT( 1)                                              ELE01320
        NODEI = RINPUT( 2)                                              ELE01330
        NODEJ = RINPUT( 3)                                              ELE01340
        VY(1) = RINPUT( 4)                                              ELE01350
        VY(2) = RINPUT( 5)                                              ELE01360
        VY(3) = RINPUT( 6)                                              ELE01370
        XS    = RINPUT( 7)                                              ELE01380
        XE    = RINPUT( 8)                                              ELE01390
        PKG   = RINPUT( 9)                                              ELE01400
        RELT  = RINPUT(10)                                              ELE01410
C                                                                       ELE01420
C------- GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S          ELE01430
        JOINTI=IQUICK(NODEI,ID,NNODE)                                   ELE01440
        JOINTJ=IQUICK(NODEJ,ID,NNODE)                                   ELE01450
C       WRITE (6,*) 'ID:',ID                                            ELE01460
C                                                                       ELE01470
C------- PRINT DATA FOR PLOTTING                                        ELE01480
        IF (BTEST(IBUG,2)) THEN                                         ELE01490
          WRITE(7,10) NODEJ, NODEI,                                     ELE01500
     &          (COORD(JOINTJ,I),I=1,3),(COORD(JOINTI,J),J=1,3)         ELE01510
   10   FORMAT (I5,1X,I5,1P,6G12.4)                                     ELE01520
        ENDIF                                                           ELE01530
C                                                                       ELE01540
C-----------------------------------------------------------------------ELE01550
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                            ELE01560
C-----------------------------------------------------------------------ELE01570
C------- DEFINE VECTOR X, LENGTH                                        ELE01580
        VX(1)=COORD(JOINTJ,1)-COORD(JOINTI,1)                           ELE01590
        VX(2)=COORD(JOINTJ,2)-COORD(JOINTI,2)                           ELE01600
        VX(3)=COORD(JOINTJ,3)-COORD(JOINTI,3)                           ELE01610
        LENGTH=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)-XS-XE                   ELE01620
        IF (LENGTH.LE.0) THEN                                           ELE01630
           WRITE(6,101) IELNO,JOINTI,JOINTJ,LENGTH                      ELE01640
  101      FORMAT(1X,20('*'),'ER',' ROR - LENGTH IS LE 0',/             ELE01650
     &     5X,'IELNO=',I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,             ELE01660
     &     5X,'LENGTH',1P,G15.6/                                        ELE01670
     &     5X,'REVISE INPUT.............................')             ELE01680
           LENGTH=1.0                                                   ELE01690
        ENDIF                                                           ELE01700
        IF ( BTEST(IBUG,7) ) THEN                                       ELE01710
           WRITE (6,*) 'JOINTS',JOINTI,JOINTJ                           ELE01720
           WRITE (6,*) 'VX',VX                                          ELE01730
           WRITE (6,*) 'VY',VY                                          ELE01740
           WRITE (6,*) 'LENGTH  ',LENGTH                                ELE01750
        ENDIF                                                           ELE01760
C------- GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES CT AND B          ELE01770
        ICS=JTCOS(JOINTI)                                               ELE01780
        ICE=JTCOS(JOINTJ)                                               ELE01790
        CALL ROTXYZ(VX,VY,COSINE(1,1,ICS),CTS, XS ,0.,0.,BS,            ELE01800
     &       CONST(JOINTI,1),MAXNOD)                                    ELE01810
        CALL ROTXYZ(VX,VY,COSINE(1,1,ICE),CTE,-XE ,0.,0.,BE,            ELE01820
     &       CONST(JOINTJ,1),MAXNOD)                                    ELE01830
C       WRITE(6,*)'LENGTH= ',LENGTH                                     ELE01840
        IF ( BTEST(IBUG,7) ) THEN                                       ELE01850
           CALL WMATRX( CTS  , 3, 3,'CTS   ')                           ELE01860
           CALL WMATRX( CTE  , 3, 3,'CTE   ')                           ELE01870
        ENDIF                                                           ELE01880
C                                                                       ELE01890
C------- GET GLOBAL CONSTRAINT MATRIX BS, BE                            ELE01900
C       BS(1,1)=CONST(JOINTI,1)*CTS(2,1)+CONST(JOINTI,2)*CTS(3,1)+BS(1,1) ELE01910
C       BS(1,2)=CONST(JOINTI,1)*CTS(2,2)+CONST(JOINTI,2)*CTS(3,2)+BS(1,2) ELE01920
C       BS(1,3)=CONST(JOINTI,1)*CTS(2,3)+CONST(JOINTI,2)*CTS(3,3)+BS(1,3) ELE01930
C       BS(2,1)=CONST(JOINTI,3)*CTS(1,1)+CONST(JOINTI,4)*CTS(3,1)+BS(2,1) ELE01940
C       BS(2,2)=CONST(JOINTI,3)*CTS(1,2)+CONST(JOINTI,4)*CTS(3,2)+BS(2,2) ELE01950
C       BS(2,3)=CONST(JOINTI,3)*CTS(1,3)+CONST(JOINTI,4)*CTS(3,3)+BS(2,3) ELE01960
C       BS(3,1)=CONST(JOINTI,5)*CTS(1,1)+CONST(JOINTI,6)*CTS(2,1)+BS(3,1) ELE01970
C       BS(3,2)=CONST(JOINTI,5)*CTS(1,2)+CONST(JOINTI,6)*CTS(2,2)+BS(3,2) ELE01980
C       BS(3,3)=CONST(JOINTI,5)*CTS(1,3)+CONST(JOINTI,6)*CTS(2,3)+BS(3,3) ELE01990
C                                                                       ELE02000
C       BE(1,1)=CONST(JOINTJ,1)*CTE(2,1)+CONST(JOINTJ,2)*CTE(3,1)+BE(1,1) ELE02010
C       BE(1,2)=CONST(JOINTJ,1)*CTE(2,2)+CONST(JOINTJ,2)*CTE(3,2)+BE(1,2) ELE02020
C       BE(1,3)=CONST(JOINTJ,1)*CTE(2,3)+CONST(JOINTJ,2)*CTE(3,3)+BE(1,3) ELE02030
C       BE(2,1)=CONST(JOINTJ,3)*CTE(1,1)+CONST(JOINTJ,4)*CTE(3,1)+BE(2,1) ELE02040
C       BE(2,2)=CONST(JOINTJ,3)*CTE(1,2)+CONST(JOINTJ,4)*CTE(3,2)+BE(2,2) ELE02050
C       BE(2,3)=CONST(JOINTJ,3)*CTE(1,3)+CONST(JOINTJ,4)*CTE(3,3)+BE(2,3) ELE02060
C       BE(3,1)=CONST(JOINTJ,5)*CTE(1,1)+CONST(JOINTJ,6)*CTE(2,1)+BE(3,1) ELE02070
C       BE(3,2)=CONST(JOINTJ,5)*CTE(1,2)+CONST(JOINTJ,6)*CTE(2,2)+BE(3,2) ELE02080
C       BE(3,3)=CONST(JOINTJ,5)*CTE(1,3)+CONST(JOINTJ,6)*CTE(2,3)+BE(3,3) ELE02090
C                                                                       ELE02100
        IF ( BTEST(IBUG,7) ) THEN                                       ELE02110
           CALL WMATRX(BS, 3, 3,'BS    ')                               ELE02120
           CALL WMATRX(BE, 3, 3,'BE    ')                               ELE02130
        ENDIF                                                           ELE02140
C                                                                       ELE02150
C------- ZERO MATRICIES                                                 ELE02160
        DO 40 I=1,12                                                    ELE02170
          FMAX(I)=0                                                     ELE02180
          F (I)=0                                                       ELE02190
          FT(I)=0                                                       ELE02200
          R (I)=0                                                       ELE02210
          RT(I)=0                                                       ELE02220
          DO 40 J=1,12                                                  ELE02230
            S(I,J)=0.                                                   ELE02240
   40       A(I,J)=0.                                                   ELE02250
C                                                                       ELE02260
C------- ASSEMBLE TRANSFORMATION MATRIX A                               ELE02270
        DO 50 I=1,3                                                     ELE02280
        DO 50 J=1,3                                                     ELE02290
          A(I   ,J  )=CTS(I,J)                                          ELE02300
          A(I+3,J+3)=CTS(I,J)                                           ELE02310
          A(I+6,J+6)=CTE(I,J)                                           ELE02320
          A(I+9,J+9)=CTE(I,J)                                           ELE02330
          A(I+3,J  )=BS(I,J)                                            ELE02340
   50     A(I+9,J+6)=BE(I,J)                                           ELE02350
        IF ( BTEST(IBUG,7) ) CALL WMATRX( A    ,12,12,' A    ')         ELE02360
C                                                                       ELE02370
C------- GET MATERIAL PROPERTIES                                        ELE02380
         FALSE=.FALSE.                                                  ELE02390
         LSTTYP=0                                                       ELE02400
         CALL MATLIB                                                    ELE02410
```

```
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,
     &  FT     ,F      ,RT    ,R    ,V     ,VL    ,LELEM ,LMAT  ,
     &  MAT    ,MATTYP,SE,IELNO,DAMAGE)
C
      IF (MATTYP.NE.1) THEN
        WRITE (6,190) IELNO,MATTYP,NODEI,NODEJ
  190   FORMAT (' ELEMENT: ',I6,' INCOMPATIBLE MATERIAL PROP.#',I5,
     &  ' START JOINT:',I5,' END JOINT:',I5,/
     &  ' THE ELEMENT IS REJECTED AND ER','ROR CHECKING CONTINUES')
        ERR=.TRUE.
        GO TO 191
      ENDIF
C
      EAX   =SE( 1)
      GAY   =SE( 2)
      GAZ   =SE( 3)
      GIX   =SE( 4)
      EIY   =SE( 5)
      EIZ   =SE( 6)
C
C-----------------------------------------------------------------
C  DETERMINE THE MEMBER RELEASE CODES
C-----------------------------------------------------------------
C     RELT  IS AN INPUT MEMBER RELEASE CODE
C     REL   IS THE INTERNAL MEMBER RELEASE CODE
C           BIT 6 AXIAL -          TRUE = AXIAL DEFORMATION CONSIDERED
C           BIT 5 MX AT END I   |
C           BIT 4 MY AT END I   |---> TRUE=RELEASE
C           BIT 3 MZ AT END I   |
C           BIT 2 MX AT END J   |
C           BIT 1 MY AT END J   |
C           BIT 0 MZ AT END J   |
C
      IF (BTEST(IBUG,7)) WRITE (6,*) 'RELT',RELT
      REL=0
      IRELC=' '
      IF (MOD(RELT,10).NE.0) THEN
          REL=IBSET(REL,0)
          IRELC(5:5)='Z'
      ENDIF
      RELT=RELT/10
      IF (MOD(RELT,10).NE.0) THEN
          REL=IBSET(REL,1)
          IRELC(5:5)='Y'
      ENDIF
      RELT=RELT/10
      IF (MOD(RELT,10).NE.0) THEN
          REL=IBSET(REL,2)
          IRELC(4:4)='X'
      ENDIF
      RELT=RELT/10
      IF (MOD(RELT,10).NE.0) THEN
          REL=IBSET(REL,3)
          IRELC(3:3)='Z'
      ENDIF
      RELT=RELT/10
      IF (MOD(RELT,10).NE.0) THEN
          REL=IBSET(REL,4)
          IRELC(2:2)='Y'
      ENDIF
      RELT=RELT/10
      IF (MOD(RELT,10).NE.0) THEN
          REL=IBSET(REL,5)
          IRELC(1:1)='X'
      ENDIF
C--------- SET STIFFNESS COIEFFICIENTS...
C--------- SET CONSISTENT MASS GEOMETRIC STIFFNESS COIEFFICIENTS...
      AZ=0
      BZ=0
      CZ=0
      DZ=0
      EZ=0
      FZ=1.0 / LENGTH
C.......... BOTH ENDS RELEASED, Z-AXIS .....................
      IF (BTEST(REL,0) .AND. BTEST(REL,3)) THEN
        KIIZ=0
        KJJZ=0
        KIJZ=0
C.......... START END RELEASED, Z-AXIS .....................
      ELSE IF (BTEST(REL,3)) THEN
        KIIZ=0
        KJJZ=3
        KIJZ=0
        BZ=  0.20 * LENGTH
        EZ=  0.20
        FZ=  1.20 / LENGTH
C.......... END END RELEASED, Z-AXIS .....................
      ELSE IF (BTEST(REL,0)) THEN
        KIIZ=3
        KJJZ=0
        KIJZ=0
        AZ=  0.20 * LENGTH
        DZ=  0.20
        FZ=  1.20 / LENGTH
C.......... BOTH ENDS RELEASED, Z-AXIS .....................
      ELSE
        KIIZ=4
        KJJZ=4
        KIJZ=2
        AZ=  2 * LENGTH / 15.
        BZ=  AZ
        CZ=  LENGTH / 30.
        DZ=  0.10
        EZ=  0.10
        FZ=  1.2 / LENGTH
      ENDIF
C
      AY=0
      BY=0
      CY=0
      DY=0
      EY=0
      FY=1.0 / LENGTH
C.......... BOTH ENDS RELEASED, Y-AXIS .....................
      IF (BTEST(REL,1) .AND. BTEST(REL,4)) THEN
        KIIY=0
        KJJY=0
        KIJY=0
C.......... START END RELEASED, Y-AXIS .....................
      ELSE IF (BTEST(REL,4)) THEN
        KIIY=0
        KJJY=3
        KIJY=0
        BY=  0.20 * LENGTH
        EY=  0.20
        FY=  1.20 / LENGTH
C.......... END END RELEASED, Y-AXIS .....................
      ELSE IF (BTEST(REL,1)) THEN
        KIIY=3
        KJJY=0
        KIJY=0
        AY=  0.20 * LENGTH
        DY=  0.20
        FY=  1.20 / LENGTH
C.......... BOTH ENDS RELEASED, Z-AXIS .....................
      ELSE
        KIIY=4
        KJJY=4
        KIJY=2
        AY=  2 * LENGTH / 15.
        BY=  AY
        CY=  LENGTH / 30.
        DY=  0.10
        EY=  0.10
        FY= 1.2 / LENGTH
      ENDIF
C
      IF ( BTEST(IBUG,7) ) WRITE (6,*) 'IRELC',IRELC,'REL',REL
C
C-----------------------------------------------------------------
C  ASSEMBLE LOCAL STIFFNESS MATRIX
C-----------------------------------------------------------------
      H=EAX/LENGTH
      Q=GIX/LENGTH
      AI=KIIZ*(EIZ/LENGTH)
      AJ=KJJZ*(EIZ/LENGTH)
      C =KIJZ*(EIZ/LENGTH)
      BI=KIIY*(EIY/LENGTH)
      BJ=KJJY*(EIY/LENGTH)
      D =KIJY*(EIY/LENGTH)
      S22 = (AI+2*C+AJ)/(LENGTH**2)
      S26 = (AI+ C   )/(LENGTH   )
      S212= (      C+AJ)/(LENGTH   )
      S33 = (BI+2*D+BJ)/(LENGTH**2)
      S35 = (BI+ D   )/(LENGTH   )
      S311= (      D+BJ)/(LENGTH   )
      S( 1, 1) = H
      S( 1, 7) =-H
      S( 2, 2) = S22
      S( 2, 6) = S26
      S( 2, 8) =-S22
      S( 2,12) = S212
      S( 3, 3) = S33
      S( 3, 5) =-S35
      S( 3, 9) =-S33
      S( 3,11) =-S311
      S( 4, 4) = Q * KTI
      S( 4,10) =-Q
      S( 5, 5) = BI
      S( 5, 9) = S35
      S( 5,11) = D
      S( 6, 6) = AI
      S( 6, 8) =-S26
      S( 6,12) = C
      S( 7, 7) = H
      S( 8, 8) = S22
      S( 8,12) =-S212
      S( 9, 9) = S33
      S( 9,11) = S311
      S(10,10) = Q * KTJ
      S(11,11) = BJ
      S(12,12) = AJ
      S( 5, 3) = S( 3, 5)
      S( 6, 2) = S( 2, 6)
      S( 7, 1) = S( 1, 7)
      S( 8, 2) = S( 2, 8)
      S( 8, 6) = S( 6, 8)
      S( 9, 3) = S( 3, 9)
      S( 9, 5) = S( 5, 9)
      S(10, 4) = S( 4,10)
      S(11, 3) = S( 3,11)
      S(11, 5) = S( 5,11)
      S(11, 9) = S( 9,11)
      S(12, 2) = S( 2,12)
      S(12, 6) = S( 6,12)
      S(12, 8) = S( 8,12)
C     IF (FIRST) CALL WMATRX( S    ,12,12,' S    ')
C
C----- CALCULATE SAT
      DO 60 I=1,12
        DO 60 J=1,12
        SAT(I,J)=0
          DO 60 K=1,12
   60       SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)
C
C----- CALCULATE ASAT
      DO 70 I=1,12
        DO 70 J=1,12
        ASAT(I,J)=0
          DO 70 K=1,12
   70       ASAT(I,J)=ASAT(I,J)+A(I,K)*SAT(K,J)
      IF ( BTEST(IBUG,8) ) CALL WMATRX(ASAT  ,12,12,' ASAT ')
C
C----- DEGREES OF FREEDOM
C      REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF...
      DO 75 I=1,6
        IF ( IDOF(JOINTI,I).NE.IDOF(JOINTJ,I) ) THEN
           LMT(I  )=IDOF(JOINTI,I)
           LMT(I+6)=IDOF(JOINTJ,I)
        ELSE
           LMT(I  )= 0
           LMT(I+6)= 0
        ENDIF
   75   CONTINUE
C
C----- CONVERT TO HALF STORAGE MODE  BY COLUMNS
C
C          1  3  6 10 15 21 28 36 45 55 66 78  NUMBER INDICATES
C             2  5  9 14 20 27 35 44 54 65 77  LOCATION OF DATA
C                4  8 13 19 26 34 43 53 64 76  IN ARRAY STIFF
C                   7 12 18 25 33 42 52 63 75
C                     11 17 24 32 41 51 62 74
C                        16 23 31 40 50 61 73
C                           22 30 39 49 60 72
C                              29 38 48 59 71
C                                 37 47 58 70
C                                    46 57 69
C                                       56 68
C                                          67
C
      L=0
      IELDOF=0
      DO 90 I=1,12
        IF (LMT(I).EQ.0 .OR. ASAT(I,I).LE.0) GO TO 90
        IELDOF=IELDOF+1
        LM(IELDOF)=LMT(I)
        DO 80 J=I,1,-1
          IF (LMT(J).EQ.0 .OR. ASAT(J,J).LE.0 ) GO TO 80
          L=L+1
          STIFF(L)=ASAT(I,J)
   80     CONTINUE
   90   CONTINUE
      IF ( BTEST(IBUG,7) ) THEN
        WRITE (6,*) 'LMT=',LMT
        WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)
        CALL LMATRX(STIFF,MD,IELDOF,(L+1),'ELEMENT STIFFNESS')
        WRITE (6,*) 'KGDATA:',KGDATA,' PKG:',PKG
      ENDIF
C-----------------------------------------------------------------
C  ASSEMBLE GEOMETRIC STIFFNESS MATRIX FOR A UNIT LOAD...
```

```
C-------------------------------------------------------------------
      LKG=L
      IF ( PKG.EQ.0 .AND. KGDATA(1).NE.2 ) THEN
         AXIAL=.FALSE.
         REL=IBCLR(REL,6)
         GO TO 180
      ENDIF
C
C------------ DETERMINE IF AXIAL DEFORMATION IS CONSIDERED
C             BIT #6 OF REL IS TRUE IF AXIAL DEFORMATION IS CONSIDERED
      DO 999 I=1,3
         CS=CTS(I,1)
         CE=CTE(I,1)
         IF (ABS(CS).LE. .0001 .AND. ABS(CE).LE. .0001 ) GO TO 999
         IS=IDOF(JOINTI,I)
         IE=IDOF(JOINTJ,I)
         IF (IS.EQ.IE) GO TO 999
           REL=IBSET(REL,6)
  999    CONTINUE
C
         AXIAL=BTEST(REL,6)
      IF ( BTEST(IBUG,7) ) WRITE (6,*) 'AXIAL=',AXIAL
C
      IF (KGDATA(1).NE.0 .AND. KGDATA(2).NE.0 .AND. BTEST(REL,6)) THEN
C
C----------- DETERMINE THE AXIAL LOAD TO BE USED FOR KG...
         IF (KGDATA(1).EQ.1) THEN
              PGEOM = PKG
         ELSE IF (KGDATA(1).EQ.2) THEN
              PGEOM =  0.50*(F(1)-F(7))
         ENDIF
C
C--ASSEMBLE LOCAL GEOMETRIC STIFFNESS MATRIX FOR A UNIT LOAD...
C          ZERO LOCAL KG MATRIX
           DO 998 I=1,12
           DO 998 J=1,12
  998        S(I,J)=0
C
C---------- LUMPED MASS FORM OF GEOMETRIC STIFFNESS
         IF (KGDATA(2).EQ.1) THEN
              F1=1/LENGTH
              S( 2, 2)=  F1
              S( 2, 8)= -F1
              S( 8, 2)= -F1
              S( 8, 8)=  F1
              S( 3, 3)=  F1
              S( 3, 9)= -F1
              S( 9, 3)= -F1
              S( 9, 9)=  F1
C
C---------- CONSISTENT MASS FORM OF GEOMETRIC STIFFNESS
         ELSE IF (KGDATA(2).EQ.2) THEN
              S( 2, 2)=  FZ
              S( 2, 6)=  DZ
              S( 2, 8)= -FZ
              S( 2,12)=  EZ
               S( 6, 2)=  DZ
               S( 6, 6)=  AZ
               S( 6, 8)= -DZ
               S( 6,12)=  CZ
              S( 8, 2)= -FZ
              S( 8, 6)= -DZ
              S( 8, 8)=  FZ
              S( 8,12)= -EZ
               S(12, 2)=  EZ
               S(12, 6)= -CZ
               S(12, 8)= -EZ
               S(12,12)=  BZ
C
              S( 3, 3)=  FY
              S( 3, 5)= -DY
              S( 3, 9)= -FY
              S( 3,11)= -EY
               S( 5, 3)= -DY
               S( 5, 5)=  AY
               S( 5, 9)=  DY
               S( 5,11)= -CY
              S( 9, 3)= -FY
              S( 9, 5)=  DY
              S( 9, 9)=  FY
              S( 9,11)=  EY
               S(11, 3)= -EY
               S(11, 5)= -CY
               S(11, 9)=  EY
               S(11,11)=  BY
         ENDIF
      IF (BTEST(IBUG,8)) CALL WMATRX( S     ,12,12,' KG -LOCAL COORD ')
C
C----- CALCULATE SAT
           DO 1060 I=1,12
             DO 1060 J=1,12
             SAT(I,J)=0
               DO 1060 K=1,12
 1060            SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)
C
C----- CALCULATE ASAT
           DO 1070 I=1,12
             DO 1070 J=1,12
             ASAT(I,J)=0
               DO 1070 K=1,12
 1070            ASAT(I,J)=ASAT(I,J)+A(I,K)*SAT(K,J)
           IF (BTEST(IBUG,8))  CALL WMATRX(ASAT   ,12,12,' A*KG*AT ')
C
C----- CONVERT TO HALF STORAGE MODE  BY COLUMNS
           KGDOF=0
           DO 1090 I=1,12
             IF (LMT(I).EQ.0 .OR. ASAT(I,I).LE.0) GO TO 1090
             KGDOF=KGDOF+1
             LMKG(KGDOF)=LMT(I)
             DO 1080 J=I,1,-1
               IF (LMT(J).EQ.0 .OR. ASAT(J,J).LE.0 ) GO TO 1080
               L=L-1
               STIFF(L)=ASAT(I,J)
 1080          CONTINUE
 1090          CONTINUE
      IF ( BTEST(IBUG,7) ) THEN
             WRITE (6,*) 'KGDOF',KGDOF,'LMKG =',(LMKG(I),I=1,KGDOF)
             CALL LMATRX(STIFF(LKG+1),MD,KGDOF,(L-LKG),
     &       'ELEMENT GEOMETRIC STIFFNESS (UNIT LOAD)')
         ENDIF
      ENDIF
C
C----- RELEASE UNUSED STORAGE
  180 IREL=156-L
C
C     PRINT COLUMN DATA
      IF (FIRST .OR. BTEST(IBUG,7) .OR. BTEST(IBUG,8) ) WRITE (6,192)
  191 WRITE(6,193) NAME,IELNO,MAT,NODEI,NODEJ,IRELC,LENGTH,
     &    VY(1),VY(2),VY(3),XS,-XE,PKG
C
  192 FORMAT('/' ELEMENT 00,  3D BEAM ELEMENT'//
     &  T22,'#  MATL START   END',' REL CD',3X,'LENGTH',3X,
     &  11('-'),'  Y-AXIS  ',
     &  12('-'),'  START DIST     END DIST      PKG')
  193 FORMAT(1X,A15,4I6,2X,A,1P,G12.4,0P,
     &       F9.5,' I',3P,F9.5,' J',F9.5,' K',1P,5S,3G12.4)
      RETURN
C------------------------- DETERMINE GEOMETRIC STIFF ------------------
  200 CONTINUE
C- KG IS DETERMINED FOR A UNIT LOAD WITH IOPT=1,
C- THE ACTUAL LOAD (PGEOM) IS DETERMINED HERE.
C- KG IS MULTIPLIED BY PGEOM WHEN THE TOTAL STIFFNESS IS ASSEMBLED
C- NEGATIVE PGEOM IS COMPRESSION....
      AXIAL=BTEST(REL,6)
C
C---------- DETERMINE THE AXIAL LOAD TO BE USED FOR KG...
      IF (AXIAL) THEN
         IF (KGDATA(1).EQ.1) THEN
              PGEOM =  PKG
         ELSE IF (KGDATA(1).EQ.2) THEN
              PGEOM =  0.50*(F(1)-F(7))
         ENDIF
      ENDIF
      IF (BTEST(IBUG,7)) WRITE (6,*) 'AXIAL ',AXIAL,'  PGEOM=',PGEOM
C
      RETURN
C------------------------- DETERMINE STIFFNESS -------------------------
  300 CONTINUE
C
C     FORCE-FRAME DISPLACEMENT MATRIX
C     THE STIFFNESS IS STORED IN STIFF, THE MAPING MATRIX IN LM
C     THESE HAVE THE GLOBAL ADDRESSES IZKE AND IZLM...
C
      RETURN
C------------------------- DETERMINE INCRIMENTAL FORCES ----------------
  400 CONTINUE
       IF (PRINT.AND.FIRST) WRITE (6,491) STEPID
C
C------- CALL BMLOA TO DETERMINE THE FIXED END FORCES
      IF (MAXELD.GT.0) THEN
         CALL BMLOA(IELNO,FEMFLG,LENGTH,REL,MAXELD,NELD,NLOAD,
     &    FEM,IELD,ELD)
       ELSE
         FEMFLG=.FALSE.
       ENDIF
C
C------ LOOP FOR EACH LOAD
      DO 430 LOAD=1,NLOAD
C
C------ TRANSFORM FRAME DISPL INTO LOCAL DISPL
        DO 410 I=1,3
         R(I  )=0.
         R(I+3)=0.
         R(I+6)=0.
         R(I+9)=0.
         DO 410 J=1,3
            D1=DISPL(IDOF(JOINTI,J  ),LOAD)
            D2=DISPL(IDOF(JOINTI,J+3),LOAD)
            D3=DISPL(IDOF(JOINTJ,J  ),LOAD)
            D4=DISPL(IDOF(JOINTJ,J+3),LOAD)
          R(I  )=R(I  )+CTS(J,I)*D1 + BS(J,I)*D2
          R(I+3)=R(I+3)+CTS(J,I)*D2
          R(I+6)=R(I+6)+CTE(J,I)*D3 + BE(J,I)*D4
  410     R(I+9)=R(I+9)+CTE(J,I)*D4
C
C------ INCRIMENTAL FORCES
        F( 1)=   H*(R( 1)-R( 7))
        F( 2)= S22*(R( 2)-R( 8))+S26*R( 6)+S212*R(12)
        F( 3)= S33*(R( 3)-R( 9))-S35*R( 5)-S311*R(11)
        F( 4)=   Q*(R( 4)-R(10))
        F( 5)=-S35*(R( 3)-R( 9))+BI*R( 5)+D*R(11)
        F( 6)= S26*(R( 2)-R( 8))+AI*R( 6)+C*R(12)
C
        F( 7)=  -H*(R( 1)-R( 7))
        F( 8)=-S22*(R( 2)-R( 8))-S26*R( 6)-S212*R(12)
        F( 9)=-S33*(R( 3)-R( 9))+S35*R( 5)+S311*R(11)
        F(10)=  -Q*(R( 4)-R(10))
        F(11)=-S311*(R( 3)-R( 9))+ D*R( 5)+BJ*R(11)
        F(12)= S212*(R( 2)-R( 8))+ C*R( 6)+AJ*R(12)
C
C------ SUBTRACT OF FIXED END FORCES
        IF (FEMFLG) THEN
          DO 420 I=1,12
  420       F(I)=F(I)-FEM(I,LOAD)
        ENDIF
C
C------ DETERMINE MAXIMUN VALUES FOR MULTIPLE LOAD CASES
        IF (NLOAD.GT.1) THEN
          DO 425 I=1,12
            IF ( ABS(F(I)).GT.ABS(FMAX(I*12-12+I)) ) THEN
              DO 426 J=0,11
  426           FMAX(I*12-11+J)=F(J+1)
            ENDIF
  425     CONTINUE
        ENDIF
C
C------ PRINT FORCES
        IF (BTEST(IBUG,0).AND.PRINT)
     &                  WRITE (6,493) IELNO,LOAD,NODEI,(R(J),J=1,6),
     &                                           NODEJ,(R(J),J=7,12)
         IF (        PRINT) WRITE (6,492) IELNO,LOAD,NODEI,(F(J),J=1,6),
     &                                           NODEJ,(F(J),J=7,12)
C
  430 CONTINUE
C
C---THIS IS FOR SOL05A USE ONLY
      IF(ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN
        DO 431 I=1,12
  431     RSAF(I)=F(I)
      ENDIF
C
C
C------ CALCULATE MEMBER FORCE ON JOINT, AND ADD TO UNBALANCED FORCES
C     THIS IS USED TO CALC UNBALANCED FORCES.
        DO 440 I=1,3
         I1=IDOF(JOINTI,I  )
         I2=IDOF(JOINTI,I+3)
         J1=IDOF(JOINTJ,I  )
         J2=IDOF(JOINTJ,I+3)
         DO 440 J=1,3
           FUB(I1)=FUB(I1)
           FUB(I2)=FUB(I2)
           FUB(I2)=FUB(I2)
           FUB(J1)=FUB(J1)
           FUB(J2)=FUB(J2)
  440      FUB(J2)=FUB(J2)
C         DO 440 J=1,3
C           FUB(I1)=FUB(I1)+CTS(I,J)*(F(J  )+FT(J  ))
C           FUB(I2)=FUB(I2)+ BS(I,J)*(F(J  )+FT(J  ))
C           FUB(I2)=FUB(I2)+CTS(I,J)*(F(J+3)+FT(J+3))
C           FUB(J1)=FUB(J1)+CTE(I,J)*(F(J+6)+FT(J+6))
C           FUB(J2)=FUB(J2)+ BE(I,J)*(F(J+6)+FT(J+6))
C 440     FUB(J2)=FUB(J2)+CTE(I,J)*(F(J+9)+FT(J+9))
C
  491 FORMAT ('/'       3D BEAM FORCES...',5X/7X,A/,
     &  '   ELEMENT LOAD         NODE',7X,'AXIAL',11X,'FY',13X,'FZ',
     &  11X,'TORSION',10X,'MY',13X,'MZ')
  492 FORMAT (I10,I5,2X,'FORCE',I6,1P,6G15.6,
     &       /15X    ,2X,'FORCE',I6,   6G15.6)
  493 FORMAT (I10,I5,2X,'DISPL',I6,1P,6G15.6,
     &       /15X    ,2X,'DISPL',I6,   6G15.6)
  494 FORMAT (7X,A)
      RETURN
C------------------------- DETERMINE TOTAL FORCES ----------------------
  500 CONTINUE
       IF (PRINT.AND.FIRST) WRITE (6,491) STEPID
       IF(ICOMN .NE. 1 .AND. ICOMN .NE. 2 ) THEN
C
C------ TOTAL FORCES AND DISPLACEMENTS
        DO 510 I=1,12
C           IF (BTEST(IBUG,7)) WRITE(6,520) I,R(I),RT(I),F(I),FT(I)
            FT(I)=FT(I)+F(I)
            RT(I)=RT(I)+R(I)
  510 CONTINUE
```

```
C                                                                       ELE08100
      DO 512 I=1,12                                                     ELE08110
         IF ( ABS(FT(I)).GT.ABS(FMAX(I*12-12+I)) ) THEN                 ELE08120
            DO 511 J=0,11                                               ELE08130
  511          FMAX(I*12-11+J)=FT(J+1)                                  ELE08140
         ENDIF                                                          ELE08150
  512 CONTINUE                                                          ELE08160
      ELSE IF(ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                     ELE08170
        DO 513 I=1,12                                                   ELE08180
  513   FT(I)=RSAF(I)                                                   ELE08190
      ENDIF                                                             ELE08200
C                                                                       ELE08210
  520 FORMAT ('   I=',I3,' R=',1P,G12.4,' RT=',G12.4,                  ELE08220
     &                ' F=',1P,G12.4,' FT=',G12.4)                     ELE08230
      LOAD=1                                                            ELE08240
      IF (BTEST(IBUG,0).AND.PRINT)                                      ELE08250
     &                    WRITE (6,493) IELNO,LOAD,NODEI,(RT(J),J=1,6), ELE08260
     &                                           NODEJ,(RT(J),J=7,12)   ELE08270
      IF (         PRINT) WRITE (6,492) IELNO,LOAD,NODEI,(FT(J),J=1,6), ELE08280
     &                                           NODEJ,(FT(J),J=7,12)   ELE08290
C                                                                       ELE08300
      RETURN                                                            ELE08310
C                                                                       ELE08320
C*********************** DETERMINE FIXED END FORCES, AND ADD TO FORCE**ELE08330
  600 IF (MAXELD.LE.0) RETURN                                           ELE08340
C                                                                       ELE08350
C------- CALL BMLOA TO DETERMINE THE FIXED END FORCES                   ELE08360
       CALL BMLOA(IELNO,FEMFLG,LENGTH,REL,MAXELD,NELD,NLOAD,            ELE08370
     &      FEM,IELD,ELD)                                               ELE08380
      FEMFLG=FEMFLG                                                     ELE08390
      IF (.NOT. FEMFLG) RETURN                                          ELE08400
C                                                                       ELE08410
C------- ADD FIXED END FORCES TO LOAD MATRIX                            ELE08420
      DO 620 LOAD=1,NLOAD                                               ELE08430
        DO 610 I=1,3                                                    ELE08440
        GFEM(I  )=0.                                                    ELE08450
        GFEM(I+3)=0.                                                    ELE08460
        GFEM(I+6)=0.                                                    ELE08470
        GFEM(I+9)=0.                                                    ELE08480
        DO 610 J=1,3                                                    ELE08490
           D1=FEM(J  ,LOAD)                                             ELE08500
           D2=FEM(J+3,LOAD)                                             ELE08510
           D3=FEM(J+6,LOAD)                                             ELE08520
           D4=FEM(J+9,LOAD)                                             ELE08530
        GFEM(I  )=GFEM(I  )+CTS(I,J)*D1                                 ELE08540
        GFEM(I+3)=GFEM(I+3)+ BS(I,J)*D1 + CTS(I,J)*D2                   ELE08550
        GFEM(I+6)=GFEM(I+6)+CTE(I,J)*D3                                 ELE08560
  610   GFEM(I+9)=GFEM(I+9)+ BE(I,J)*D3 + CTE(I,J)*D4                   ELE08570
      IF (BTEST(IBUG,7)) THEN                                           ELE08580
        WRITE(6,*) 'IELNO',IELNO                                        ELE08590
        CALL WMATRX( FEM  ,1  ,12,'LOCAL  FIXED END FORCES')            ELE08600
        CALL WMATRX(GFEM  ,1 ,12,'GLOBAL FIXED END FORCES')             ELE08610
      ENDIF                                                             ELE08620
C                                                                       ELE08630
      DO 620 J=1,6                                                      ELE08640
        JI=IDOF(JOINTI,J)                                               ELE08650
        JJ=IDOF(JOINTJ,J)                                               ELE08660
        FORCE(JI,LOAD)=FORCE(JI,LOAD)+GFEM(J  )                         ELE08670
  620   FORCE(JJ,LOAD)=FORCE(JJ,LOAD)+GFEM(J+6)                         ELE08680
      IF (BTEST(IBUG,7) ) THEN                                          ELE08690
        CALL WMATRX(FORCE ,NDOF,NLOAD,'GLOBAL LOAD MATRIX')             ELE08700
      ENDIF                                                             ELE08710
C                                                                       ELE08720
      RETURN                                                            ELE08730
C                                                                       ELE08740
C********************** WRITE MEMBER FORCES TO OUTPUT FILE **************ELE08750
  700 CONTINUE                                                          ELE08760
C                                                                       ELE08770
      WRITE (IUNIT,710)  NODEI,(FT(J),J=1,6),NODEJ,(FT(J),J=7,12)      ELE08780
      WRITE (IUNIT,710)  NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12)      ELE08790
  710 FORMAT (I6,1P,6G12.5)                                             ELE08800
C                                                                       ELE08810
      RETURN                                                            ELE08820
C                                                                       ELE08830
C********************** READ AND PRINT MEMBER FORCES FROM OUTPUT FILE ***ELE08840
  800 CONTINUE                                                          ELE08850
      BACKSPACE(IUNIT)                                                  ELE08860
      READ (IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT                        ELE08870
      WRITE (6,805)                                                     ELE08880
      ISTEP=-IWRITE                                                     ELE08890
  810 READ (IUNIT,815,END=830)                                          ELE08900
     &   NODEI,(FT(J),J=1,6),NODEJ,(FT(J),J=7,12)                      ELE08910
      READ (IUNIT,815,END=830)                                          ELE08920
     &   NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12)                      ELE08930
  815 FORMAT (I6,6G12.5)                                                ELE08940
        ISTEP=ISTEP+IWRITE                                              ELE08950
        T=TO+DT*ISTEP                                                   ELE08960
CCC     MODU=MOD(ISTEP,INC)                                             ELE08970
        IF (MOD(ISTEP,INC).EQ.0) THEN                                   ELE08980
          WRITE (6,820) ISTEP,T,NODEI,(FT(J),J=1,6),                    ELE08990
     &                           NODEJ,(FT(J),J=7,12)                   ELE09000
          WRITE (6,920)                (RT(J),J=1,12)                   ELE09010
        ENDIF                                                           ELE09020
        GO TO 810                                                       ELE09030
C                                                                       ELE09040
  805 FORMAT (//'       3D BEAM FORCES...',/                            ELE09050
     &  '      STEP      TIME          NODE',7X,'AXIAL',11X,'FT',13X,'FZ',ELE09060
     &  11X,'TORSION',10X,'MT',13X,'MZ')                                ELE09070
  820 FORMAT (/,I10,1P,G15.5,0P,I6,1P,6G15.6,                           ELE09080
     &        /25X,          I6,   6G15.6)                             ELE09090
  920 FORMAT ( 25X,'DISP  ',     1P,6G15.6,                             ELE09100
     &        /31X,                6G15.6)                             ELE09110
C                                                                       ELE09120
  830 RETURN                                                            ELE09130
C                                                                       ELE09140
C********************** DETERMINE THE LENGTH OF THE MATERIAL DATA       ELE09150
  900 CONTINUE                                                          ELE09160
      NWTST=1                                                           ELE09170
      ISE=0                                                             ELE09180
C                                                                       ELE09190
        MAT   = RINPUT( 1)                                              ELE09200
        CALL MATLIB                                                     ELE09210
     &  ( 0    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,     ELE09220
     &    FT   ,F   ,RT   ,R    ,V    ,VL   ,LELEM ,LMAT  ,            ELE09230
     &    MAT  ,MATTYP,SE,IELNO,DAMAGE)                                 ELE09240
        ISE=ISE+LELEM                                                   ELE09250
      RETURN                                                            ELE09260
C                                                                       ELE09270
C********************** DETERMINE THE TOTAL STRAIN ENERGY               ELE09280
 1000 CONTINUE                                                          ELE09290
      EESE=0                                                            ELE09300
      EPSE=0                                                            ELE09310
C                                                                       ELE09320
      DO 1010 I=1,12                                                    ELE09330
 1010   EESE=EESE + (FT(I)+F(I))*(R(I)+RT(I))/2.                       ELE09340
C                                                                       ELE09350
      ENGDAT(1) = EESE                                                  ELE09360
      ENGDAT(2) = 0                                                     ELE09370
C                                                                       ELE09380
      RETURN                                                            ELE09390
C                                                                       ELE09400
C********************** DUCTILITIES AND EXCURSION RATIOS                ELE09410
C                       NO D & E RATIOS FOR ELASTIC ELEMENT             ELE09420
 1100 RETURN                                                            ELE09430
C                                                                       ELE09440
C                                                                       ELE09450
C********************** PRINT MAXIMUM ELEMENT LOADS                     ELE09460
C                                                                       ELE09470
 1200 CONTINUE                                                          ELE09480
      LOAD2=0                                                           ELE09490
      IF (FIRST) WRITE(6,494)' COLUMN ( LOAD) REPRESENTS THE DOF WHICH' ELE09500
     &          //' HAS MAXIMUM VALUE'                                 ELE09510
      IF (FIRST) WRITE (6,491) ' MAXIMUM VALUES FOR ALL STEPS '         ELE09520
     &          //' NOTE: MAXIMUM VALUES WITH THE OTHER DOFS FORCES'    ELE09530
     &          //' ARE PRINT OUT AT THE SAME TIME'                     ELE09540
        DO 309 I=0,11                                                   ELE09550
          LOAD2=I+1                                                     ELE09560
          IF( FMAX(I*12+1+I) .EQ. 0. ) GO TO 309                        ELE09570
          WRITE (6,492) IELNO,LOAD2,NODEI,(FMAX(I*12+J),J=1,6),         ELE09580
     &                              NODEJ,(FMAX(I*12+J),J=7,12)         ELE09590
  309 CONTINUE                                                          ELE09600
C                                                                       ELE09610
      RETURN                                                            ELE09620
C                                                                       ELE09630
C                                                                       ELE09640
C******************** RESET ELEMENT FORCES                             ELE09650
C                                                                       ELE09660
 1300 CONTINUE                                                          ELE09670
C----- ZERO MATRICIES                                                  ELE09680
        DO 1340 I=1,12                                                  ELE09690
          FMAX(I*12-11)=0                                               ELE09700
          F (I)=0                                                       ELE09710
          FT(I)=0                                                       ELE09720
          R (I)=0                                                       ELE09730
 1340     RT(I)=0                                                       ELE09740
      WRITE (6,1330) IELNO                                              ELE09750
 1330 FORMAT (' 3D BEAM........ ELEMENT #',I6,                          ELE09760
     &   ' INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO')    ELE09770
        FALSE=.FALSE.                                                   ELE09780
        LSTTYP=0                                                        ELE09790
        CALL MATLIB                                                     ELE09800
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,     ELE09810
     &    FT   ,F   ,RT   ,R    ,V    ,VL   ,LELEM ,LMAT  ,            ELE09820
     &    MAT  ,MATTYP,SE,IELNO,DAMAGE)                                 ELE09830
C                                                                       ELE09840
      RETURN                                                            ELE09850
C                                                                       ELE09860
C******************** DAMAGE INDEX                                     ELE09870
C                                                                       ELE09880
 1400 DAMAGE = 0.                                                       ELE09890
C                                                                       ELE09900
      RETURN                                                            ELE09910
C                                                                       ELE09920
      END                                                               ELE09930
C*************************************************************************ELE00010
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    ELE00020
C           END DEBUG                                                   ELE00030
C*************************************************************************ELE00040
      SUBROUTINE ELE02                                                  ELE00050
     &  (IOPT       ,IUNIT      ,NEWK       ,FIRST      ,PRINT      ,   ELE00060
     &   NDOF       ,NLOAD      ,MAXNOD     ,NNODE      ,NCOS       ,   ELE00070
     &   ID         ,IDOF       ,COORD      ,COSINE     ,JTCOS      ,   ELE00080
     &   CONST      ,DISPL      ,VELOC      ,FORCE      ,FUB        ,   ELE00090
     &   IREL       ,BUG        ,IBUG       ,ACCEL      ,KGDATA     ,   ELE00100
     &   GRAV       ,PGEOM      ,ELSTIC     ,NAME       ,STEPID     ,   ELE00110
     &   IELNO      ,RINPUT     ,EESE       ,EPSE       ,INC        ,   ELE00120
     &   DAMAGE     ,DUCFAG     ,                                       ELE00130
     &   IELTYP     ,IELDOF     ,NODEI      ,NODEJ      ,JOINTI     ,   ELE00140
     &   JOINTJ     ,R          ,RT         ,F          ,FT         ,   ELE00150
     &   V          ,VT         ,ISE        ,H          ,LENGTH     ,   ELE00160
     &   MAT        ,A          ,LM         ,KTYPE      ,FMAX       ,   ELE00170
     &   SE         ,STIFF      )                                       ELE00180
C                                                                       ELE00190
      IMPLICIT REAL(A-H,O-Z)                                            ELE00200
C                                                                       ELE00210
      COMMON /RSA/ICOMN, RSAF(12)                                       ELE00220
      REAL ICOMN                                                        ELE00230
C                                                                       ELE00240
      LOGICAL ERR,FIRST,PRINT,BUG,FALSE,NEWK,ELSTIC                     ELE00250
      LOGICAL BTEST                                                     ELE00260
      CHARACTER*80 NAME                                                 ELE00270
      CHARACTER*(*) STEPID                                              ELE00280
      CHARACTER*9  TYPE(6)                                              ELE00290
      DIMENSION  IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS)       ELE00300
      DIMENSION CONST(MAXNOD,6),   ID(MAXNOD)  ,JTCOS(MAXNOD)          ELE00310
      DIMENSION SAT(2,12),STIFF(78)                                     ELE00320
      DIMENSION A(12,2),S(2,2),LM(12),FMAX(2)                           ELE00330
      DIMENSION CTS( 3, 3),CTE( 3, 3),VX(3),VY(3),VZ(3)                 ELE00340
      DIMENSION BS( 3, 3),BE( 3, 3),SE(ISE)                             ELE00350
      DIMENSION DISPL(NDOF,NLOAD),FUB(NDOF)                             ELE00360
      DIMENSION VELOC(NDOF)                                             ELE00370
      DIMENSION FORCE(NDOF,NLOAD)                                       ELE00380
      DIMENSION RINPUT(100),WORK(6),DUCT(3),EXCR(6)                     ELE00390
      REAL LENGTH                                                       ELE00400
      REAL IX,IY,IZ,KIIZ,KJJZ,KIJZ,KIIY,KJJY,KIJY,KTI,TYJ              ELE00410
      INTEGER REL,RELT                                                  ELE00420
      DIMENSION MD(13)                                                  ELE00430
      DATA MD/1,2,4,7,11,16,22,29,37,46,56,67,79/                       ELE00440
      DATA TYPE/'    AXIAL',' SHEAR-Y',' SHEAR-Z',                      ELE00450
     &         ' TORSION','BENDING-Y','BENDING-Z'/                      ELE00460
C                                                                       ELE00470
C*************************************************************************ELE00480
C VARIABLES:                                                            ELE00490
C-------------------------------------------------------------------------ELE00500
C---------- GLOBAL VARIABLES                                           ELE00510
C IOPT   = 1, INITIALIZE BEAM COLUMN ELEMENT                            ELE00520
C        = 2, CALCULATE GEOMETRIC STIFFNESS                             ELE00530
C        = 3, FORM STIFFNESS                                            ELE00540
C        = 4, CALCULATE INCREMENTAL FORCES                              ELE00550
C        = 5, CALCULATE TOTAL FORCES AND ENERGIES                       ELE00560
C IUNIT  = OUTPUT FILE UNIT # FOR PRINTING OUTPUT                       ELE00570
C ERR    = ERR OR FLAG, IF AN ERR OR HAS OCCURED, ERR=.TRUE.            ELE00580
C FIRST  = FLAG, FIRST=.TRUE., PRINT HEADERS                            ELE00590
C PRINT  = FLAG, PRINT=.TRUE., PRINT DATA                               ELE00600
C NDOF   = # OF GLOBAL DOF                                              ELE00610
C NLOAD  = # OF LOAD COMBINATIONS                                       ELE00620
C MAXNOD = # ROW DIMENSION OF NODE ARRAY                                ELE00630
C NNODE  = # OF NODES                                                   ELE00640
C ID     = ARRAY OF EXTERNAL NODE NUMBERS                               ELE00650
C IDOF   = ARRAY OF DEGREES OF FREEDOM                                  ELE00660
C COORD  = ARRAY OF COORDINATES                                         ELE00670
C COSINE = ARRAY OF DIRECTION COSINES OF NODES                          ELE00680
C CONST  = ARRAY OF CONSTRAINT TRANSFORMATIONS                          ELE00690
C DISPL  = GLOBAL DISPLACEMENT MATRIX                                   ELE00700
C Z      = REAL GLOBAL STORTAG VECTOR                                   ELE00710
C NZ     = INTEGER GLOBAL STORTAG VECTOR                                ELE00720
C IZMAT  = LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR           ELE00730
C NAME   = ELEMENT NAME                                                 ELE00740
C IELNO  = ELEMENT NUMBER                                               ELE00750
C RINPUT = INPUT DATA                                                   ELE00760
C STIFF  = OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)             ELE00770
C LM     = OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX              ELE00780
C---------- ELEMENT VARIABLES                                           ELE00790
C H,Q,AI,AJ,C,BI,BJ,S22,S12,SJJ,SJI,S11,L                               ELE00800
C        = INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX         ELE00810
C PKG    = INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX...         ELE00820
C R      = INCRIMENTAL DISPLACEMENTS                                    ELE00830
C RT     = TOTAL DISPLACEMENTS                                          ELE00840
C F      = INCRIMENTAL FORCES                                           ELE00850
C FT     = TOTAL FORCES                                                 ELE00860
C LM     = LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                     ELE00870
C CTS    = LOCAL TO GLOBAL ROTATION MATRIX START                        ELE00880
C CTE    = LOCAL TO GLOBAL ROTATION MATRIX END                          ELE00890
C BS     = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX STARTELE00900
C BE     = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END  ELE00910
C STIFF  = ELEMENT STIFFNESS                                            ELE00920
C IELTYP = ELEMENT TYPE = 10                                            ELE00930
C NODEI  = EXTERNAL JOINT NUMBER, END I                                 ELE00940
C NODEJ  = EXTERNAL JOINT NUMBER, END J                                 ELE00950
C JOINTI = INTERNAL JOINT NUMBER, END I                                 ELE00960
C JOINTJ = INTERNAL JOINT NUMBER, END J                                 ELE00970
C---------- TEMPROARY VARIABLES                                         ELE00980
C VX     = VECTOR DEFINING THE ELEMENT X AXIS                           ELE00990
C VY     = VECTOR DEFINING THE ELEMENT Y AXIS                           ELE01000
```

```
C COSLOC = LOCAL COSINE MATRIX                                          ELE01010
C A      = GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                  ELE01020
C S      = ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART   ELE01030
C SAT    = PRODUCT OF S*A'                                              ELE01040
C ASAT   = PRODUCT OF A*SAT                                             ELE01050
C-----------------------------------------------------------------------ELE01060
C                                                                       ELE01070
C                                                                       ELE01080
C------- CHOOSE OPTION                                                  ELE01090
       IF (IOPT.LT.1 .OR.IOPT.GT.14)                                    ELE01100
     & WRITE (6,*) 'INVALID OPTION IN ELE-02, IOPT=',IOPT               ELE01110
C                                                                       ELE01120
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,        ELE01130
     &                                          1300,1400),IOPT         ELE01140
C                                                                       ELE01150
  100 CONTINUE                                                          ELE01160
C                                                                       ELE01170
C------ INTERPERTATE INPUT DATA                                         ELE01180
      IELTYP= 2                                                         ELE01190
      MAT   = RINPUT( 1)                                                ELE01200
      NODEI = RINPUT( 2)                                                ELE01210
      NODEJ = RINPUT( 3)                                                ELE01220
      KTYPE = RINPUT( 4)                                                ELE01230
      XLEN  = RINPUT( 5)                                                ELE01240
      VY(1) = RINPUT( 6)                                                ELE01250
      VY(2) = RINPUT( 7)                                                ELE01260
      VY(3) = RINPUT( 8)                                                ELE01270
      XS    = RINPUT( 9)                                                ELE01280
      XE    = RINPUT(10)                                                ELE01290
C                                                                       ELE01300
C------ GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S           ELE01310
      JOINTI=IQUICK(NODEI,ID,NNODE)                                     ELE01320
      JOINTJ=IQUICK(NODEJ,ID,NNODE)                                     ELE01330
C     WRITE (6,*) 'ID=',ID                                              ELE01340
C                                                                       ELE01350
      IF (KTYPE.LT.1 .OR. KTYPE.GT.6) THEN                              ELE01360
         WRITE(6,102) IELNO,NODEI,NODEJ,KTYPE                           ELE01370
  102    FORMAT(1X,20('*'),'ER',' ROR - INVALID KTYPE',/                ELE01380
     &    5X,'IELNO=',I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,              ELE01390
     &    5X,'KTYPE=',I5,/                                              ELE01400
     &    5X,'REVISE INPUT  1 LE KTYPE LE 6 .............')            ELE01410
         LENGTH=1.0                                                     ELE01420
      ENDIF                                                             ELE01430
C                                                                       ELE01440
C------ PRINT DATA FOR PLOTTING                                         ELE01450
      IF (BTEST(IBUG,2)) THEN                                           ELE01460
        WRITE(7,10) NODEJ, NODEI,                                       ELE01470
     &              (COORD(JOINTJ,I),I=1,3),(COORD(JOINTI,J),J=1,3)     ELE01480
   10  FORMAT (I5,1X,I5,1P,6G12.4)                                      ELE01490
      ENDIF                                                             ELE01500
C                                                                       ELE01510
C-----------------------------------------------------------------------ELE01520
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                            ELE01530
C-----------------------------------------------------------------------ELE01540
C------ DEFINE VECTOR X, LENGTH                                         ELE01550
      VX(1)=COORD(JOINTJ,1)-COORD(JOINTI,1)                             ELE01560
      VX(2)=COORD(JOINTJ,2)-COORD(JOINTI,2)                             ELE01570
      VX(3)=COORD(JOINTJ,3)-COORD(JOINTI,3)                             ELE01580
      LENGTH=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)-XS-XE                     ELE01590
      IF (LENGTH.EQ.0) THEN                                             ELE01600
        DO 103 II=1,3                                                   ELE01610
          VX(II)= COSINE(1,II,JTCOS(JOINTI))                            ELE01620
  103     VY(II)= COSINE(2,II,JTCOS(JOINTI))                            ELE01630
        LENGTH=1.00                                                     ELE01640
      ELSE                                                              ELE01650
        LENGTH=LENGTH-XS-XE                                             ELE01660
      ENDIF                                                             ELE01670
      IF (XLEN.NE.0) THEN                                               ELE01680
         LENGTH=XLEN                                                    ELE01690
         XS=0                                                           ELE01700
         XE=0                                                           ELE01710
      ENDIF                                                             ELE01720
      IF (LENGTH.LE.0) THEN                                             ELE01730
         WRITE(6,101) IELNO,JOINTI,JOINTJ,LENGTH                        ELE01740
  101    FORMAT(1X,20('*'),'ER',' ROR - LENGTH IS LE 0',/               ELE01750
     &    5X,'IELNO=',I5,5X,'JOINTI=',I5.5X,'JOINTJ=',I5,              ELE01760
     &    5X,'LENGTH',1P,G15.6/                                         ELE01770
     &    5X,'REVISE INPUT.............................')              ELE01780
         LENGTH=1.0                                                     ELE01790
      ENDIF                                                             ELE01800
      IF ( BTEST(IBUG,7) ) THEN                                         ELE01810
         WRITE (6,*) 'JOINTS',JOINTI,JOINTJ                             ELE01820
         WRITE (6,*) 'VX',VX                                            ELE01830
         WRITE (6,*) 'VY',VY                                            ELE01840
         WRITE (6,*) 'LENGTH  ',LENGTH                                  ELE01850
      ENDIF                                                             ELE01860
C------ GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES CT AND B           ELE01870
      ICS=JTCOS(JOINTI)                                                 ELE01880
      ICE=JTCOS(JOINTJ)                                                 ELE01890
      CALL ROTXYZ(VX,VY,COSINE(1,1,ICS),CTS, XS ,0. ,0. ,BS,            ELE01900
     &       CONST(JOINTI,1),MAXNOD)                                    ELE01910
      CALL ROTXYZ(VX,VY,COSINE(1,1,ICE),CTE,-XE ,0. ,0. ,BE,            ELE01920
     &       CONST(JOINTJ,1),MAXNOD)                                    ELE01930
      IF ( BTEST(IBUG,7) ) THEN                                         ELE01940
         CALL WMATRX( CTS  , 3, 3,'CTS   ')                             ELE01950
         CALL WMATRX( CTE  , 3, 3,'CTE   ')                             ELE01960
      ENDIF                                                             ELE01970
C                                                                       ELE01980
C------ GET GLOBAL CONSTRAINT MATRIX BS, BE                             ELE01990
C     BS(1,1)=CONST(JOINTI,1)*CTS(2,1)+CONST(JOINTI,2)*CTS(3,1)+BS(1,1) ELE02000
C     BS(1,2)=CONST(JOINTI,1)*CTS(2,2)+CONST(JOINTI,2)*CTS(3,2)+BS(1,2) ELE02010
C     BS(1,3)=CONST(JOINTI,1)*CTS(2,3)+CONST(JOINTI,2)*CTS(3,3)+BS(1,3) ELE02020
C     BS(2,1)=CONST(JOINTI,3)*CTS(1,1)+CONST(JOINTI,4)*CTS(3,1)+BS(2,1) ELE02030
C     BS(2,2)=CONST(JOINTI,3)*CTS(1,2)+CONST(JOINTI,4)*CTS(3,2)+BS(2,2) ELE02040
C     BS(2,3)=CONST(JOINTI,3)*CTS(1,3)+CONST(JOINTI,4)*CTS(3,3)+BS(2,3) ELE02050
C     BS(3,1)=CONST(JOINTI,5)*CTS(1,1)+CONST(JOINTI,6)*CTS(2,1)+BS(3,1) ELE02060
C     BS(3,2)=CONST(JOINTI,5)*CTS(1,2)+CONST(JOINTI,6)*CTS(2,2)+BS(3,2) ELE02070
C     BS(3,3)=CONST(JOINTI,5)*CTS(1,3)+CONST(JOINTI,6)*CTS(2,3)+BS(3,3) ELE02080
C                                                                       ELE02090
C     BE(1,1)=CONST(JOINTJ,1)*CTE(2,1)+CONST(JOINTJ,2)*CTE(3,1)+BE(1,1) ELE02100
C     BE(1,2)=CONST(JOINTJ,1)*CTE(2,2)+CONST(JOINTJ,2)*CTE(3,2)+BE(1,2) ELE02110
C     BE(1,3)=CONST(JOINTJ,1)*CTE(2,3)+CONST(JOINTJ,2)*CTE(3,3)+BE(1,3) ELE02120
C     BE(2,1)=CONST(JOINTJ,3)*CTE(1,1)+CONST(JOINTJ,4)*CTE(3,1)+BE(2,1) ELE02130
C     BE(2,2)=CONST(JOINTJ,3)*CTE(1,2)+CONST(JOINTJ,4)*CTE(3,2)+BE(2,2) ELE02140
C     BE(2,3)=CONST(JOINTJ,3)*CTE(1,3)+CONST(JOINTJ,4)*CTE(3,3)+BE(2,3) ELE02150
C     BE(3,1)=CONST(JOINTJ,5)*CTE(1,1)+CONST(JOINTJ,6)*CTE(2,1)+BE(3,1) ELE02160
C     BE(3,2)=CONST(JOINTJ,5)*CTE(1,2)+CONST(JOINTJ,6)*CTE(2,2)+BE(3,2) ELE02170
C     BE(3,3)=CONST(JOINTJ,5)*CTE(1,3)+CONST(JOINTJ,6)*CTE(2,3)+BE(3,3) ELE02180
C                                                                       ELE02190
      IF ( BTEST(IBUG,7) ) THEN                                         ELE02200
         CALL WMATRX(BS, 3, 3,'BS   ')                                  ELE02210
         CALL WMATRX(BE, 3, 3,'BE   ')                                  ELE02220
      ENDIF                                                             ELE02230
C                                                                       ELE02240
C------ ZERO MATRICIES                                                  ELE02250
         F =0                                                           ELE02260
         FT=0                                                           ELE02270
         R =0                                                           ELE02280
         RT=0                                                           ELE02290
         V =0                                                           ELE02300
         VT=0                                                           ELE02310
         FMAX(1)=0                                                      ELE02320
         FMAX(2)=0                                                      ELE02330
C                                                                       ELE02340
C------ ASSEMBLE TRANSFORMATION MATRIX A                                ELE02350
C  NOTE: A IS DIMMENSIONED AS A 12X2 MATRIX, BUT ONLY THE FIRST IELDOF  ELE02360
C        ROW'S CONTAIN INFORMATION. THE BALANCE OF THE MATRIX IS        ELE02370
C        UNDEFINED.  THE GLOBAL DOF COORESPONDING TO THE ROW'S OF A     ELE02380
C        ARE STORED IN VECTOR LM.  AGAIN, ONLY THE FIRST IELDOF ROW'S   ELE02390
C        OF LM ARE DEFINED.                                             ELE02400
       IELDOF=0                                                         ELE02410
       DO 50 I=1,12                                                     ELE02420
C                                                                       ELE02430
C              REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF...     ELE02440
         J=I                                                            ELE02450
         IF (I.GT.6) J= I-6                                             ELE02460
         IF ( IDOF(JOINTI,J).EQ.IDOF(JOINTJ,J) ) GO TO 50               ELE02470
         IF (I.LE.6) THEN                                               ELE02480
            LMTI=IDOF(JOINTI,I)                                         ELE02490
         ELSE                                                           ELE02500
            LMTI=IDOF(JOINTJ,J)                                         ELE02510
         ENDIF                                                          ELE02520
C                                                                       ELE02530
         A1=0                                                           ELE02540
         A2=0                                                           ELE02550
         IF (I.LE.3) THEN                                               ELE02560
            IF (KTYPE.LE.3) A1=CTS(I,KTYPE)                             ELE02570
         ELSE IF (I.LE.6) THEN                                          ELE02580
            IF (KTYPE.LE.3) THEN                                        ELE02590
               A1=BS(I-3,KTYPE)                                         ELE02600
            ELSE                                                        ELE02610
               A1=CTS(I-3,KTYPE-3)                                      ELE02620
            ENDIF                                                       ELE02630
         ELSE IF (I.LE.9) THEN                                          ELE02640
            IF (KTYPE.LE.3) A2=CTE(I-6,KTYPE)                           ELE02650
         ELSE                                                           ELE02660
            IF (KTYPE.LE.3) THEN                                        ELE02670
               A2=BE(I-9,KTYPE)                                         ELE02680
            ELSE                                                        ELE02690
               A2=CTE(I-9,KTYPE-3)                                      ELE02700
            ENDIF                                                       ELE02710
         ENDIF                                                          ELE02720
         IF (A1.EQ.0 .AND. A2.EQ.0) GO TO 50                            ELE02730
         IELDOF=IELDOF+1                                                ELE02740
         A(IELDOF,1)=A1                                                 ELE02750
         A(IELDOF,2)=A2                                                 ELE02760
         LM(IELDOF)=LMTI                                                ELE02770
C                                                                       ELE02780
   50    CONTINUE                                                       ELE02790
      IF ( BTEST(IBUG,7) ) CALL WMTRX2( A    ,IELDOF,2,' A    ',12)   ELE02800
C                                                                       ELE02810
C----- GET SPRING STIFFNESS                                             ELE02820
         FALSE=.FALSE.                                                  ELE02830
         LSTTYP=0                                                       ELE02840
         CALL MATLIB                                                    ELE02850
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,  DUCT,EXCR,  ELE02860
     &    PT   ,P      ,MT    ,R   ,0.    ,0.   ,DELEM ,LMAT  ,        ELE02870
     &    MAT  ,MATTYP,SE,IELNO,DAMAGE)                                 ELE02880
C                                                                       ELE02890
C     WRITE (6,*) 'AFTER MA','TLIB IN ELE02 '                           ELE02900
  111 FORMAT (0P,' ZMATRX(',I6,')=',I15,1P,G20.10)                      ELE02910
C------- R IS THE SPRING STIFFNESS                                      ELE02920
C                                                                       ELE02930
         R = SE(1)                                                      ELE02940
         HH= SE(1)/LENGTH                                               ELE02950
         S(1,1)= HH                                                     ELE02960
         S(1,2)=-HH                                                     ELE02970
         S(2,1)=-HH                                                     ELE02980
         S(2,2)= HH                                                     ELE02990
C                                                                       ELE03000
C----- CALCULATE SAT                                                    ELE03010
       DO 60 I=1,2                                                      ELE03020
         DO 60 J=1,IELDOF                                               ELE03030
         SAT(I,J)=0                                                     ELE03040
           DO 60 K=1,2                                                  ELE03050
   60      SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)                              ELE03060
C                                                                       ELE03070
C----- CALCULATE ASAT AND CONVERT TO HALF STORAGE MODE  BY COLUMNS      ELE03080
C                                                                       ELE03090
C          1  3  6 10 15 21 28 36 45 55 66 78  NUMBER INDICATES        ELE03100
C             2  5  9 14 20 27 35 44 54 65 77  LOCATION OF DATA        ELE03110
C                4  8 13 19 26 34 43 53 64 76  IN ARRAY STIFF          ELE03120
C                   7 12 18 25 33 42 52 63 75                          ELE03130
C                  11 17 24 32 41 51 62 74                             ELE03140
C                     16 23 31 40 50 61 73                             ELE03150
C                        22 30 39 49 60 72                             ELE03160
C                           29 38 48 59 71                             ELE03170
C                              37 47 58 70                             ELE03180
C                                 46 57 69                             ELE03190
C                                    56 68                             ELE03200
C                                       67                             ELE03210
       L=0                                                              ELE03220
       DO 70 J=1,IELDOF                                                 ELE03230
         DO 70 I=J,1,-1                                                 ELE03240
         L=L+1                                                          ELE03250
         STIFF(L)=0                                                     ELE03260
           DO 70 K=1,2                                                  ELE03270
   70      STIFF(L)=STIFF(L)+A(I,K)*SAT(K,J)                            ELE03280
C                                                                       ELE03290
       IF ( BTEST(IBUG,7) ) THEN                                        ELE03300
         WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)          ELE03310
         CALL LMATRX(STIFF,MD,IELDOF,(L+1),'SPRING STIFFNESS')          ELE03320
       ENDIF                                                            ELE03330
C                                                                       ELE03340
C------ RELEASE UNUSED STORAGE                                          ELE03350
  180 IREL=78-L                                                         ELE03360
C                                                                       ELE03370
C     PRINT COLUMN DATA                                                 ELE03380
      IF (FIRST .OR. BTEST(IBUG,7) .OR. BTEST(IBUG,8) ) WRITE (6,192)   ELE03390
  191 WRITE(6,193) NAME,IELNO,MAT,NODEI,NODEJ,TYPE(KTYPE),LENGTH,       ELE03400
     &    VY(1),VY(2),VY(3),XS,-XE                                      ELE03410
C                                                                       ELE03420
  192 FORMAT(/' ELEMENT 02, ONE (LOCAL) DOF SPRING ELEMENT'//           ELE03430
     &  T22,'@  MATL START   END',2X,'--TYPE---',6X,'LENGTH',3X,        ELE03440
     &  11('-'),'  Y-AXIS  ',                                           ELE03450
     &  12('-'),'  START DIST    END DIST')                             ELE03460
  193 FORMAT(1X,A15,4I6,2X,A9,3X,1P,G12.4,0P,                           ELE03470
     &       F9.5,' I',SP,F9.5,' J',F9.5,' K',1P,5E,3G12.4)             ELE03480
      RETURN                                                            ELE03490
C------------------------- DETERMINE GEOMETRIC STIFF ------------------ELE03500
  200 RETURN                                                            ELE03510
C    THE GEOMETRIC STIFFNESS MATRIX IS NOT AVIALABLE FOR THIS ELEMENT   ELE03520
C                                                                       ELE03530
C------------------------- DETERMINE STIFFNESS ------------------------ELE03540
  300 CONTINUE                                                          ELE03550
C                                                                       ELE03560
C------ R CONTAINS THE SPRING STIFFNESS, WHICH WAS DETERMINED IN THE    ELE03570
C       LAST CALL TO MAT_LIB                                            ELE03580
C       IF H=SE(1) THEN THE STIFFNESS HAS NOT CHANGED, RETURN           ELE03590
C                                                                       ELE03600
      IF (H.EQ.SE(1)) RETURN                                            ELE03610
C                                                                       ELE03620
         H = SE(1)                                                      ELE03630
         HH= SE(1)/LENGTH                                               ELE03640
         S(1,1)= HH                                                     ELE03650
         S(1,2)=-HH                                                     ELE03660
         S(2,1)=-HH                                                     ELE03670
         S(2,2)= HH                                                     ELE03680
C                                                                       ELE03690
C------ CALCULATE SAT                                                   ELE03700
       DO 360 I=1,2                                                     ELE03710
         DO 360 J=1,IELDOF                                              ELE03720
         SAT(I,J)=0                                                     ELE03730
           DO 360 K=1,2                                                 ELE03740
  360      SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)                              ELE03750
C                                                                       ELE03760
C------ CALCULATE ASAT, STORE IN UPPER TRIANGULAR MATRIX                ELE03770
       L=0                                                              ELE03780
       DO 370 J=1,IELDOF                                                ELE03790
         DO 370 I=J,1,-1                                                ELE03800
         L=L+1                                                          ELE03810
         STIFF(L)=0                                                     ELE03820
           DO 370 K=1,2                                                 ELE03830
  370      STIFF(L)=STIFF(L)+A(I,K)*SAT(K,J)                            ELE03840
```

```
C                                                                       ELE03850
      IF ( BTEST(IBUG,7) ) THEN                                         ELE03860
        WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)           ELE03870
        CALL LMATRX(STIFF,MD,IELDOF,(L+1),'SPRING STIFFNESS')           ELE03880
      ENDIF                                                             ELE03890
C                                                                       ELE03900
      RETURN                                                            ELE03910
C************************* DETERMINE INCRIMENTAL FORCES ****************ELE03920
  400 CONTINUE                                                          ELE03930
       IF (PRINT.AND.FIRST) WRITE (6,491) STEPID                        ELE03940
C                                                                       ELE03950
C------ LOOP FOR EACH LOAD                                             ELE03960
      DO 430 LOAD=1,NLOAD                                               ELE03970
C                                                                       ELE03980
C------ TRANSFORM FRAME DISPL INTO LOCAL DISPL                         ELE03990
C... LOCAL DISPL ARE AT THE END END OF THE SPRING, THUS                 ELE04000
C... POSITIVE IS TENSION AND NEGATIVE IS COMPRESSION                    ELE04010
        R=0                                                             ELE04020
        V=0                                                             ELE04030
        DO 410 I=1,IELDOF                                               ELE04040
           J=LM(I)                                                      ELE04050
              AI1= A(I,1)                                               ELE04060
              AI2= A(I,2)                                               ELE04070
              DISP= DISPL(J,LOAD)                                       ELE04080
              VEL= VELOC(J)                                             ELE04090
C                                                                       ELE04100
           R= R + (A(I,2)-A(I,1)) * DISPL(J,LOAD)                       ELE04110
  410      V= V + (A(I,2)-A(I,1)) * VELOC(J)                            ELE04120
C                                                                       ELE04130
C------ INCRIMENTAL FORCES                                             ELE04140
      F= H*R/LENGTH                                                     ELE04150
C                                                                       ELE04160
C------ CHECK STIFFNESS AND CALCULATE UNBALANCED FORCES                ELE04170
C       THIS IS PREFORMED ONLY FOR INELASTIC ANALYSIS                   ELE04180
      P =FT+F                                                           ELE04190
      IF (.NOT. ELSTIC) THEN                                            ELE04200
C          -- SET UP TEMPORARY - TOTAL LOADS, DISPL'S AND VELOC         ELE04210
           PL=FT                                                        ELE04220
           D =(RT+R)/LENGTH                                             ELE04230
           DL=RT /LENGTH                                                ELE04240
           VC=VT+V                                                      ELE04250
           VL=VT                                                        ELE04260
C          -- TRANSFER PERMANENT HYSTERESIS DATA TO TEMPROARY STORAGE... ELE04270
           ISE2=ISE/2                                                   ELE04280
           DO 415 I=1,ISE2                                              ELE04290
  415         SE(I+ISE2)=SE(I)                                          ELE04300
C          -- CALL HYST MODEL TO GET NEW STIFFNESS AND                  ELE04310
C             THE TOTAL FORCE P AT DISPL D.                             ELE04320
           MOPT=3                                                       ELE04330
           CALL MATLIB                                                  ELE04340
     &     (MOPT ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE, DUCT,EXCR,  ELE04350
     &      P    ,PL    ,D     ,DL    ,VC    ,VL    ,LELEM ,LMAT  ,    ELE04360
     &      MAT  ,MATTYP,SE(ISE2+1),IELNO,DAMAGE)                       ELE04370
           IF (SE(ISE2+1).NE.SE(1)) NEWK=.TRUE.                         ELE04380
      ENDIF                                                             ELE04390
C                                                                       ELE04400
C------ SAVE PEAK VALUES FOR MULTIPLE LOAD CASES                       ELE04410
      IF (NLOAD.GT.1 .AND.  ABS(F).GT.ABS(FMAX(1)) ) THEN               ELE04420
        FMAX(1)=F                                                       ELE04430
        FMAX(2)=R                                                       ELE04440
      ENDIF                                                             ELE04450
C                                                                       ELE04460
C------ PRINT DATA                                                     ELE04470
      HH= H/LENGTH                                                      ELE04480
      IF (PRINT) WRITE (6,492) IELNO,LOAD,TYPE(KTYPE),NODEI,NODEJ,F,R,HHELE04490
C                                                                       ELE04500
  430 CONTINUE                                                          ELE04510
C                                                                       ELE04520
C------ THIS IS FOR SOL05A USED ONLY                                   ELE04530
      IF(ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                          ELE04540
        RSAF(1)=F                                                       ELE04550
      ENDIF                                                             ELE04560
C                                                                       ELE04570
C------ CALCULATE THE UNBALANCED MEMBER FORCE ON A JOINT.              ELE04580
           UBAL= F+FT-P                                                 ELE04590
           P   = P                                                      ELE04600
           FT  = FT                                                     ELE04610
           P   = P                                                      ELE04620
           IF (BTEST(IBUG,7)) WRITE (6,493) F,FT,P                      ELE04630
           DO 440 I=1,IELDOF                                            ELE04640
              J=LM(I)                                                   ELE04650
              AI1= A(I,1)                                               ELE04660
              AI2= A(I,2)                                               ELE04670
              FUB(J)= FUB(J) + (A(I,1)-A(I,2)) * UBAL                   ELE04680
              IF (BTEST(IBUG,7))                                        ELE04690
     &        WRITE (6,494) (II,LM(II),FUB(II),II=1,IELDOF)             ELE04700
  440      CONTINUE                                                     ELE04710
C                                                                       ELE04720
  491 FORMAT (//'       SPRING FORCES...',5X,A/                         ELE04730
     & '   ELEMENT LOAD',T17,'      TYPE',' NODEI',' NODEJ',           ELE04740
     & '       FORCE     ',' DISPLACMENT ',' STIFFNESS   ')             ELE04750
  492 FORMAT (I10,I5,2X,A8,2I6,1P,3G15.6)                               ELE04760
  493 FORMAT ('  F:',1P,G16.8,'  FT:',G16.8,'  P:',G16.8)               ELE04770
  494 FORMAT ('  I:',I5,'  LM(I):',I6,'  FUB(I):',1P,G16.8)             ELE04780
C                                                                       ELE04790
C------ ADJUST F FOR UNBALANCED LOAD P                                 ELE04800
        F=P-FT                                                          ELE04810
      RETURN                                                            ELE04820
C************************* DETERMINE TOTAL FORCES, ENERGIES ECT *********ELE04830
  500 CONTINUE                                                          ELE04840
      IF( ICOMN .NE. 1 .AND. ICOMN .NE. 2 ) THEN                        ELE04850
       IF (PRINT.AND.FIRST) WRITE (6,491) STEPID                        ELE04860
C                                                                       ELE04870
C                                                                       ELE04880
C------ TOTAL FORCES, DISPLACEMENTS AND VELOCITIES                      ELE04890
        FT=FT+F                                                         ELE04900
        RT=RT+R                                                         ELE04910
        VT=VT+V                                                         ELE04920
        IF ( ABS(FT).GT.ABS(FMAX(1)) ) THEN                             ELE04930
          FMAX(1)=FT                                                    ELE04940
          FMAX(2)=RT                                                    ELE04950
        ENDIF                                                           ELE04960
C                                                                       ELE04970
C------ PRINT DATA                                                     ELE04980
        HH= H/LENGTH                                                    ELE04990
        IF (PRINT) WRITE(6,492) IELNO,1,TYPE(KTYPE),NODEI,NODEJ,FT,RT,HHELE05000
      ELSE IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                    ELE05010
          FT=RSAF(1)                                                    ELE05020
          IF (PRINT.AND.FIRST) WRITE (6,497) STEPID                     ELE05030
          IF (PRINT) WRITE(6,492) IELNO,1,TYPE(KTYPE),NODEI,NODEJ,FT    ELE05040
      ENDIF                                                             ELE05050
C                                                                       ELE05060
  497 FORMAT (//'       SPRING FORCES...',5X,A/                         ELE05070
     & '   ELEMENT LOAD',T17,'      TYPE',' NODEI',' NODEJ',           ELE05080
     & '       FORCE     ')                                            ELE05090
C                                                                       ELE05100
C------ TRANSFER TEMPROARY HYSTERESIS DATA TO PERMANENT STORAGE...      ELE05110
      ISE2=ISE/2                                                        ELE05120
      IF (.NOT. ELSTIC) THEN                                            ELE05130
        DO 510 I=1,ISE2                                                 ELE05140
  510     SE(I)=SE(I+ISE2)                                              ELE05150
      ENDIF                                                             ELE05160
C                                                                       ELE05170
      RETURN                                                            ELE05180
C                                                                       ELE05190
C************************* DETERMINE FIXED END FORCES, AND ADD TO FORCE=ELE05200
  600 RETURN                                                            ELE05210
C                                                                       ELE05220
C------- FIXED END FORCES ARE NOT AVAILABLE FOR THE SPRING              ELE05230
C                                                                       ELE05240
C********************* WRITE MEMBER FORCES TO OUTPUT FILE **************ELE05250
  700 CONTINUE                                                          ELE05260
C                                                                       ELE05270
      CALL MATLIB                                                       ELE05280
     &  ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE, DUCT,EXCR,    ELE05290
     &    FT   ,F      ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,      ELE05300
     &    MAT  ,MATTYP,SE,IELNO,DAMAGE)                                 ELE05310
C                                                                       ELE05320
      WRITE (IUNIT,710)  NODEI,NODEJ,KTYPE,FT,RT,H,EESE,EPSE,          ELE05330
     &                   (DUCT(I),EXCR(I),I=1,3)                       ELE05340
  710 FORMAT (3I6,1P,5G12.4/6G12.4)                                     ELE05350
C                                                                       ELE05360
      RETURN                                                            ELE05370
C                                                                       ELE05380
C******************** READ AND PRINT MEMBER FORCES FROM OUTPUT FILE *****ELE05390
  800 CONTINUE                                                          ELE05400
      BACKSPACE(IUNIT)                                                  ELE05410
      READ (IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT                        ELE05420
      ISTEP=-IWRITE                                                     ELE05430
C                                                                       ELE05440
  810 READ (IUNIT,815,END=830) NODEI,NODEJ,KTYPE,FT,RT,H,ESE,PSE,       ELE05450
     &                   (WORK(I),I=1,6)                               ELE05460
  815 FORMAT (3I6,5G12.4/6G12.4)                                        ELE05470
        ISTEP=ISTEP+IWRITE                                              ELE05480
        T=TO+DT*ISTEP                                                   ELE05490
        IF (ISTEP.EQ.0) WRITE (6,805) NODEI,NODEJ,TYPE(KTYPE)           ELE05500
        IF (MOD(ISTEP,INC).EQ.0)                                        ELE05510
     &  WRITE (6,820) ISTEP,T,FT,RT,H,ESE,PSE                           ELE05520
        GO TO 810                                                       ELE05530
C                                                                       ELE05540
  805 FORMAT (//'       SPRING FORCES...',/                             ELE05550
     & 5X,'NODEI:',I6,5X,'NODEJ:',I6,5X,'SPRING TYPE:',A,//             ELE05560
     & '       STEP     TIME       ',                                  ELE05570
     & '     FORCE     ',' DISPLACMENT ',' STIFFNESS',                 ELE05580
     & '       ESE       ',' PSE       ')                              ELE05590
  820 FORMAT (I10,1P,G15.5,6G13.5)                                      ELE05600
C                                                                       ELE05610
  830 WRITE (6,840) (WORK(I),I=1,6)                                     ELE05620
  840 FORMAT (/'T10,' MAXIMUN DUCTILITIES AND EXCURSION RATIOS'/        ELE05630
     &        T10,'----------------------------------------'/          ELE05640
     &  T15,'DISPLACEMENT DEFINATION: U1=',1P,G12.4,5X,'E1=',G12.4/     ELE05650
     &  T15,'ENERGY DEFINATION #1:    U2=',1P,G12.4,5X,'E2=',G12.4/     ELE05660
     &  T15,'ENERGY DEFINATION #2:    U3=',1P,G12.4,5X,'E3=',G12.4)     ELE05670
      RETURN                                                            ELE05680
C                                                                       ELE05690
C******************** DETERMINE THE LENGTH OF THE MATERIAL DATA         ELE05700
  900 CONTINUE                                                          ELE05710
      NHYST=1                                                           ELE05720
      ISE=0                                                             ELE05730
C                                                                       ELE05740
        MAT   = RINPUT( 1)                                              ELE05750
        CALL MATLIB                                                     ELE05760
     &  ( 0    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE, DUCT,EXCR,    ELE05770
     &    FT   ,F      ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,      ELE05780
     &    MAT  ,MATTYP,SE,IELNO,DAMAGE)                                 ELE05790
        ISE=ISE+LELEM                                                   ELE05800
C                                                                       ELE05810
C------- DOUBLE THE STORAGE, TO ALLOW FOR TEMPROARY VALUES...           ELE05820
        ISE=ISE*2                                                       ELE05830
      RETURN                                                            ELE05840
C                                                                       ELE05850
C******************** DETERMINE THE STRAIN ENERGY                       ELE05860
 1000 CONTINUE                                                          ELE05870
C                                                                       ELE05880
C******************** DUCTILITIES AND EXCURSION RATIO'S                 ELE05890
 1100 CONTINUE                                                          ELE05900
C.... OPTIONS 10 AND 11 HAVE THE SAME CODE FOR THIS ELEMENT...          ELE05910
C.... THE ENERGY WAS CALCULATED WITH THE LAST CALL FOR INCRIMENTAL FORCEELE05920
C      THE TOTAL FORCES HAVE NOT BEEN CALLED YET,THUS THE ENERGIES RESIDEELE05930
C      IN THE MATRIX SE BEGINING AT ADDRESS ISE21.                      ELE05940
C                                                                       ELE05950
      ISE21=ISE/2 +1                                                    ELE05960
      CALL MATLIB                                                       ELE05970
     &  ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE, DUCT,EXCR,    ELE05980
     &    FT   ,F      ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,      ELE05990
     &    MAT  ,MATTYP,SE(ISE21),IELNO,DAMAGE)                          ELE06000
C                                                                       ELE06010
      IF ( DUCFAG .EQ. 0 ) THEN                                         ELE06020
        IF( ABS(DUCT(1)) .LT. 1 ) DUCFAG=0                              ELE06030
        IF( ABS(DUCT(1)) .GE. 1 ) DUCFAG=ABS(DUCT(1))                   ELE06040
      ENDIF                                                             ELE06050
C                                                                       ELE06060
      IF (ELSTIC) EESE=(FT+F)*(RT+R)*0.50                               ELE06070
C                                                                       ELE06080
      IF (PRINT.AND.FIRST) WRITE (6,1191)                               ELE06090
      IF (PRINT) WRITE (6,1192) IELNO,(DUCT(I),EXCR(I),I=1,3)           ELE06100
 1191 FORMAT (//'  SPRING DUCTILITIES AND EXCURSION RATIOS....',5X/     ELE06110
     & '   ELEMENT',T12,'|- DUCTILITY DEFINATION #1 -| ',               ELE06120
C                    123456789ABCDE123456789ABCDE                       ELE06130
     &                '|- DUCTILITY DEFINATION #2 -| ',               ELE06140
     &                '|- DUCTILITY DEFINATION #3 -| ',/ T12,          ELE06150
     & '   DUCTILITY    EXCURSION     ',                               ELE06160
     & '   DUCTILITY    EXCURSION     ',                               ELE06170
     & '   DUCTILITY    EXCURSION     ')                               ELE06180
C       123456789ABCDE123456789ABCDE                                    ELE06190
 1192 FORMAT (I10,1P,6G15.6)                                            ELE06200
C                                                                       ELE06210
      RETURN                                                            ELE06220
C                                                                       ELE06230
C******************** MAXIMUM FORCES                                    ELE06240
 1200 CONTINUE                                                          ELE06250
      IF (FIRST) WRITE (6,491) 'MAXIMUN LOAD AND DISPL AT MAXIMUM LOAD 'ELE06260
     &          //' NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'  ELE06270
      WRITE(6,492) IELNO,0,TYPE(KTYPE),NODEI,NODEJ,FMAX(1),FMAX(2)      ELE06280
      RETURN                                                            ELE06290
C                                                                       ELE06300
C                                                                       ELE06310
C******************** RESET INTERNAL FORCES                             ELE06320
 1300 CONTINUE                                                          ELE06330
C                                                                       ELE06340
C----- ZERO MATRICIES                                                  ELE06350
          F =0                                                          ELE06360
          FT=0                                                          ELE06370
          R =0                                                          ELE06380
          RT=0                                                          ELE06390
          V =0                                                          ELE06400
          VT=0                                                          ELE06410
          FMAX(1)=0                                                     ELE06420
          FMAX(2)=0                                                     ELE06430
      DO 1310 I=1,ISE                                                   ELE06440
 1310   SE(I)=0                                                         ELE06450
      WRITE (6,1330) IELNO                                              ELE06460
 1330 FORMAT (' SPRING......... ELEMENT #',I6,                          ELE06470
     & '  INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO')      ELE06480
      CALL MATLIB                                                       ELE06490
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE, DUCT,EXCR,    ELE06500
     &    FT   ,F      ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,      ELE06510
     &    MAT  ,MATTYP,SE,IELNO,DAMAGE)                                 ELE06520
      RETURN                                                            ELE06530
C                                                                       ELE06540
C                                                                       ELE06550
C******************** DAMAGE INDEX ***********************             ELE06560
 1400 IF (ELSTIC) RETURN                                                ELE06570
C                                                                       ELE06580
      FMAX1=FMAX(1)                                                     ELE06590
      FMAX2=FMAX(2)                                                     ELE06600
      CALL MATLIB                                                       ELE06610
     &  ( 5    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE, DUCT,EXCR,    ELE06620
     &    FMAX1,0.     ,FMAX2,  0., 0.    ,0.    ,LELEM ,LMAT  ,      ELE06630
     &    MAT  ,MATTYP,SE,IELNO,DAMAGE)                                 ELE06640
C                                                                       ELE06650
      IF (PRINT.AND.FIRST) WRITE (6,1410)                               ELE06660
      IF (PRINT) WRITE (6,1420) IELNO,DAMAGE,FMAX1,FMAX2,EESE,EPSE      ELE06670
 1410 FORMAT (/                                                         ELE06680
```

```
     &  '    ELEMENT',T12,' DAMAGE INDEX       F MAX      ',                ELE06690
C                          1234567890ABCDE1234567890ABCDE                    ELE06700
     &                    '      D MAX        ESE        ',                ELE06710
     &                    '      PSE        ')                              ELE06720
 1420 FORMAT (I10,1P,6G15.6)                                                 ELE06730
C                                                                            ELE06740
      DAMAGE=DAMAGE*(EESE+EPSE)                                              ELE06750
      RETURN                                                                 ELE06760
C                                                                            ELE06770
      END                                                                    ELE06780
C*************************************************************************** ELE00010
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                           ELE00020
C         END DEBUG                                                          ELE00030
C*************************************************************************** ELE00040
      SUBROUTINE ELE03                                                       ELE00050
     &  (IOPT       ,IUNIT      ,NEWK       ,FIRST      ,PRINT      ,       ELE00060
     &   NDOF       ,NLOAD      ,MAXNOD     ,NNODE      ,NCOS       ,       ELE00070
     &   ID         ,IDOF       ,COORD      ,COSINE     ,JTCOS      ,       ELE00080
     &   CONST      ,DISPL      ,VELOC      ,FORCE      ,FVB        ,       ELE00090
     &   IREL       ,BUG        ,IBUG       ,ACCEL      ,KGDATA     ,       ELE00100
     &   GRAV       ,PGEOM      ,ELSTIC     ,NAME       ,STEPID     ,       ELE00110
     &   ISLNO      ,RINPUT     ,AXIAL      ,ESE        ,PSE        ,       ELE00120
     &   INC        ,DAMAGE     ,DUCFAG     ,                               ELE00130
     &   IELTYP     ,IELDOF     ,ISEB       ,ISES       ,ISEA       ,       ELE00140
     &   ISE        ,J1         ,J2         ,J3         ,J4         ,       ELE00150
     &   MATB       ,MATS       ,MATA       ,BLANK      ,R          ,       ELE00160
     &   RT         ,F          ,FT         ,V          ,VT         ,       ELE00170
     &   PKG        ,LKG        ,LM         ,LMKG       ,A          ,       ELE00180
     &   SB         ,SS         ,SA         ,TEMP1      ,TEMP2      ,       ELE00190
     &   KGDOF      ,LENGTH     ,FMAX       ,SE         ,STIFF      )       ELE00200
C                                                                            ELE00210
      IMPLICIT REAL(A-H,O-Z)                                                 ELE00220
C                                                                            ELE00230
      COMMON /RSA/ICOMN,RSAF(12)                                             ELE00240
      REAL ICOMN                                                             ELE00250
C                                                                            ELE00260
      DOUBLE PRECISION SX,SXX,SY,SXY,SN                                      ELE00270
      LOGICAL ERR,FIRST,PRINT,BUG,FALSE,NEWK,ELSTIC ,AXIAL,BTEST             ELE00280
      CHARACTER*80 NAME                                                      ELE00290
      CHARACTER*(*) STEPID                                                   ELE00300
      DOUBLE PRECISION   SAT,DSTIFF(600)                                     ELE00310
      DIMENSION  IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS)            ELE00320
      DIMENSION CONST(MAXNOD,6),    ID(MAXNOD)   ,JTCOS(MAXNOD)              ELE00330
      DIMENSION SAT(3,24),STIFF(600),KGDATA(3) ,SAT2(6,24)                   ELE00340
      DIMENSION AT(24,6),A(24,3),S(6,6),LM(24),AC(6,3),LMKG(24)              ELE00350
      DIMENSION CT(3,3),BS(3,3),VX(3),VY(3),VZB(3),VXT(3),VZT(3),VZB(3)      ELE00360
      DIMENSION SE(ISE),A11(6,3),A22(6,3),A44(6,3),A33(6,3)                  ELE00370
      DIMENSION F(3),R(3),V(3),FT(3),RT(3),VT(3),P(3)                        ELE00380
      DIMENSION DISPL(NDOF,NLOAD),FVB(NDOF),EFVB(24),FMAX(6)                 ELE00390
      DIMENSION VELOC(NDOF)                                                  ELE00400
      DIMENSION FORCE(NDOF,NLOAD)                                            ELE00410
      DIMENSION RINPUT(100),WORK(1),DUCT(3,3),EXCR(6,3)                      ELE00420
      REAL LENGTH                                                            ELE00430
      DIMENSION MD(25)                                                       ELE00440
      DATA MD/ 1,  2,  4,  7, 11, 16, 22, 29, 37, 46, 56, 67,               ELE00450
     &         79, 92,106,121,137,154,172,191,211,232,254,277,301/           ELE00460
C                                                                            ELE00470
C*************************************************************************** ELE00480
C VARIABLES:                                                                 ELE00490
C*************************************************************************** ELE00500
C---------- GLOBAL VARIABLES                                                 ELE00510
C IOPT   = 1, INITIALIZE BEAM COLUMN ELEMENT                                 ELE00520
C        = 2, CALCULATE GEOMETRIC STIFFNESS                                  ELE00530
C        = 3, FORM STIFFNESS                                                 ELE00540
C        = 4, CALCULATE INCREMENTAL FORCES                                   ELE00550
C        = 5, CALCULATE TOTAL FORCES AND ENERGIES                            ELE00560
C IUNIT  = OUTPUT FILE UNIT # FOR PRINTING OUTPUT                            ELE00570
C ERR    = ERR OR FLAG. IF AN ERR OR HAS OCCURED, ERR=.TRUE.                 ELE00580
C FIRST  = FLAG, FIRST=.TRUE., PRINT HEADERS                                 ELE00590
C PRINT  = FLAG, PRINT=.TRUE., PRINT DATA                                    ELE00600
C NDOF   = # OF GLOBAL DOF                                                   ELE00610
C NLOAD  = # OF LOAD COMBINATIONS                                            ELE00620
C MAXNOD = # ROW DIMENSION OF NODE ARRAY                                     ELE00630
C NNODE  = # OF NODES                                                        ELE00640
C ID     = ARRAY OF EXTERNAL NODE NUMBERS                                    ELE00650
C IDOF   = ARRAY OF DEGREES OF FREEDOM                                       ELE00660
C COORD  = ARRAY OF COORDINATES                                              ELE00670
C COSINE = ARRAY OF DIRECTION COSINES OF NODES                               ELE00680
C CONST  = ARRAY OF CONSTRAINT TRANSFORMATIONS                               ELE00690
C DISPL  = GLOBAL DISPLACEMENT MATRIX                                        ELE00700
C Z      = REAL GLOBAL STORTAG VECTOR                                        ELE00710
C NZ     = INTEGER GLOBAL STORTAG VECTOR                                     ELE00720
C I3MAT  = LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR                ELE00730
C NAME   = ELEMENT NAME                                                      ELE00740
C ISLNO  = ELEMENT NUMBER                                                    ELE00750
C RINPUT = INPUT DATA                                                        ELE00760
C STIFF  = OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)                  ELE00770
C LM     = OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                   ELE00780
C---------- ELEMENT VARIABLES                                                ELE00790
C H,Q,AI,AJ,C,BI,BJ,S22,S26,S212,S33,S35,S311,D                              ELE00800
C        = INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX              ELE00810
C PKG    = INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX...              ELE00820
C R      = INCRIMENTAL DISPLACEMENTS                                         ELE00830
C RT     = TOTAL DISPLACEMENTS                                               ELE00840
C F      = INCRIMENTAL FORCES                                                ELE00850
C FT     = TOTAL FORCES                                                      ELE00860
C LM     = LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                          ELE00870
C CTS    = LOCAL TO GLOBAL ROTATION MATRIX START                             ELE00880
C CTE    = LOCAL TO GLOBAL ROTATION MATRIX END                               ELE00890
C BS     = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX START     ELE00900
C BE     = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END       ELE00910
C STIFF  = ELEMENT STIFFNESS                                                 ELE00920
C IELTYP = ELEMENT TYPE = 10                                                 ELE00930
C NODEI  = EXTERNAL JOINT NUMBER, END I                                      ELE00940
C NODEJ  = EXTERNAL JOINT NUMBER, END J                                      ELE00950
C JOINTI = INTERNAL JOINT NUMBER, END I                                      ELE00960
C JOINTJ = INTERNAL JOINT NUMBER, END J                                      ELE00970
C---------- TEMPROARY VARIABLES                                              ELE00980
C VX     = VECTOR DEFINING THE ELEMENT X AXIS                                ELE00990
C VY     = VECTOR DEFINING THE ELEMENT Y AXIS                                ELE01000
C COSLOC = LOCAL COSINE MATRIX                                               ELE01010
C A      = GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                       ELE01020
C S      = ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART        ELE01030
C SAT    = PRODUCT OF S*A                                                    ELE01040
C ASAT   = PRODUCT OF A*SAT                                                  ELE01050
C*************************************************************************** ELE01060
C                                                                            ELE01070
C                                                                            ELE01080
C------- CHOOSE OPTION                                                       ELE01090
       IF (IOPT.LT.1 .OR.IOPT.GT.14)                                         ELE01100
     & WRITE (6,*) 'INVALID OPTION IN ELE-02, IOPT=',IOPT                    ELE01110
C                                                                            ELE01120
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,             ELE01130
     &                                          1300,1400),IOPT               ELE01140
C                                                                            ELE01150
  100 CONTINUE                                                               ELE01160
C                                                                            ELE01170
C----- INTERPERTATE INPUT DATA                                               ELE01180
       IELTYP= 3                                                             ELE01190
       MATB  = RINPUT( 1)                                                    ELE01200
       MATS  = RINPUT( 2)                                                    ELE01210
       MATA  = RINPUT( 3)                                                    ELE01220
       NODE1 = RINPUT( 4)                                                    ELE01230
       NODE2 = RINPUT( 5)                                                    ELE01240
       NODE3 = RINPUT( 6)                                                    ELE01250
       NODE4 = RINPUT( 7)                                                    ELE01260
       ALPIN = RINPUT( 8)                                                    ELE01270
       PKG   = RINPUT( 9)                                                    ELE01280
C                                                                            ELE01290
C----- GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S                 ELE01300
       J1=IQUICK(NODE1,ID,NNODE)                                             ELE01310
       J2=IQUICK(NODE2,ID,NNODE)                                             ELE01320
       J3=IQUICK(NODE3,ID,NNODE)                                             ELE01330
       J4=IQUICK(NODE4,ID,NNODE)                                             ELE01340
C                                                                            ELE01350
C*************************************************************************** ELE01360
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                                 ELE01370
C*************************************************************************** ELE01380
C----- DEFINE VECTOR X, AT BOTTOM OF THE WALL                                ELE01390
       VXB(1)=COORD(J4,1)-COORD(J3,1)                                        ELE01400
       VXB(2)=COORD(J4,2)-COORD(J3,2)                                        ELE01410
       VXB(3)=COORD(J4,3)-COORD(J3,3)                                        ELE01420
C                                                                            ELE01430
C----- DEFINE VECTOR Y, AT TOP OF THE WALL                                   ELE01440
       VXT(1)=COORD(J1,1)-COORD(J2,1)                                        ELE01450
       VXT(2)=COORD(J1,2)-COORD(J2,2)                                        ELE01460
       VXT(3)=COORD(J1,3)-COORD(J2,3)                                        ELE01470
C                                                                            ELE01480
C----- CALCULATE Y VECTOR ...                                                ELE01490
       VT(1)=(COORD(J1,1)+COORD(J2,1)-COORD(J3,1)-COORD(J4,1))/2.            ELE01500
       VT(2)=(COORD(J1,2)+COORD(J2,2)-COORD(J3,2)-COORD(J4,2))/2.            ELE01510
       VT(3)=(COORD(J1,3)+COORD(J2,3)-COORD(J3,3)-COORD(J4,3))/2.            ELE01520
C                                                                            ELE01530
C----- CALCULATE AVERAGE LENGTH,ALPHA AND BETA...                            ELE01540
       LENGTH=SQRT(VT(1)**2+VT(2)**2+VT(3)**2)                               ELE01550
       IF (LENGTH.LE.0) THEN                                                 ELE01560
          WRITE(6,101) ISLNO,LENGTH                                          ELE01570
  101     FORMAT(1X,2Q('*'),'ER','ROR - LENGTH IS LE 0',/                    ELE01580
     &    5X,'IELNO=',I5,5X,5X,'LENGTH',1P,G15.6/                            ELE01590
     &    5X,'REVISE INPUT.............................')                    ELE01600
          LENGTH=1.0                                                         ELE01610
       ENDIF                                                                 ELE01620
C                                                                            ELE01630
       ALPHA=ALPIN*LENGTH                                                    ELE01640
       BETA = LENGTH-ALPHA                                                   ELE01650
C                                                                            ELE01660
C----- NORMALIZE Y VECTOR ...                                                ELE01670
       VT(1)=VT(1)/LENGTH                                                    ELE01680
       VT(2)=VT(2)/LENGTH                                                    ELE01690
       VT(3)=VT(3)/LENGTH                                                    ELE01700
C                                                                            ELE01710
C----- CALCULATE THE WIDTH ...                                               ELE01720
       CALL VCROSS(VXT,VT,VXT)                                               ELE01730
       CALL VCROSS(VZB,VT,VZB)                                               ELE01740
       WT = SQRT(VXT(1)**2 + VXT(2)**2 + VXT(3)**2)                          ELE01750
       WB = SQRT(VZB(1)**2 + VZB(2)**2 + VZB(3)**2)                          ELE01760
       W  = (WT+WB)/2.0                                                      ELE01770
C                                                                            ELE01780
C----- CHECK FOR TWIST                                                       ELE01790
       TWIST=ACOS(VCOS(VZT,VZB))*57.2958                                     ELE01800
       IF (ABS(TWIST).GT. 5.0) WRITE (6,102) TWIST                           ELE01810
  102 FORMAT(5X,10('*'),' WARNING THE WALL IS TWISTED=',1P,G12.4,            ELE01820
     &   ' DEGREES, REVISE INPUT AND RERUN')                                 ELE01830
C                                                                            ELE01840
C----- CALCULATE VX, AND NORMALIZE                                           ELE01850
       CALL VCROSS(VX,VT,VZB)                                                ELE01860
       WX=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)                                   ELE01870
       VX(1)=VX(1)/WX                                                        ELE01880
       VX(2)=VX(2)/WX                                                        ELE01890
       VX(3)=VX(3)/WX                                                        ELE01900
C                                                                            ELE01910
C----- CHECK FOR TWIST                                                       ELE01920
       IF ( BTEST(IBUG,7) ) THEN                                             ELE01930
          WRITE (6,*) 'JOINTS',J1,J2,J3,J4                                   ELE01940
          WRITE (6,*) '                VX',VX                                ELE01950
          WRITE (6,*) '                VT',VT                                ELE01960
          WRITE (6,*) '                VZ',VZB                               ELE01970
          WRITE (6,*) 'LENGTH,ALPHA,BETA',LENGTH,ALPHA,BETA                  ELE01980
       ENDIF                                                                 ELE01990
C                                                                            ELE02000
C----- GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES                          ELE02010
C--- JOINT #1                                                                ELE02020
          CALL ROTXYZ(VX,VT,COSINE(1,1,JTCOS(J1)),CT,                        ELE02030
     &                      0.,0.,0.,BS,  CONST(J1,1),MAXNOD)                ELE02040
          DO 21 I=1,3                                                        ELE02050
          DO 21 J=1,3                                                        ELE02060
            A11(I  ,J)=CT(I,J)                                               ELE02070
   21       A11(I+3,J)=BS(I,J)                                               ELE02080
C--- JOINT #2                                                                ELE02090
          CALL ROTXYZ(VX,VT,COSINE(1,1,JTCOS(J2)),CT,                        ELE02100
     &                      0.,0.,0.,BS,  CONST(J2,1),MAXNOD)                ELE02110
          DO 22 I=1,3                                                        ELE02120
          DO 22 J=1,3                                                        ELE02130
            A22(I  ,J)=CT(I,J)                                               ELE02140
   22       A22(I+3,J)=BS(I,J)                                               ELE02150
C--- JOINT #3                                                                ELE02160
          CALL ROTXYZ(VX,VT,COSINE(1,1,JTCOS(J3)),CT,                        ELE02170
     &                      0.,0.,0.,BS,  CONST(J3,1),MAXNOD)                ELE02180
          DO 23 I=1,3                                                        ELE02190
          DO 23 J=1,3                                                        ELE02200
            A33(I  ,J)=CT(I,J)                                               ELE02210
   23       A33(I+3,J)=BS(I,J)                                               ELE02220
C--- JOINT #4                                                                ELE02230
          CALL ROTXYZ(VX,VT,COSINE(1,1,JTCOS(J4)),CT,                        ELE02240
     &                      0.,0.,0.,BS,  CONST(J4,1),MAXNOD)                ELE02250
          DO 24 I=1,3                                                        ELE02260
          DO 24 J=1,3                                                        ELE02270
            A44(I  ,J)=CT(I,J)                                               ELE02280
   24       A44(I+3,J)=BS(I,J)                                               ELE02290
C                                                                            ELE02300
          IF ( BTEST(IBUG,7) ) THEN                                          ELE02310
               WRITE (6,*) 'JOINT #1,  J1',J1                                ELE02320
               CALL WMATRX(A11  , 6, 3,'A11  ')                              ELE02330
               WRITE (6,*) 'JOINT #2,  J2',J2                                ELE02340
               CALL WMATRX(A22  , 6, 3,'A22  ')                              ELE02350
               WRITE (6,*) 'JOINT #3,  J3',J3                                ELE02360
               CALL WMATRX(A33  , 6, 3,'A33  ')                              ELE02370
               WRITE (6,*) 'JOINT #4,  J4',J4                                ELE02380
               CALL WMATRX(A44  , 6, 3,'A44  ')                              ELE02390
          ENDIF                                                              ELE02400
C                                                                            ELE02410
C----- ZERO MATRICIES                                                        ELE02420
       DO 25 I=1,3                                                           ELE02430
          FMAX(I  )= 0                                                       ELE02440
          FMAX(I+3)= 0                                                       ELE02450
          F (I)=0                                                            ELE02460
          FT(I)=0                                                            ELE02470
          R (I)=0                                                            ELE02480
          RT(I)=0                                                            ELE02490
          V (I)=0                                                            ELE02500
   25     VT(I)=0                                                           ELE02510
       DO 27 I=1,24                                                          ELE02520
          LM  (I)=0                                                          ELE02530
          LMKG(I)=0                                                          ELE02540
          DO 26 J=1,3                                                        ELE02550
   26       A(I,J)=0.00                                                     ELE02560
          DO 27 K=1,6                                                        ELE02570
   27       AT(I,K)=0.00                                                    ELE02580
C                                                                            ELE02590
C----- ASSEMBLE TRANSFORMATION MATRIX A = A5 X A4 X A3                       ELE02600
C  NOTE: AT IS DIMENSIONED AS A 24X6 MATRIX, BUT ONLY THE FIRST IELDOF       ELE02610
C        ROW'S CONTAIN INFORMATION. THE BALANCE OF THE MATRIX IS             ELE02620
C        UNDEFINED.  THE GLOBAL DOF COORESPONDING TO THE ROW'S OF A          ELE02630
C        ARE STORED IN VECTOR LM.  AGAIN, ONLY THE FIRST IELDOF ROW'S        ELE02640
C        OF LM ARE DEFINED.                                                  ELE02650
       IELDOF=0                                                              ELE02660
       DO 50 I=1,4                                                           ELE02670
       DO 50 J=1,6                                                           ELE02680
C                                                                            ELE02690
C              REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF...          ELE02700
          IF ( (IDOF(J1,J).EQ.IDOF(J4,J)) .OR.                               ELE02710
     &         (IDOF(J1,J).EQ.IDOF(J3,J)) .OR.                               ELE02720
     &         (IDOF(J2,J).EQ.IDOF(J4,J)) .OR.                               ELE02730
     &         (IDOF(J2,J).EQ.IDOF(J3,J)) ) GO TO 50                         ELE02740
```

```
C
      IELDOF=IELDOF+1
      IF (I.EQ.1) THEN
          LMTI=IDOF(J1,J)
          AT(IELDOF,1)=A11(J,1)
          AT(IELDOF,2)=A11(J,2)
      ELSE IF (I.EQ.2) THEN
          LMTI=IDOF(J2,J)
          AT(IELDOF,3)=A22(J,2)
      ELSE IF (I.EQ.3) THEN
          LMTI=IDOF(J3,J)
          AT(IELDOF,4)=A33(J,2)
      ELSE IF (I.EQ.4) THEN
          LMTI=IDOF(J4,J)
          AT(IELDOF,5)=A44(J,1)
          AT(IELDOF,6)=A44(J,2)
      ENDIF
C
      LM(IELDOF)=LMTI
      IF((AT(IELDOF,1).EQ.0).AND.(AT(IELDOF,2).EQ.0) .AND.
     &    (AT(IELDOF,3).EQ.0).AND.(AT(IELDOF,4).EQ.0) .AND.
     &    (AT(IELDOF,5).EQ.0).AND.(AT(IELDOF,6).EQ.0)) THEN
          IELDOF=IELDOF-1
      ELSE IF (IELDOF.GT.1) THEN
          DO 49  II=1,IELDOF-1
            IF (LM(II).EQ.LM(IELDOF)) THEN
              DO 48 JJ=1,6
                AT(II,JJ)=AT(II,JJ)+AT(IELDOF,JJ)
48              AT(IELDOF,JJ)=0
              IELDOF=IELDOF-1
              GO TO 50
            ENDIF
49        CONTINUE
      ENDIF
C
50    CONTINUE
C
C------ A2A1 FOR STIFFNESS ONLY...
      AC(1,1)=  0.
      AC(2,1)= -1./W
      AC(3,1)=  1./W
      AC(4,1)= -1./W
      AC(5,1)=  0.
      AC(6,1)=  1./W
      AC(1,2)=  1.
      AC(2,2)=  ALPHA/W
      AC(3,2)= -ALPHA/W
      AC(4,2)= -BETA /W
      AC(5,2)= -1.
      AC(6,2)=  BETA /W
      AC(1,3)=  0.
      AC(2,3)=  1./2.
      AC(3,3)=  1./2.
      AC(4,3)= -1./2.
      AC(5,3)=  0.
      AC(6,3)= -1./2.
C
      DO 55 I=1,IELDOF
      DO 55 J=1,3
        A(I,J)=0.
        DO 55 K=1,6
55        A(I,J)=A(I,J)+AT(I,K)*AC(K,J)
C
      IF ( BTEST(IBUG,7) ) THEN
          CALL WMTRX2( AT    ,IELDOF,6,' AT    ',24)
          CALL WMATRX( AC    , 6,3,' AC  ')
          CALL WMTRX2( A     ,IELDOF,3,' A     ',24)
      ENDIF
C
C
C----- GET SPRING STIFFNESS
       FALSE=.FALSE.
       LSTTYP=0
       CALL MATLIB
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &    ESEB,PSEB,DUCT(1,1),EXCR(1,1),
     &    FT    ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,
     &    MATB  ,MATTYP,SE(ISEB),IELNO,DAMAGE)
       CALL MATLIB
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &    ESES,,PSES,DUCT(1,2),EXCR(1,2),
     &    FT    ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,
     &    MATS  ,MATTYP,SE(ISES),IELNO,DAMAGE)
       CALL MATLIB
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &    ESEA,PSEA,DUCT(1,3),EXCR(1,3),
     &    FT    ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,
     &    MATA  ,MATTYP,SE(ISEA),IELNO,DAMAGE)
C
C
      ESE=0.
      PSE=0.
C
C     WRITE (6,*) 'AFTER MAT','LIB IN ELE02 '
  111 FORMAT (OP,' ZMATRX(',I6,')=',I15,1P,G20.10)
C
C------- SET UP THE LOCAL STIFFNESS MATRIX
      SB= SE(ISEB)
      SS= SE(ISES)
      SA= SE(ISEA)
      IF ( BTEST(IBUG,7) ) THEN
        WRITE (6,*) 'BENDING  STIFFNESS KB=',SB
        WRITE (6,*) 'SHEAR    STIFFNESS KS=',SS
        WRITE (6,*) 'AXIAL    STIFFNESS KA=',SA
      ENDIF
C
C----- CALCULATE SAT
      SBL=SB/LENGTH
      SSL=SS/LENGTH
      SAL=SA/LENGTH
      DO 60 J=1,IELDOF
        SAT(1,J)=SBL*A(J,1)
        SAT(2,J)=SSL*A(J,2)
60      SAT(3,J)=SAL*A(J,3)
C
C----- CALCULATE ASAT AND CONVERT TO HALF STORAGE MODE  BY COLUMNS
C
C         1  3  6 10 15 21 28 36 45 55 66 78     300.  NUMBER INDICATES
C            2  5  9 14 20 27 35 44 54 65 77      .    LOCATION OF DATA
C               4  8 13 19 26 34 43 53 64 76      .    IN ARRAY STIFF
C                  7 12 18 25 33 42 52 63 75      .
C                    11 17 24 32 41 51 62 74      .
C                       16 23 31 40 50 61 73      .
C                          22 30 39 49 60 72      .
C                             29 38 48 59 71      .
C                                37 47 58 70      .
C                                   46 57 69      .
C                                      56 68      .
C                                         67      .
C                                            .    .
C                                             .   .
C                                              .  .
C                                               . .
      L=0
      DO 70 J=1,IELDOF
        DO 70 I=J,1,-1
        L=L+1
        DSTIFF(L)=0
        DO 70 K=1,3
70        DSTIFF(L)=DSTIFF(L)+A(I,K)*SAT(K,J)
      DO 71 J=1,L
71        STIFF(J)=DSTIFF(J)
C
      IF ( BTEST(IBUG,7) ) THEN
         WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)
         CALL LMATRX(STIFF,MD,IELDOF,(L+1),'WALL STIFFNESS')
      ENDIF
C
C-----------------------------------------------------------------
C  ASSEMBLE GEOMETRIC STIFFNESS MATRIX FOR A UNIT LOAD...
C-----------------------------------------------------------------
      LKG=L+1
      IF ( PKG.EQ.0 .AND. KGDATA(1).NE.2 ) GO TO 180
      IF (KGDATA(1).EQ.0 .OR. KGDATA(2).EQ.0 ) GO TO 180
      AXIAL=.TRUE.
C
C---------- DETERMINE THE AXIAL LOAD TO BE USED FOR KG...
          IF (KGDATA(1).EQ.1) THEN
                PGEOM =  PKG
          ELSE IF (KGDATA(1).EQ.2) THEN
                PGEOM = -F(3)
          ENDIF
C
C----- ASSEMBLE TRANSFORMATION MATRIX A
C  NOTE: AT IS DIMMENSIONED AS A 12X6 MATRIX, BUT ONLY THE FIRST IELDOF
C        ROW'S CONTAIN INFORMATION. THE BALANCE OF THE MATRIX IS
C        UNDEFINED.  THE GLOBAL DOF COORESPONDING TO THE ROW'S OF A
C        ARE STORED IN VECTOR LM.  AGAIN, ONLY THE FIRST IELDOF ROW'S
C        OF LM ARE DEFINED.
       DO 140 I=1,24
       DO 140 K=1,6
  140    AT(I,K)=0.00
       KGDOF=0
       DO 145 I=1,4
       DO 145 J=1,6
C
          KGDOF =KGDOF +1
          IF (I.EQ.1) THEN
              LMTI=IDOF(J1,J)
              AT(KGDOF,1)=A11(J,1)
              AT(KGDOF,2)=A11(J,3)
          ELSE IF (I.EQ.2) THEN
              LMTI=IDOF(J2,J)
              AT(KGDOF,3)=A22(J,3)
          ELSE IF (I.EQ.3) THEN
              LMTI=IDOF(J3,J)
              AT(KGDOF,4)=A33(J,3)
          ELSE IF (I.EQ.4) THEN
              LMTI=IDOF(J4,J)
              AT(KGDOF,5)=A44(J,1)
              AT(KGDOF,6)=A44(J,3)
          ENDIF
          LMKG(KGDOF)=LMTI
          IF((AT(KGDOF,1).EQ.0).AND.(AT(KGDOF,2).EQ.0).AND.
     &       (AT(KGDOF,3).EQ.0).AND.(AT(KGDOF,4).EQ.0).AND.
     &       (AT(KGDOF,5).EQ.0).AND.(AT(KGDOF,6).EQ.0)) THEN
             KGDOF=KGDOF-1
          ELSE IF (KGDOF.GT.1) THEN
             DO 149  II=1,KGDOF-1
               IF (LMKG(II).EQ.LMKG(KGDOF)) THEN
                 DO 148 JJ=1,6
                   AT(II,JJ)=AT(II,JJ)+AT(KGDOF,JJ)
148                AT(KGDOF,JJ)=0
                 KGDOF=KGDOF-1
                 GO TO 145
               ENDIF
149          CONTINUE
          ENDIF
C
145   CONTINUE
C
C----- LOCAL GEOMETRIC STIFFNESS MATRIX
       DO 150 I2=1,12
       DO 150 JI=1,12
  150    S(I2,JI)=0.0
       S(1,1)=  1.0/LENGTH
       S(2,2)=  0.5/LENGTH
       S(3,3)=  0.5/LENGTH
       S(4,4)=  0.5/LENGTH
       S(5,5)=  1.0/LENGTH
       S(6,6)=  0.5/LENGTH
       S(1,5)= -1.0/LENGTH
       S(2,6)= -0.5/LENGTH
       S(3,4)= -0.5/LENGTH
       S(5,1)= -1.0/LENGTH
       S(6,2)= -0.5/LENGTH
       S(4,3)= -0.5/LENGTH
C
C-----  CALCULATE SAT
       DO 160 I=1,6
         DO 160 J=1,KGDOF
         SAT2(I,J)=0
         DO 160 K=1,6
  160      SAT2(I,J)=SAT2(I,J)+S(I,K)*AT(J,K)
C
C-----  CALCULATE ASAT AND CONVERT TO HALF STORAGE MODE  BY COLUMNS
       DO 170 J=1,KGDOF
         DO 170 I=J,1,-1
         L=L+1
         STIFF(L)=0
         DO 170 K=1,6
  170      STIFF(L)=STIFF(L)+AT(I,K)*SAT2(K,J)
C
C-----  PRINT GEOMETRIC STIFF DATA FOR DEB_UGGING
       IF ( BTEST(IBUG,7) ) THEN
          CALL WMTRX2( AT    ,KGDOF,6,' AT    ',24)
          CALL WMATRX( S     , 6,6,' KG-S  ')
          CALL WMATRX(SAT2  ,6,KGDOF,'SAT2  ')
          WRITE (6,*) 'KGDOF',KGDOF,'LMKG =',(LMKG(I),I=1,KGDOF)
          CALL LMATRX(STIFF(LKG),MD,KGDOF,300,'WALL GEOMETRIC STIFF.')
       ENDIF
C
C----- RELEASE UNUSED STORAGE
  180 IREL=600-L
C
C     PRINT COLUMN DATA
       IF (FIRST .OR. BTEST(IBUG,7) .OR. BTEST(IBUG,8) ) WRITE (6,192)
  191 WRITE(6,193) NAME,IELNO,MATB,MATS,MATA,
     &             ID(J1),ID(J2),ID(J3),ID(J4),
     &     LENGTH,W,ALPIN,PKG
C
  192 FORMAT(/' ELEMENT 03, R/C SHEAR WALL ELEMENT'// T17,
     &     ' ELEM',' BEND',' SHEAR',' AXIAL',' JOINT',' JOINT',
     &     ' JOINT',' JOINT',    LENGTH    ',    WIDTH   ',
     &     '    ALPHA    ',    PKG     '//,T17,
     &     '    #',' MATL',' MATL',' MATL',' MATL',    #1',    #2',
     &     '    #3',    #4'/)
C
  193 FORMAT(1X,A15,8I6,2X,1P,4G12.4)
      IOPT=IOPT
      RETURN
C---------------------- DETERMINE GEOMETRIC STIFF ----------------------
  200 CONTINUE
C- KG IS DETERMINED FOR A UNIT LOAD WITH IOPT=1.
C- THE ACTUAL LOAD (PGEOM) IS DETERMINED HERE.
C- KG IS MULTIPLIED BY PGEOM WHEN THE TOTAL STIFFNESS IS ASSEMBLED
C- NEGATIVE PGEOM IS COMPRESSION....
C
      AXIAL=.TRUE.
C---------- DETERMINE THE AXIAL LOAD TO BE USED FOR KG...
      IF (KGDATA(1).EQ.1) THEN
            PGEOM =  PKG
      ELSE IF (KGDATA(1).EQ.2) THEN
            PGEOM = -F(3)
      ENDIF
```

```
      IF (BTEST(IBUG,7)) WRITE (6,*) 'AXIAL ',AXIAL,'  PGEOM=',PGEOM
C
      RETURN
C
C*********************** DETERMINE STIFFNESS **************************
  300 CONTINUE
      SB=SB
      SS=SS
      SA=SA
      IF (BTEST(IBUG,7)) WRITE (6,402) IELNO,SB,SS,SA
      IF (BTEST(IBUG,7)) WRITE (6,402) IELNO,SE(ISEB),SE(ISES),SE(ISEA)
C
C------ SB, SS AND SA CONTAIN THE STIFFNESSES, WHICH WERE DETERMINED IN
C       THE LAST CALL TO MA_TLIB
C       IF SB=SE(ISEB) AND SS=SE(ISES) AND SA=SE(ISEA)
C       THEN THE STIFFNESS HAS NOT CHANGED, RETURN
C
      IF (SB.EQ.SE(ISEB).AND.SS.EQ.SE(ISES).AND.SA.EQ.SE(ISEA)) RETURN
C
C------- SET UP THE LOCAL STIFFNESS MATRIX
      SB= SE(ISEB)
      SS= SE(ISES)
      SA= SE(ISEA)
      IF (BTEST(IBUG,7)) WRITE (6,402) IELNO,SB,SS,SA
      SBL= SE(ISEB)/LENGTH
      SSL= SE(ISES)/LENGTH
      SAL= SE(ISEA)/LENGTH
C
C----- CALCULATE SAT
      DO 360 J=1,IELDOF
        SAT(1,J)=SBL*A(J,1)
        SAT(2,J)=SSL*A(J,2)
  360   SAT(3,J)=SAL*A(J,3)
C
C----- CALCULATE ASAT, STORE IN UPPER TRIANGULAR MATRIX
      L=0
      DO 370 J=1,IELDOF
        DO 370 I=J,1,-1
        L=L+1
        DSTIFF(L)=0
        DO 370 K=1,3
  370     DSTIFF(L)=DSTIFF(L)+A(I,K)*SAT(K,J)
      DO 371 J=1,L
  371     STIFF(J)=DSTIFF(J)
C
      IF ( BTEST(IBUG,7) ) THEN
        WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)
        CALL LMATRX(STIFF,MD,IELDOF,(L+1),'WALL STIFFNESS')
      ENDIF
C
      RETURN
C*********************** DETERMINE INCRIMENTAL FORCES *****************
  400 CONTINUE
       IF (PRINT.AND.FIRST) WRITE (6,491) STEPID
       IF (BTEST(IBUG,7)) THEN
         WRITE (6,401) IELNO
  401    FORMAT (1X,30('*'),'SHEAR WALL, ELEMENT#',I6,1X,30('*') )
       ENDIF
C
C------- SET UP THE COMPONENT STIFFNESSES
      SB= SE(ISEB)
      SS= SE(ISES)
      SA= SE(ISEA)
      IF (BTEST(IBUG,7)) WRITE (6,402) IELNO,SB,SS,SA
  402 FORMAT(' IELNO:',I1,' SB:',1P,G12.5,' SS:',G12.5,' SA:',G12.5)
C
C------ LOOP FOR EACH LOAD
      DO 430 LOAD=1,NLOAD
C
C------ TRANSFORM FRAME DISPL INTO LOCAL DISPL
      DO 410 I=1,3
        R(I)=0
        V(I)=0
        DO 410 J=1,IELDOF
          K=LM(J)
          R(I)=R(I)+A(J,I)*DISPL(K,LOAD)
  410     V(I)=V(I)+A(J,I)*VELOC(K)
C
C------ INCRIMENTAL FORCES
      F(1)=SB*R(1)/LENGTH
      F(2)=SS*R(2)/LENGTH
      F(3)=SA*R(3)/LENGTH
C
C------ CHECK STIFFNESS AND CALCULATE UNBALANCED FORCES
C       THIS IS PREFORMED ONLY FOR INELASTIC ANALYSIS
      P(1) =FT(1)+F(1)
      P(2) =FT(2)+F(2)
      P(3) =FT(3)+F(3)
      IF (.NOT. ELSTIC) THEN
C         -- TRANSFER PERMANENT HYSTERESIS DATA TO TEMPROARY STORAGE...
          ISE2=ISE/2
          DO 415 I=1,ISE2
  415       SE(I+ISE2)=SE(I)
C         -- BENDING ---------------
C         -- SET UP TEMPORARY - TOTAL LOADS, DISPL'S AND VELOC
          PL=FT(1)
          D =(RT(1)+R(1))/LENGTH
          DL=RT(1) /LENGTH
          VC=VT(1)+V(1)
          VL=VT(1)
C         -- CALL HYST MODEL TO GET NEW STIFFNESS AND
C            THE TOTAL FORCE P AT DISPL D.
          MOPT=3
          CALL MATLIB
     &   (MOPT,LSTTYP,FALSE ,IREL,FALSE,NAME,
     &   ESEB,PSEB,DUCT(1,1),EXCR(1,1),
     &     P(1),PL,D,DL,VC,VL,LELEM ,LMAT   ,
     &     MATB,MATTYP,SE(ISE2+ISEB),IELNO,DAMAGE)
          IF (SE(ISE2+ISEB).NE.SE(ISEB)) NEWK=.TRUE.
C         -- SHEAR -----------------
C         -- SET UP TEMPORARY - TOTAL LOADS, DISPL'S AND VELOC
          PL=FT(2)
          D =(RT(2)+R(2))/LENGTH
          DL=RT(2) /LENGTH
          VC=VT(2)+V(2)
          VL=VT(2)
C         -- CALL HYST MODEL TO GET NEW STIFFNESS AND
C            THE TOTAL FORCE P AT DISPL D.
          MOPT=3
          CALL MATLIB
     &   (MOPT,LSTTYP,FALSE ,IREL,FALSE,NAME,
     &   ESES,PSES,DUCT(1,2),EXCR(1,2),
     &     P(2),PL,D,DL,VC,VL,LELEM ,LMAT   ,
     &     MATS,MATTYP,SE(ISE2+ISES),IELNO,DAMAGE)
          IF (SE(ISE2+ISES).NE.SE(ISES)) NEWK=.TRUE.
C         -- AXIAL -----------------
C         -- SET UP TEMPORARY - TOTAL LOADS, DISPL'S AND VELOC
          PL=FT(3)
          D =(RT(3)+R(3))/LENGTH
          DL=RT(3) /LENGTH
          VC=VT(3)+V(3)
          VL=VT(3)
C         -- CALL HYST MODEL TO GET NEW STIFFNESS AND
C            THE TOTAL FORCE P AT DISPL D.
          MOPT=3
          CALL MATLIB
     &   (MOPT,LSTTYP,FALSE ,IREL,FALSE,NAME,
     &   ESEA,PSEA,DUCT(1,3),EXCR(1,3),
     &     P(3),PL,D,DL,VC,VL,LELEM ,LMAT   ,
     &     MATA  ,MATTYP,SE(ISE2+ISEA),IELNO,DAMAGE)
          IF (SE(ISE2+ISEA).NE.SE(ISEA)) NEWK=.TRUE.
       ENDIF
C
      IF (ELSTIC) THEN
         ESEB=0.50*(FT(1)+F(1))*(RT(1)+R(1))
         ESES=0.50*(FT(2)+F(2))*(RT(2)+R(2))
         ESEA=0.50*(FT(3)+F(3))*(RT(3)+R(3))
         ESE= ESEB+ESES+ESEA
         PSE=0.0
      ELSE
         ESE=(ESEB+ESES+ESEA)*LENGTH
         PSE=(PSEB+PSES+PSEA)*LENGTH
      ENDIF
C
C------ SAVE MAXIMUM LOADS
      IF (NLOAD.GT.1) THEN
        DO 389 I=1,3
          IF (ABS(F(I)).GT.ABS(FMAX(I)) ) THEN
            FMAX(I  )= F(I)
            FMAX(I+3)= R(I)
          ENDIF
  389   CONTINUE
      ENDIF
C
C------ PRINT DATA
      IF (PRINT) WRITE (6,392) IELNO,LOAD,
     &             ID(J1),ID(J2),ID(J3),ID(J4),
     &                                  (F(I),R(I),I=1,3)
  392 FORMAT (I10,I5,2X,4I5,1P,6G15.6)
C
C------ CALCULATE THE UNBALANCED MEMBER FORCE ON A JOINT.
      IF (LOAD.EQ.1) THEN
            UBM = -( F(1)+FT(1)-P(1) )
            UBS = -( F(2)+FT(2)-P(2) )
            UBA = -( F(3)+FT(3)-P(3) )
            DO 440 I=1,IELDOF
              K=LM(I)
              EFUB(I)= A(I,1)*UBM + A(I,2)*UBS + A(I,3)*UBA
  440         FUB(K)= FUB(K) + EFUB(I)
          IF (BTEST(IBUG,7)) THEN
            WRITE (6,493) (I,F(I),FT(I),P(I),I=1,3)
            WRITE (6,*) 'UB: M,S,A =',UBM,UBS,UBA
            WRITE (6,494) (I,LM(I),EFUB(I),I=1,IELDOF)
          ENDIF
      ENDIF
      F(1)=P(1)-FT(1)
      F(2)=P(2)-FT(2)
      F(3)=P(3)-FT(3)
  430 CONTINUE
C
C------ THIS IS FOR SOL05A ONLY
      IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN
        DO 431 I=1,3
  431     RSAP(I)=F(I)
      ENDIF
C
  491 FORMAT (//'    R/C SHEAR WALL FORCES...',5X,A/
     & '   ELEMENT LOAD',T11,
     & ' JNT-1',' JNT-2',' JNT-3',' JNT-4',
     & '     MOMENT      ','    ROTATION    ',
     & '      SHEAR      ',' SHEAR DISPL.  ',
     & '      AXIAL      ',' AXIAL DISPL.  '/)
  492 FORMAT (I10,I5,2X,4I5,1P,6G15.6)
  493 FORMAT ('  I:',I1,'  F:',1P,G12.4,'  FT:',G12.4,'  P:',G12.4)
  494 FORMAT ('  I:',I5,'  LM(I):',I6,'  FUB(I):',1P,G12.4)
C
      IOPT=IOPT
      RETURN
C*********************** DETERMINE TOTAL FORCES, ENERGIES ECT **********
  500 CONTINUE
      IF( ICOMN .NE. 1 .AND. ICOMN .NE. 2 ) THEN
       IF (PRINT.AND.FIRST) WRITE (6,491) STEPID
C
C------ TOTAL FORCES, DISPLACEMENTS AND VELOCITIES
      DO 510 I=1,3
      FT(I) = FT(I) + F(I)
      RT(I) = RT(I) + R(I)
      VT(I) = VT(I) + V(I)
      IF ( ABS(FT(I)).GT.ABS(FMAX(I)) ) THEN
        FMAX(I  )=FT(I)
        FMAX(I+3)=RT(I)
      ENDIF
  510 CONTINUE
C
C------ PRINT DATA
      IF (PRINT) WRITE (6,492) IELNO,LOAD,
     &             ID(J1),ID(J2),ID(J3),ID(J4),
     &                                  (FT(I),RT(I),I=1,3)
      ELSE IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN
       IF( PRINT .AND. FIRST) WRITE(6,497) STEPID
         DO 513 I=1,3
  513      FT(I)=RSAP(I)
        IF (PRINT) WRITE (6,492) IELNO,LOAD,
     &             ID(J1),ID(J2),ID(J3),ID(J4),
     &                                  (FT(I),I=1,3)
      ENDIF
  497 FORMAT (//'    R/C SHEAR WALL FORCES...',5X,A/
     & '   ELEMENT LOAD',T17,
     & ' JNT-1',' JNT-2',' JNT-3',' JNT-4',
     & '     MOMENT      ','      SHEAR      ',
     & '      AXIAL      '/)
C
C------ TRANSFER TEMPROARY HYSTERESIS DATA TO PERMANENT STORAGE...
      ISE2=ISE/2
      IF (.NOT.ELSTIC) THEN
        DO 520 I=1,ISE2
  520     SE(I)=SE(I+ISE2)
      ENDIF
C
      RETURN
C
C*********************** DETERMINE FIXED END FORCES, AND ADD TO FORCE*
  600 RETURN
C
C------- FIXED END FORCES ARE NOT AVAILABLE FOR R/C SHEAR WALL
C
C********************** WRITE MEMBER FORCES TO OUTPUT FILE **************
  700 CONTINUE
C
C.......... BENDING ENERGY, DUCTILITY AND EXCURSION .....
      IF (ELSTIC) THEN
         ESEB=0.50*(FT(1)+F(1))*(RT(1)+R(1))
         ESES=0.50*(FT(2)+F(2))*(RT(2)+R(2))
         ESEA=0.50*(FT(3)+F(3))*(RT(3)+R(3))
         ESE= ESEB+ESES+ESEA
         PSE=0.0
      ELSE
         CALL MATLIB
     &  ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &  ESEB,PSEB,DUCT(1,1),EXCR(1,1),
     &  FT    ,F    ,RT    ,R    ,0.    ,0.    ,LELEM ,LMAT   ,
     &  MATB  ,MATTYP,SE(ISEB),IELNO,DAMAGE)
C.......... SHEAR ENERGY, DUCTILITY AND EXCURSION .....
         CALL MATLIB
     &  ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &  ESES,PSES,DUCT(1,2),EXCR(1,2),
     &  FT    ,F    ,RT    ,R    ,0.    ,0.    ,LELEM ,LMAT   ,
     &  MATS  ,MATTYP,SE(ISES),IELNO,DAMAGE)
C.......... AXIAL ENERGY, DUCTILITY AND EXCURSION .....
         CALL MATLIB
     &  ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &  ESEA,PSEA,DUCT(1,3),EXCR(1,3),
     &  FT    ,F    ,RT    ,R    ,0.    ,0.    ,LELEM ,LMAT   ,
     &  MATA  ,MATTYP,SE(ISEA),IELNO,DAMAGE)
```

```
C                                                                       ELE08430
          ESE=(ESE8+ESES+ESEA)*LENGTH                                   ELE08440
          PSE=(PSE8+PSES+PSEA)*LENGTH                                   ELE08450
        ENDIF                                                           ELE08460
C                                                                       ELE08470
        IF (PRINT) WRITE (IUNIT,710) ID(J1),ID(J2),ID(J3),ID(J4)        ELE08480
        WRITE (IUNIT,711) (FT(I),RT(I),I=1,3),ESE,PSE                   ELE08490
C     &       ((DUCT(I,J),EXCR(I,J),I=1,3),J=1,3)                       ELE08500
  710 FORMAT (4I5,1P,6E10.3/(8E10.3) )                                  ELE08510
  711 FORMAT (1P,8E10.3)                                                ELE08520
C                                                                       ELE08530
        RETURN                                                          ELE08540
C                                                                       ELE08550
C=================== READ AND PRINT MEMBER FORCES FROM OUTPUT FILE ====ELE08560
  800 CONTINUE                                                          ELE08570
        SX =0                                                           ELE08580
        SXX=0                                                           ELE08590
        ST =0                                                           ELE08600
        SXT=0                                                           ELE08610
        SN =0                                                           ELE08620
        BACKSPACE(IUNIT)                                                ELE08630
        READ (IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT                      ELE08640
        READ (IUNIT,710) NODE1,NODE2,NODE3,NODE4                        ELE08650
        ISTEP=-IWRITE                                                   ELE08660
        AVEMV=0                                                         ELE08670
        ICOUNT=0                                                        ELE08680
C                                                                       ELE08690
CC810 READ (IUNIT,815,END=830)NODE1,NODE2,NODE3,NODE4,(F(I),R(I),I=1,3),ELE08700
CC    &      ESE,PSE,((DUCT(I,J),EXCR(I,J),I=1,3),J=1,3)                ELE08710
CC815 FORMAT (4I5,    6E10.3/(8E10.3) )                                 ELE08720
  810 READ (IUNIT,815,END=830) (F(I),R(I),I=1,3),ESE,PSE                ELE08730
  815 FORMAT (8E10.3)                                                   ELE08740
        ISTEP=ISTEP+IWRITE                                              ELE08750
        T=TO+DT*ISTEP                                                   ELE08760
        IF (ISTEP.EQ.0) WRITE (6,805) NODE1,NODE2,NODE3,NODE4           ELE08770
        IF (MOD(ISTEP,INC).EQ.0)                                        ELE08780
     &  WRITE (6,820) ISTEP,T,(F(I),R(I),I=1,3),ESE,PSE                 ELE08790
        SX  = SX  + F(2)                                                ELE08800
        SXX = SXX + F(2)*F(2)                                           ELE08810
        ST  = ST  + F(1)                                                ELE08820
        SXT = SXT + F(1)*F(2)                                           ELE08830
        SN  = SN  + 1                                                   ELE08840
        IF (F(2).NE.0) THEN                                             ELE08850
            RATMV= F(1)/F(2)                                            ELE08860
C           IF (ABS(RATMV).LT.25) THEN                                  ELE08870
              AVEMV=AVEMV+RATMV                                         ELE08880
              ICOUNT=ICOUNT+1                                           ELE08890
C           ENDIF                                                       ELE08900
        ENDIF                                                           ELE08910
        GO TO 810                                                       ELE08920
C                                                                       ELE08930
  830 CONTINUE                                                          ELE08940
      AVEMV=AVEMV/ MAX(ICOUNT,1)                                        ELE08950
      WRITE (6,831) AVEMV,ICOUNT                                        ELE08960
      IF (ST.NE.0 ) THEN                                                ELE08970
        RM=ST/SX                                                        ELE08980
        ICOUNT=SN                                                       ELE08990
        WRITE (6,833) RM,ICOUNT                                         ELE09000
      ENDIF                                                             ELE09010
      DM = SN*SXX - SX*SX                                               ELE09020
      DB = SX*SX  - SXX*SN                                              ELE09030
      IF (DM.NE.0 .AND. DB.NE.0) THEN                                   ELE09040
        RB = ( SXT*SX - ST*SXX ) / DB                                   ELE09050
        RM = ( SN*SXT - SX*ST  ) / DM                                   ELE09060
        ICOUNT=SN                                                       ELE09070
        WRITE (6,832) RM,RB,ICOUNT                                      ELE09080
      ENDIF                                                             ELE09090
  831 FORMAT(/T10,'AVERAGE MOMENT/SHEAR RATIO=',1P,G12.4,5X,0P,I6,      ELE09100
     & ' POINTS WERE USED, ',                                           ELE09110
     & ' POINTS WITH V=0 ARE EXCLUDED')                                 ELE09120
  832 FORMAT( T10,'REGRESSION MOMENT-SHEAR EQUN:   ',                   ELE09130
     &        ' M =', 1P,G12.4, '* V +', G12.4,5X,                      ELE09140
     & 0P,I6,' POINTS WERE USED')                                       ELE09150
  833 FORMAT( T10,'REGRESSION MOMENT-SHEAR EQUN:   ',                   ELE09160
     &        ' M =', 1P,G12.4, '* V  ', 12X  ,5X,                      ELE09170
     & 0P,I6,' POINTS WERE USED')                                       ELE09180
C                                                                       ELE09190
      WRITE (6,835)                                                     ELE09200
C                                                                       ELE09210
C     WRITE (6,840) 'BENDING COMPONENT',(DUCT(I,1),EXCR(I,1),I=1,3)     ELE09220
C     WRITE (6,840) 'SHEAR COMPONENT'  ,(DUCT(I,2),EXCR(I,2),I=1,3)     ELE09230
C     WRITE (6,840) 'AXIAL COMPONENT'  ,(DUCT(I,3),EXCR(I,3),I=1,3)     ELE09240
C                                                                       ELE09250
  805 FORMAT (//'    R/C SHEAR WALL FORCES...',/                        ELE09260
     & 5X,'NODE1=',I6,5X,'NODE2=',I6,5X,'NODE3=',I6,5X,'NODE4=',I6,//   ELE09270
     & '     STEP     TIME     ',                                       ELE09280
     & '    MOMENT    ','   ROTATION   ',                               ELE09290
     & '     SHEAR    ',' SHEAR DISPL. ',                              ELE09300
     & '     AXIAL    ',' AXIAL DISPL. ',                              ELE09310
     & '   ESE        ','      PSE      '/)                            ELE09320
  820 FORMAT (I6,1P,G12.4,6G14.5,2G12.4)                                ELE09330
  835 FORMAT (/T10,' MAXIMUN DUCTILITIES AND EXCURSION RATIOS'/         ELE09340
     &        T10,'-----------------------------------------------')    ELE09350
  840 FORMAT (/T10,A/                                                   ELE09360
     & T15,'DISPLACEMENT DEFINATION: U1=',1P,G12.4,5X,'E1=',G12.4/      ELE09370
     & T15,'ENERGY DEFINATION #1:    U2=',1P,G12.4,5X,'E2=',G12.4/      ELE09380
     & T15,'ENERGY DEFINATION #2:    U3=',1P,G12.4,5X,'E3=',G12.4)      ELE09390
      RETURN                                                            ELE09400
C                                                                       ELE09410
C==================== DETERMINE THE LENGTH OF THE MATERIAL DATA        ELE09420
  900 CONTINUE                                                          ELE09430
      ISE=0                                                             ELE09440
C------ LOOP FOR EACH MATEIAL --------                                  ELE09450
      DO 910 I=1,3                                                      ELE09460
        IF (I.EQ.1) THEN                                                ELE09470
          ISEB=ISE+1                                                    ELE09480
        ELSE IF (I.EQ.2) THEN                                           ELE09490
          ISES=ISE+1                                                    ELE09500
        ELSE IF (I.EQ.3) THEN                                           ELE09510
          ISEA=ISE+1                                                    ELE09520
        ENDIF                                                           ELE09530
        MATI = RINPUT(I)                                                ELE09540
        CALL MATLIB                                                     ELE09550
     &  ( 0    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                         ELE09560
     &    ESE, PSE,DUCT(1,2),EXCR(1,2),                                ELE09570
     &  FT    ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,       ELE09580
     &  MATI  ,MATTYP,SE,IELNO,DAMAGE)                                  ELE09590
  910 ISE=ISE + LELEM                                                   ELE09600
C                                                                       ELE09610
C------- DOUBLE THE STORAGE, TO ALLOW FOR TEMPROARY VALUES...           ELE09620
      ISE=ISE*2                                                         ELE09630
C                                                                       ELE09640
      RETURN                                                            ELE09650
C                                                                       ELE09660
C==================== DETERMINE THE STRAIN ENERGY                      ELE09670
 1000 CONTINUE                                                          ELE09680
C==================== DUCTILITEIS AND EXCURSION RATIOS                 ELE09690
 1100 CONTINUE                                                          ELE09700
C     ... NOTE ELE03 WAS LAST CALLED TO CALC INCIRMENTAL DISPL,         ELE09710
C              THE ENERGIES ASSOCIATED WITH INCR DISPL ARE STORED IN     ELE09720
C              THE ADDRESSES: ISEB+ISE/2                                ELE09730
C                             ISES+ISE/2                                ELE09740
C                             ISEA+ISE/2                                ELE09750
C.......... BENDING ENERGY, DUCTILITY AND EXCURSION .....              ELE09760
      ISE2=ISE/2                                                        ELE09770
      CALL MATLIB                                                       ELE09780
     &  ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                         ELE09790
     &    ESEB,PSEB,DUCT(1,1),EXCR(1,1),                               ELE09800
     &  FT    ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,       ELE09810
     &  MATB  ,MATTYP,SE(ISEB+ISE2),IELNO,DAMAGE)                      ELE09820
C.......... SHEAR ENERGY, DUCTILITY AND EXCURSION .....                ELE09830
      CALL MATLIB                                                       ELE09840
     &  ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                         ELE09850
     &    ESES,PSES,DUCT(1,2),EXCR(1,2),                               ELE09860
     &  FT    ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,       ELE09870
     &  MATS  ,MATTYP,SE(ISES+ISE2),IELNO,DAMAGE)                      ELE09880
C.......... AXIAL ENERGY, DUCTILITY AND EXCURSION .....                ELE09890
      CALL MATLIB                                                       ELE09900
     &  ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                         ELE09910
     &    ESEA,PSEA,DUCT(1,3),EXCR(1,3),                               ELE09920
     &  FT    ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,       ELE09930
     &  MATA  ,MATTYP,SE(ISEA+ISE2),IELNO,DAMAGE)                      ELE09940
C                                                                       ELE09950
      IF( DUCFAG .EQ. 0 ) THEN                                          ELE09960
      IF( ABS(DUCT(1,1)) .GE. 1 .OR. ABS(DUCT(1,2)) .GE. 1              ELE09970
     &      .OR. ABS(DUCT(1,3)) .GE. 1 ) THEN                           ELE09980
            DUCFAG=ABS(DUCT(1,1))                                       ELE09990
            IF( DUCFAG .LT. ABS(DUCT(1,2)) ) DUCFAG=ABS(DUCT(1,2))      ELE10000
            IF( DUCFAG .LT. ABS(DUCT(1,3)) ) DUCFAG=ABS(DUCT(1,3))      ELE10010
      ENDIF                                                             ELE10020
      ENDIF                                                             ELE10030
C                                                                       ELE10040
      IF (ELSTIC) THEN                                                  ELE10050
        ESEB=0.50*(FT(1)+F(1))*(RT(1)+R(1))                             ELE10060
        ESES=0.50*(FT(2)+F(2))*(RT(2)+R(2))                             ELE10070
        ESEA=0.50*(FT(3)+F(3))*(RT(3)+R(3))                             ELE10080
        ESE= ESEB+ESES+ESEA                                             ELE10090
        PSE=0.0                                                         ELE10100
      ELSE                                                              ELE10110
        ESE=(ESE8+ESES+ESEA)*LENGTH                                     ELE10120
        PSE=(PSE8+PSES+PSEA)*LENGTH                                     ELE10130
      ENDIF                                                             ELE10140
C                                                                       ELE10150
      IF (BTEST(IBUG,7)) THEN                                           ELE10160
          WRITE (6,*) 'ESEB ESES ESEA  ',ESEB,ESES,ESEA                 ELE10170
          WRITE (6,*) 'PSEB PSES PSEA  ',PSEB,PSES,PSEA                 ELE10180
          WRITE (6,*) 'ESE  PSE  LENGTH',ESE ,PSE ,LENGTH               ELE10190
      ENDIF                                                             ELE10200
       IF (PRINT.AND.FIRST) WRITE (6,1191)                              ELE10210
      IF (PRINT) WRITE (6,1192)IELNO,((DUCT(I,J),EXCR(I,J),I=1,3),J=1,3)ELE10220
 1192 FORMAT (1X,I5,10F7.2)                                             ELE10230
 1191 FORMAT (/'    R/C SHEAR WALL DUCTILITY AND EXCURSION RATIOS'/6X,  ELE10240
     &  ' |-------- BENDING COMPONENT ------------|',                    ELE10250
     &  ' |---------- SHEAR COMPONENT ------------|',                    ELE10260
     &  ' |---------- AXIAL COMPONENT ------------|'/,' ELEM#',         ELE10270
C          12345678901234567890123456789012345678901234567890123456789012ELE10280
     &  ' |-- DEFN 1--- --- DEFN 2--- --- DEFN 3--|',                    ELE10290
     &  ' |-- DEFN 1--- --- DEFN 2--- --- DEFN 3--|',                    ELE10300
     &  ' |-- DEFN 1--- --- DEFN 2--- --- DEFN 3--|'/6X,                 ELE10310
     &  '  DUCT.  EXCR.  DUCT.  EXCR.  DUCT.  EXCR.',                   ELE10320
     &  '  DUCT.  EXCR.  DUCT.  EXCR.  DUCT.  EXCR.',                   ELE10330
     &  '  DUCT.  EXCR.  DUCT.  EXCR.  DUCT.  EXCR.')                   ELE10340
C          12345678901234567890123456789012345678901234567890123456789012ELE10350
      RETURN                                                            ELE10360
C                                                                       ELE10370
C                                                                       ELE10380
C======================== WRITE MAXIMUM AND MINIMUM VALUES              ELE10390
 1200 CONTINUE                                                          ELE10400
      IF (FIRST) WRITE(6,491)'MAXIMUM LOADS AND DISPL AT MAXIMUM LOADS 'ELE10410
     &          //' NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'ELE10420
     &           WRITE (6,492) IELNO,LOAD,                              ELE10430
     &           ID(J1),ID(J2),ID(J3),ID(J4),                           ELE10440
     &               (FMAX(I),FMAX(3+I),I=1,3)                          ELE10450
C                                                                       ELE10460
      RETURN                                                            ELE10470
C                                                                       ELE10480
C========================== RESET INTERNAL FORCES                      ELE10490
 1300 CONTINUE                                                          ELE10500
C                                                                       ELE10510
C----- ZERO MATRICIES                                                   ELE10520
      DO 1310 I=1,ISE                                                   ELE10530
 1310   SE(I)=0                                                         ELE10540
      DO 1325 I=1,3                                                     ELE10550
        FMAX(I)=0                                                       ELE10560
        FMAX(I+3)=0                                                     ELE10570
        F (I)=0                                                         ELE10580
        FT(I)=0                                                         ELE10590
        R (I)=0                                                         ELE10600
        RT(I)=0                                                         ELE10610
        V (I)=0                                                         ELE10620
 1325   VT(I)=0                                                         ELE10630
      WRITE (6,1330) IELNO                                              ELE10640
 1330 FORMAT (' SHEAR WALL..... ELEMENT #',I6,                          ELE10650
     &  '  INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO')     ELE10660
C                                                                       ELE10670
        CALL MATLIB                                                     ELE10680
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                         ELE10690
     &    ESEB,PSEB,DUCT(1,1),EXCR(1,1),                               ELE10700
     &  FT    ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,       ELE10710
     &  MATB  ,MATTYP,SE(ISEB),IELNO,DAMAGE)                           ELE10720
        CALL MATLIB                                                     ELE10730
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                         ELE10740
     &    ESES,PSES,DUCT(1,2),EXCR(1,2),                               ELE10750
     &  FT    ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,       ELE10760
     &  MATS  ,MATTYP,SE(ISES),IELNO,DAMAGE)                           ELE10770
        CALL MATLIB                                                     ELE10780
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                         ELE10790
     &    ESEA,PSEA,DUCT(1,3),EXCR(1,3),                               ELE10800
     &  FT    ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,       ELE10810
     &  MATA  ,MATTYP,SE(ISEA),IELNO,DAMAGE)                           ELE10820
C                                                                       ELE10830
      RETURN                                                            ELE10840
C==================== DAMAGE INDEX                                     ELE10850
 1400 IF (ELSTIC) RETURN                                                ELE10860
      FMAX1=FMAX(1)                                                     ELE10870
      FMAX2=FMAX(2)                                                     ELE10880
      FMAX3=FMAX(3)                                                     ELE10890
      FMAX4=FMAX(4)                                                     ELE10900
      FMAX5=FMAX(5)                                                     ELE10910
      FMAX6=FMAX(6)                                                     ELE10920
C.......... BENDING ENERGY, AND DAMAGE INDEX                           ELE10930
      CALL MATLIB                                                       ELE10940
     &  ( 5    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                         ELE10950
     &    ESEB,PSEB,DUCT(1,1),EXCR(1,1),                               ELE10960
     &  FMAX1,0.    ,FMAX4,0.    ,0.    ,0.    ,LELEM ,LMAT  ,       ELE10970
     &  MATB  ,MATTYP,SE(ISEB),IELNO,DAMB)                             ELE10980
C.......... BENDING ENERGY, AND DAMAGE INDEX                           ELE10990
      CALL MATLIB                                                       ELE11000
     &  ( 5    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                         ELE11010
     &    ESES,PSES,DUCT(1,2),EXCR(1,2),                               ELE11020
     &  FMAX2,0.    ,FMAX5,0.    ,0.    ,0.    ,LELEM ,LMAT  ,       ELE11030
     &  MATS  ,MATTYP,SE(ISES),IELNO,DAMS)                             ELE11040
C.......... AXIAL ENERGY, AND DAMAGE INDEX                             ELE11050
      CALL MATLIB                                                       ELE11060
     &  ( 5    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                         ELE11070
     &  ESEA,PSEA,DUCT(1,3),EXCR(1,3),                                  ELE11080
     &  FMAX3,0.    ,FMAX6,0.    ,0.    ,0.    ,LELEM ,LMAT  ,       ELE11090
     &  MATA  ,MATTYP,SE(ISEA),IELNO,DAMA)                             ELE11100
C                                                                       ELE11110
      ESE=(ESEB+ESES+ESEA)*LENGTH                                       ELE11120
      PSE=(PSEB+PSES+PSEA)*LENGTH                                       ELE11130
      DAMAGE=(DAMB*(ESEB+PSEB)+DAMS*(ESES+PSES)+DAMA*(ESEA+PSEA))*LENGTHELE11140
      IF (FIRST) WRITE (6,1420)                                         ELE11150
      WRITE (6,1410)IELNO,DAMB,ESEB,PSEB,DAMS,ESES,PSES,DAMA,ESEA,PSEA  ELE11160
 1410 FORMAT (1X,I5,1P,J(F14.5,2G14.5))                                 ELE11170
 1420 FORMAT (/5X,                                                      ELE11180
     &  ' |-------- BENDING COMPONENT ------------|',                    ELE11190
     &  ' |---------- SHEAR COMPONENT ------------|',                    ELE11200
     &  ' |---------- AXIAL COMPONENT ------------|'/,' ELEM#',         ELE11210
C          12345678901234567890123456789012345678901234567890123456789012ELE11220
     &  ' |-- DAMAGE -- ---- ESE ---- ---- PSE ---|',                    ELE11230
     &  ' |-- DAMAGE -- ---- ESE ---- ---- PSE ---|',                    ELE11240
     &  ' |-- DAMAGE -- ---- ESE ---- ---- PSE ---|')                    ELE11250
      RETURN                                                            ELE11260
```

```
      END                                                               ELE11270
C-----------------------------------------------------------------------CELE00010
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    ELE00020
C           END DEBUG                                                   ELE00030
C-----------------------------------------------------------------------CELE00040
      SUBROUTINE ELE08                                                  ELE00050
     &  (IOPT       ,IUNIT      ,NEWK       ,FIRST      ,PRINT      ,   ELE00060
     &   NDOF       ,NLOAD      ,MAXNOD     ,NNODE      ,NCOS       ,   ELE00070
     &   ID         ,IDOF       ,COORD      ,COSINE     ,JTCOS      ,   ELE00080
     &   CONST      ,DISPL      ,VELOC      ,FORCE      ,FUB        ,   ELE00090
     &   IREL       ,BUG        ,IBUG       ,ACCEL      ,KGDATA     ,   ELE00100
     &   GRAV       ,PGEOM      ,ELSTIC     ,NAME       ,STEPID     ,   ELE00110
     &   IELNO      ,RINPUT     ,EESE       ,EPSE       ,INC        ,   ELE00120
     &   DAMAGE     ,DUCFAG     ,           ,           ,           ,   ELE00130
     &   IELTYP     ,IELDOF     ,NODEI      ,NODEJ      ,JOINTI     ,   ELE00140
     &   JOINTJ     ,R          ,RT         ,F          ,FT         ,   ELE00150
     &   V          ,VT         ,ISE        ,H          ,LENGTH     ,   ELE00160
     &   MAT        ,A          ,LM         ,KTYPE      ,FMAX       ,   ELE00170
     &   SE         ,STIFF      )                                       ELE00180
C                                                                       ELE00190
      IMPLICIT REAL(A-H,O-Z)                                            ELE00200
C                                                                       ELE00210
      COMMON /RSA/ICOMN, RSAF(12)                                       ELE00220
      COMMON /IDSTEP/ISTEP                                              ELE00230
      REAL ICOMN                                                        ELE00240
C                                                                       ELE00250
      LOGICAL ERR,FIRST,PRINT,BUG,FALSE,NEWK,ELSTIC                     ELE00260
      LOGICAL BTEST                                                     ELE00270
      CHARACTER*80 NAME                                                 ELE00280
      CHARACTER*(*) STEPID                                              ELE00290
      CHARACTER*9  TYPE(1)                                              ELE00300
      DIMENSION  IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS)       ELE00310
      DIMENSION CONST(MAXNOD,6),   ID(MAXNOD)  ,JTCOS(MAXNOD)          ELE00320
      DIMENSION SAT(2,12),STIFF(78)                                     ELE00330
      DIMENSION A(12,2),S(2,2),LM(12),FMAX(2)                           ELE00340
      DIMENSION CTS( 3, 3),CTE( 3, 3),VX(3),VT(3),V3(3)                 ELE00350
      DIMENSION BS( 3, 3),BE( 3, 3),SE(ISE)                             ELE00360
      DIMENSION DISPL(NDOF,NLOAD),FUB(NDOF)                             ELE00370
      DIMENSION VELOC(NDOF)                                             ELE00380
      DIMENSION FORCE(NDOF,NLOAD)                                       ELE00390
      DIMENSION RINPUT(100),WORK(6),DUCT(3),EXCR(6)                     ELE00400
      REAL LENGTH                                                       ELE00410
      REAL IX,IY,IZ,KIIZ,KJJZ,KIJZ,KIIT,KJJT,KIJT,KTI,TTJ              ELE00420
      INTEGER REL,RELT                                                  ELE00430
      DIMENSION MD(13)                                                  ELE00440
      DATA MD/1,2,4,7,11,16,22,29,37,46,56,67,79/                       ELE00450
      DATA TYPE/'     AXIAL'/                                           ELE00460
C                                                                       ELE00470
C-----------------------------------------------------------------------ELE00480
C VARIABLES:                                                            ELE00490
C-----------------------------------------------------------------------ELE00500
C---------- GLOBAL VARIABLES                                           ELE00510
C IOPT   = 1, INITIALISE BEAM COLUMN ELEMENT                            ELE00520
C        = 2, CALCULATE GEOMETRIC STIFFNESS                             ELE00530
C        = 3, FORM STIFFNESS                                            ELE00540
C        = 4, CALCULATE INCREMENTAL FORCES                              ELE00550
C        = 5, CALCULATE TOTAL FORCES AND ENERGIES                       ELE00560
C IUNIT  = OUTPUT FILE UNIT # FOR PRINTING OUTPUT                       ELE00570
C ERR    = ERR OR FLAG, IF AN ERR OR HAS OCCURED, ERR=.TRUE.            ELE00580
C FIRST  = FLAG, FIRST=.TRUE., PRINT HEADERS                            ELE00590
C PRINT  = FLAG, PRINT=.TRUE., PRINT DATA                               ELE00600
C NDOF   = # OF GLOBAL DOF                                              ELE00610
C NLOAD  = # OF LOAD COMBINATIONS                                       ELE00620
C MAXNOD = # ROW DIMENSION OF NODE ARRAY                                ELE00630
C NNODE  = # OF NODES                                                   ELE00640
C ID     = ARRAY OF EXTERNAL NODE NUMBERS                               ELE00650
C IDOF   = ARRAY OF DEGREES OF FREEDOM                                  ELE00660
C COORD  = ARRAY OF COORDINATES                                         ELE00670
C COSINE = ARRAY OF DIRECTION COSINES OF NODES                          ELE00680
C CONST  = ARRAY OF CONSTRAINT TRANSFORMATIONS                          ELE00690
C DISPL  = GLOBAL DISPLACEMENT MATRIX                                   ELE00700
C Z      = REAL GLOBAL STORTAG VECTOR                                   ELE00710
C NZ     = INTEGER GLOBAL STORTAG VECTOR                                ELE00720
C IZMAT  = LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR           ELE00730
C NAME   = ELEMENT NAME                                                 ELE00740
C IELNO  = ELEMENT NUMBER                                               ELE00750
C RINPUT = INPUT DATA                                                   ELE00760
C STIFF  = OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)             ELE00770
C LM     = OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX              ELE00780
C---------- ELEMENT VARIABLES                                          ELE00790
C H,Q,AI,AJ,C,BI,BJ,S22,S26,S212,S33,S35,S311,D                         ELE00800
C        = INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX         ELE00810
C PKG    = INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX...         ELE00820
C R      = INCRIMENTAL DISPLACEMENTS                                    ELE00830
C RT     = TOTAL DISPLACEMENTS                                          ELE00840
C F      = INCRIMENTAL FORCES                                           ELE00850
C FT     = TOTAL FORCES                                                 ELE00860
C LM     = LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                     ELE00870
C CTS    = LOCAL TO GLOBAL ROTATION MATRIX START                        ELE00880
C CTE    = LOCAL TO GLOBAL ROTATION MATRIX END                          ELE00890
C BS     = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX STARTELE00900
C BE     = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END  ELE00910
C STIFF  = ELEMENT STIFFNESS                                            ELE00920
C IELTYP = ELEMENT TYPE = 10                                            ELE00930
C NODEI  = EXTERNAL JOINT NUMBER, END I                                 ELE00940
C NODEJ  = EXTERNAL JOINT NUMBER, END J                                 ELE00950
C JOINTI = INTERNAL JOINT NUMBER, END I                                 ELE00960
C JOINTJ = INTERNAL JOINT NUMBER, END J                                 ELE00970
C---------- TEMPROARY VARIABLES                                        ELE00980
C VX     = VECTOR DEFINING THE ELEMENT X AXIS                           ELE00990
C VT     = VECTOR DEFINING THE ELEMENT Y AXIS                           ELE01000
C COSLOC = LOCAL COSINE MATRIX                                          ELE01010
C A      = GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                  ELE01020
C S      = ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART   ELE01030
C SAT    = PRODUCT OF S*A'                                              ELE01040
C ASAT   = PRODUCT OF A*SAT                                             ELE01050
C-----------------------------------------------------------------------ELE01060
C                                                                       ELE01070
C                                                                       ELE01080
C------- CHOOSE OPTION                                                  ELE01090
      IF (IOPT.LT.1 .OR.IOPT.GT.14)                                     ELE01100
     & WRITE (6,*) 'INVALID OPTION IN ELE-08, IOPT=',IOPT               ELE01110
C                                                                       ELE01120
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,        ELE01130
     &                                          1300,1400),IOPT         ELE01140
C                                                                       ELE01150
  100 CONTINUE                                                          ELE01160
C                                                                       ELE01170
C----- INTERPERTATE INPUT DATA                                          ELE01180
      IELTYP= 8                                                         ELE01190
      MAT    = RINPUT( 1)                                               ELE01200
      NODEI  = RINPUT( 2)                                               ELE01210
      NODEJ  = RINPUT( 3)                                               ELE01220
      KTYPE  = RINPUT( 4)                                               ELE01230
      SLK    = RINPUT( 5)                                               ELE01240
      VT(1)  = RINPUT( 6)                                               ELE01250
      VT(2)  = RINPUT( 7)                                               ELE01260
      VT(3)  = RINPUT( 8)                                               ELE01270
      XS     = RINPUT( 9)                                               ELE01280
      XE     = RINPUT(10)                                               ELE01290
C                                                                       ELE01300
C----- GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S            ELE01310
      JOINTI=IQUICK(NODEI,ID,NNODE)                                     ELE01320
      JOINTJ=IQUICK(NODEJ,ID,NNODE)                                     ELE01330
C     WRITE (6,*) 'IDs',ID                                              ELE01340
C                                                                       ELE01350
      IF (KTYPE.LT.1 .OR. KTYPE.GT.1) THEN                              ELE01360
         WRITE(6,102) IELNO,NODEI,NODEJ,KTYPE                           ELE01370
  102    FORMAT(1X,20('*'),'ER',/ROR - INVALID KTYPE',/                 ELE01380
     &    5X,'IELNO=',I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,               ELE01390
     &    5X,'KTYPE=',I5,/                                              ELE01400
     &    5X,'REVISE INPUT  FOR KTYPE EQ 1 ..............')             ELE01410
      ENDIF                                                             ELE01420
      ENDIF                                                             ELE01430
C                                                                       ELE01440
C----- PRINT DATA FOR PLOTTING                                          ELE01450
      IF (BTEST(IBUG,2)) THEN                                           ELE01460
       WRITE(7,10) NODEJ, NODEI,                                        ELE01470
     &            (COORD(JOINTJ,I),I=1,3),(COORD(JOINTI,J),J=1,3)     ELE01480
   10  FORMAT (15,1X,15,1P,6G12.4)                                      ELE01490
      ENDIF                                                             ELE01500
C                                                                       ELE01510
C-----------------------------------------------------------------------ELE01520
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                            ELE01530
C-----------------------------------------------------------------------ELE01540
C----- DEFINE VECTOR X, LENGTH                                          ELE01550
      VX(1)=COORD(JOINTJ,1)-COORD(JOINTI,1)                             ELE01560
      VX(2)=COORD(JOINTJ,2)-COORD(JOINTI,2)                             ELE01570
      VX(3)=COORD(JOINTJ,3)-COORD(JOINTI,3)                             ELE01580
      LENGTH=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)-XS-XE                     ELE01590
      SE(5)=LENGTH                                                      ELE01600
      SE(2)=SLK                                                         ELE01610
      IF (LENGTH.LE.0) THEN                                             ELE01620
         WRITE(6,101) IELNO,JOINTI,JOINTJ,LENGTH                        ELE01630
  101    FORMAT(1X,20('*'),'ER','ROR - LENGTH IS LE 0',/                ELE01640
     &    5X,'IELNO=',I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,               ELE01650
     &    5X,'LENGTH',1P,G15.6/                                         ELE01660
     &    5X,'REVISE INPUT.............................')              ELE01670
         LENGTH=1.0                                                     ELE01680
      ENDIF                                                             ELE01690
      IF ( BTEST(IBUG,7) ) THEN                                         ELE01700
         WRITE (6,*) 'JOINTS',JOINTI,JOINTJ                             ELE01710
         WRITE (6,*) 'VX',VX                                            ELE01720
         WRITE (6,*) 'VT',VT                                            ELE01730
         WRITE (6,*) 'LENGTH  ',LENGTH                                  ELE01740
      ENDIF                                                             ELE01750
C----- GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES CT AND B            ELE01760
      ICS=JTCOS(JOINTI)                                                 ELE01770
      ICE=JTCOS(JOINTJ)                                                 ELE01780
      CALL ROTXYZ(VX,VT,COSINE(1,1,ICS),CTS, XS ,0. ,0. ,BS,            ELE01790
     &      CONST(JOINTI,1),MAXNOD)                                     ELE01800
      CALL ROTXYZ(VX,VT,COSINE(1,1,ICE),CTE,-XE ,0. ,0. ,BE,            ELE01810
     &      CONST(JOINTJ,1),MAXNOD)                                     ELE01820
      IF ( BTEST(IBUG,7) ) THEN                                         ELE01830
           CALL WMATRX( CTS  , 3, 3,'CTS   ')                           ELE01840
           CALL WMATRX( CTE  , 3, 3,'CTE   ')                           ELE01850
      ENDIF                                                             ELE01860
C                                                                       ELE01870
C----- GET GLOBAL CONSTRAINT MATRIX BS, BE                              ELE01880
C     BS(1,1)=CONST(JOINTI,1)*CTS(2,1)+CONST(JOINTI,2)*CTS(3,1)+BS(1,1) ELE01890
C     BS(1,2)=CONST(JOINTI,1)*CTS(2,2)+CONST(JOINTI,2)*CTS(3,2)+BS(1,2) ELE01900
C     BS(1,3)=CONST(JOINTI,1)*CTS(2,3)+CONST(JOINTI,2)*CTS(3,3)+BS(1,3) ELE01910
C     BS(2,1)=CONST(JOINTI,3)*CTS(1,1)+CONST(JOINTI,4)*CTS(3,1)+BS(2,1) ELE01920
C     BS(2,2)=CONST(JOINTI,3)*CTS(1,2)+CONST(JOINTI,4)*CTS(3,2)+BS(2,2) ELE01930
C     BS(2,3)=CONST(JOINTI,3)*CTS(1,3)+CONST(JOINTI,4)*CTS(3,3)+BS(2,3) ELE01940
C     BS(3,1)=CONST(JOINTI,5)*CTS(1,1)+CONST(JOINTI,6)*CTS(2,1)+BS(3,1) ELE01950
C     BS(3,2)=CONST(JOINTI,5)*CTS(1,2)+CONST(JOINTI,6)*CTS(2,2)+BS(3,2) ELE01960
C     BS(3,3)=CONST(JOINTI,5)*CTS(1,3)+CONST(JOINTI,6)*CTS(2,3)+BS(3,3) ELE01970
C                                                                       ELE01980
C     BE(1,1)=CONST(JOINTJ,1)*CTE(2,1)+CONST(JOINTJ,2)*CTE(3,1)+BE(1,1) ELE01990
C     BE(1,2)=CONST(JOINTJ,1)*CTE(2,2)+CONST(JOINTJ,2)*CTE(3,2)+BE(1,2) ELE02000
C     BE(1,3)=CONST(JOINTJ,1)*CTE(2,3)+CONST(JOINTJ,2)*CTE(3,3)+BE(1,3) ELE02010
C     BE(2,1)=CONST(JOINTJ,3)*CTE(1,1)+CONST(JOINTJ,4)*CTE(3,1)+BE(2,1) ELE02020
C     BE(2,2)=CONST(JOINTJ,3)*CTE(1,2)+CONST(JOINTJ,4)*CTE(3,2)+BE(2,2) ELE02030
C     BE(2,3)=CONST(JOINTJ,3)*CTE(1,3)+CONST(JOINTJ,4)*CTE(3,3)+BE(2,3) ELE02040
C     BE(3,1)=CONST(JOINTJ,5)*CTE(1,1)+CONST(JOINTJ,6)*CTE(2,1)+BE(3,1) ELE02050
C     BE(3,2)=CONST(JOINTJ,5)*CTE(1,2)+CONST(JOINTJ,6)*CTE(2,2)+BE(3,2) ELE02060
C     BE(3,3)=CONST(JOINTJ,5)*CTE(1,3)+CONST(JOINTJ,6)*CTE(2,3)+BE(3,3) ELE02070
C                                                                       ELE02080
      IF ( BTEST(IBUG,7) ) THEN                                         ELE02090
           CALL WMATRX(BS, 3, 3,'BS   ')                                ELE02100
           CALL WMATRX(BE, 3, 3,'BE   ')                                ELE02110
      ENDIF                                                             ELE02120
C                                                                       ELE02130
C----- ZERO MATRICIES                                                   ELE02140
          F =0                                                          ELE02150
          FT=0                                                          ELE02160
          R =0                                                          ELE02170
          RT=0                                                          ELE02180
          V =0                                                          ELE02190
          VT=0                                                          ELE02200
          FMAX(1)=0                                                     ELE02210
          FMAX(2)=0                                                     ELE02220
C                                                                       ELE02230
C----- ASSEMBLE TRANSFORMATION MATRIX A                                 ELE02240
C  NOTE: A IS DIMMENSIONED AS A 12X2 MATRIX, BUT ONLY THE FIRST IELDOF  ELE02250
C        ROW'S CONTAIN INFORMATION. THE BALANCE OF THE MATRIX IS        ELE02260
C        UNDEFINED.  THE GLOBAL DOF COORESPONDING TO THE ROW'S OF A     ELE02270
C        ARE STORED IN VECTOR LM.  AGAIN, ONLY THE FIRST IELDOF ROW'S   ELE02280
C        OF LM ARE DEFINED.                                             ELE02290
       IELDOF=0                                                         ELE02300
       DO 50 I=1,12                                                     ELE02310
C                                                                       ELE02320
C              REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF...     ELE02330
          J=I                                                           ELE02340
          IF (I.GT.6) J= I-6                                            ELE02350
          IF ( IDOF(JOINTI,J).EQ.IDOF(JOINTJ,J) ) GO TO 50              ELE02360
          IF (I.LE.6) THEN                                              ELE02370
              LMTI=IDOF(JOINTI,I)                                       ELE02380
          ELSE                                                          ELE02390
              LMTI=IDOF(JOINTJ,J)                                       ELE02400
          ENDIF                                                         ELE02410
C                                                                       ELE02420
          A1=0                                                          ELE02430
          A2=0                                                          ELE02440
          IF (I.LE.3) THEN                                              ELE02450
               IF (KTYPE.LE.3) A1=CTS(I,KTYPE)                          ELE02460
          ELSE IF (I.LE.6) THEN                                         ELE02470
               IF (KTYPE.LE.3) THEN                                     ELE02480
                   A1=BS(I-3,KTYPE)                                     ELE02490
               ELSE                                                     ELE02500
                   A1=CTS(I-3,KTYPE-3)                                  ELE02510
               ENDIF                                                    ELE02520
          ELSE IF (I.LE.9) THEN                                         ELE02530
               IF (KTYPE.LE.3) A2=CTE(I-6,KTYPE)                        ELE02540
          ELSE                                                          ELE02550
               IF (KTYPE.LE.3) THEN                                     ELE02560
                   A2=BE(I-9,KTYPE)                                     ELE02570
               ELSE                                                     ELE02580
                   A2=CTE(I-9,KTYPE-3)                                  ELE02590
               ENDIF                                                    ELE02600
          ENDIF                                                         ELE02610
          IF (A1.EQ.0 .AND. A2.EQ.0) GO TO 50                           ELE02620
          IELDOF=IELDOF+1                                               ELE02630
          A(IELDOF,1)=A1                                                ELE02640
          A(IELDOF,2)=A2                                                ELE02650
          LM(IELDOF)=LMTI                                               ELE02660
C                                                                       ELE02670
   50    CONTINUE                                                       ELE02680
      IF ( BTEST(IBUG,7) ) CALL WMTRX2( A    ,IELDOF,2,' A    ',12)     ELE02690
C                                                                       ELE02700
C----- GET GOEL STRUT STIFFNESS                                         ELE02710
        FALSE=.FALSE.                                                   ELE02720
        LSTTYP=0                                                        ELE02730
        CALL MATLIB                                                     ELE02740
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE, DUCT,EXCR,    ELE02750
     &    FT   ,F      ,RT    ,R    ,0.    ,0.    ,LELEM ,LMAT  ,       ELE02760
     &    MAT  ,MATTYP,SE,IELNO,DAMAGE)                                 ELE02770
C                                                                       ELE02780
C     WRITE (6,*) 'AFTER MA','TLIB IN ELE08 '                           ELE02790
  111 FORMAT (OP,' ZMATRX(',I6,')=',I15,1P,G20.10)                      ELE02800
C-------- H IS THE STRUT STIFFNESS                                      ELE02810
C                                                                       ELE02820
C     WRITE(6,*)'AFTER MATLIB-2, STIFF=',SE(1)                          ELE02830
```

```
      H = SE(1)                                                         ELE02840
      HH= SE(1)                                                         ELE02850
      S(1,1)= HH                                                        ELE02860
      S(1,2)=-HH                                                        ELE02870
      S(2,1)=-HH                                                        ELE02880
      S(2,2)= HH                                                        ELE02890
C                                                                       ELE02900
C----- CALCULATE SAT                                                   ELE02910
      DO 60 I=1,2                                                       ELE02920
        DO 60 J=1,IELDOF                                                ELE02930
        SAT(I,J)=0                                                      ELE02940
          DO 60 K=1,2                                                   ELE02950
   60       SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)                            ELE02960
C                                                                       ELE02970
C----- CALCULATE ASAT AND CONVERT TO HALF STORAGE MODE  BY COLUMNS      ELE02980
C                                                                       ELE02990
C          1  3  6 10 15 21 28 36 45 55 66 78  NUMBER INDICATES        ELE03000
C             2  5  9 14 20 27 35 44 54 65 77  LOCATION OF DATA        ELE03010
C                4  8 13 19 26 34 43 53 64 76  IN ARRAY STIFF          ELE03020
C                   7 12 18 25 33 42 52 63 75                          ELE03030
C                  11 17 24 32 41 51 62 74                             ELE03040
C                     16 23 31 40 50 61 73                             ELE03050
C                        22 30 39 49 60 72                             ELE03060
C                           29 38 48 59 71                             ELE03070
C                              37 47 58 70                             ELE03080
C                                 46 57 69                             ELE03090
C                                    56 68                             ELE03100
C                                       67                             ELE03110
      L=0                                                               ELE03120
      DO 70 J=1,IELDOF                                                  ELE03130
        DO 70 I=J,1,-1                                                  ELE03140
        L=L+1                                                           ELE03150
        STIFF(L)=0                                                      ELE03160
          DO 70 K=1,2                                                   ELE03170
   70       STIFF(L)=STIFF(L)+A(I,K)*SAT(K,J)                          ELE03180
C                                                                       ELE03190
      IF ( BTEST(IBUG,7) ) THEN                                         ELE03200
        WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)           ELE03210
        CALL LMATRX(STIFF,MD,IELDOF,(L+1),'STRUT STIFFNESS')            ELE03220
      ENDIF                                                             ELE03230
C                                                                       ELE03240
C----- RELEASE UNUSED STORAGE                                          ELE03250
  180 IREL=78-L                                                         ELE03260
C                                                                       ELE03270
C     PRINT COLUMN DATA                                                 ELE03280
      IF (FIRST .OR. BTEST(IBUG,7) .OR. BTEST(IBUG,8) ) WRITE (6,192)   ELE03290
  191 WRITE(6,193) NAME,IELNO,MAT,NODEI,NODEJ,TYPE(KTYPE),LENGTH,       ELE03300
     &    VT(1),VT(2),VT(3),XS,-XE                                     ELE03310
C                                                                       ELE03320
  192 FORMAT(//' ELEMENT 08, ONE (LOCAL) DOF STRUT ELEMENT'//          ELE03330
     &  T22,'#  MATL START   END',2X,'--TYPE---',6X,'LENGTH',3X,        ELE03340
     & 11('-'),'  T-AXIS  ',                                            ELE03350
     & 12('-'),'  START DIST    END DIST')                              ELE03360
  193 FORMAT(1X,A15,A16,2X,A9,3X,1P,G12.4,0P,                           ELE03370
     &       F9.5,' I',SP,F9.5,' J',F9.5,' K',1P,SS,3G12.4)           ELE03380
      IOPT=IOPT                                                         ELE03390
      RETURN                                                            ELE03400
C------------------------ DETERMINE GEOMETRIC STIFF ------------------- ELE03410
  200 RETURN                                                            ELE03420
C     THE GEOMETRIC STIFFNESS MATRIX IS NOT AVIALABLE FOR THIS ELEMENT  ELE03430
C                                                                       ELE03440
C------------------------ DETERMINE STIFFNESS ------------------------- ELE03450
  300 CONTINUE                                                          ELE03460
C                                                                       ELE03470
C------ H CONTAINS THE STRUT STIFFNESS, WHICH WAS DETERMINED IN THE     ELE03480
C       LAST CALL TO MAT_LIB                                            ELE03490
C       IF H=SE(1) THEN THE STIFFNESS HAS NOT CHANGED, RETURN           ELE03500
C                                                                       ELE03510
      IF (H.EQ.SE(1)) RETURN                                            ELE03520
C                                                                       ELE03530
      H = SE(1)                                                         ELE03540
      HH= SE(1)                                                         ELE03550
      S(1,1)= HH                                                        ELE03560
      S(1,2)=-HH                                                        ELE03570
      S(2,1)=-HH                                                        ELE03580
      S(2,2)= HH                                                        ELE03590
C                                                                       ELE03600
C----- CALCULATE SAT                                                   ELE03610
      DO 360 I=1,2                                                      ELE03620
        DO 360 J=1,IELDOF                                               ELE03630
        SAT(I,J)=0                                                      ELE03640
          DO 360 K=1,2                                                  ELE03650
  360       SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)                           ELE03660
C                                                                       ELE03670
C----- CALCULATE ASAT, STORE IN UPPER TRIANGULAR MATRIX                ELE03680
      L=0                                                               ELE03690
      DO 370 J=1,IELDOF                                                 ELE03700
        DO 370 I=J,1,-1                                                 ELE03710
        L=L+1                                                           ELE03720
        STIFF(L)=0                                                      ELE03730
          DO 370 K=1,2                                                  ELE03740
  370       STIFF(L)=STIFF(L)+A(I,K)*SAT(K,J)                         ELE03750
C                                                                       ELE03760
      IF ( BTEST(IBUG,7) ) THEN                                         ELE03770
        WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)           ELE03780
        CALL LMATRX(STIFF,MD,IELDOF,(L+1),'STRUT STIFFNESS')            ELE03790
      ENDIF                                                             ELE03800
C                                                                       ELE03810
      IOPT=IOPT                                                         ELE03820
      RETURN                                                            ELE03830
C------------------------ DETERMINE INCRIMENTAL FORCES ---------------- ELE03840
  400 CONTINUE                                                          ELE03850
      IF (PRINT.AND.FIRST) WRITE (6,491) STEPID                         ELE03860
C                                                                       ELE03870
C------ LOOP FOR EACH LOAD                                             ELE03880
      DO 430 LOAD=1,NLOAD                                               ELE03890
C                                                                       ELE03900
C------ TRANSFORM FRAME DISPL INTO LOCAL DISPL                          ELE03910
C... LOCAL DISPL ARE AT THE END END OF THE STRUT, THUS                  ELE03920
C... POSITIVE IS TENSION AND NEGATIVE IS COMPRESSION                    ELE03930
        R=0                                                             ELE03940
        V=0                                                             ELE03950
        DO 410 I=1,IELDOF                                               ELE03960
          J=LM(I)                                                       ELE03970
            AI1= A(I,1)                                                 ELE03980
            AI2= A(I,2)                                                 ELE03990
            DISP= DISPL(J,LOAD)                                         ELE04000
            VEL= VELOC(J)                                               ELE04010
                                                                        ELE04020
            R= R + (A(I,2)-A(I,1)) * DISPL(J,LOAD)                      ELE04030
  410       V= V + (A(I,2)-A(I,1)) * VELOC(J)                         ELE04040
C                                                                       ELE04050
C------ INCRIMENTAL FORCES                                             ELE04060
      F= H*R                                                            ELE04070
C                                                                       ELE04080
C------ CHECK STIFFNESS AND CALCULATE UNBALANCED FORCES                 ELE04090
C       THIS IS PREFORMED ONLY FOR INELASTIC ANALYSIS                   ELE04100
      P =FT+F                                                           ELE04110
      IF (.NOT. ELSTIC) THEN                                            ELE04120
C           -- SET UP TEMPORARY - TOTAL LOADS, DISPL'S AND VELOC        ELE04130
            PL=FT                                                       ELE04140
            D =(RT+R)                                                   ELE04150
            DL=RT                                                       ELE04160
            VC=VT+V                                                     ELE04170
            VL=VT                                                       ELE04180
C           -- TRANSFER PERMANENT HYSTERESIS DATA TO TEMPROARY STORAGE...ELE04190
            ISE2=ISE/2                                                  ELE04200
            DO 415 I=1,ISE2                                             ELE04210
  415         SE(I+ISE2)=SE(I)                                          ELE04220
C           -- CALL HIST MODEL TO GET NEW STIFFNESS AND                 ELE04230
C              THE TOTAL FORCE P AT DISPL D.                            ELE04240
            MOPT=3                                                      ELE04250
            CALL MATLIB                                                 ELE04260
     &     (MOPT ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,  DUCT,EXCR,  ELE04270
     &      P    ,PL    ,D    ,DL    ,VC    ,VL    ,LELEM ,LMAT  ,      ELE04280
     &      MAT  ,MATTYP,SE(ISE2+1),IELNO,DAMAGE)                      ELE04290
            IF (SE(ISE2+1).NE.SE(1)) NEWK=.TRUE.                        ELE04300
                                                                        ELE04310
C     WRITE(6,*)'AFTER MATLIB-2, STIF=',SE(ISE2+1)                      ELE04320
      ENDIF                                                             ELE04330
C                                                                       ELE04340
C------ SAVE PEAK VALUES FOR MULTIPLE LOAD CASES(FOR STATICS CASE)      ELE04350
      IF (NLOAD.GT.1 .AND.  ABS(F).GT.ABS(FMAX(1)) ) THEN               ELE04360
        FMAX(1)=F                                                       ELE04370
        FMAX(2)=R                                                       ELE04380
      ENDIF                                                             ELE04390
C                                                                       ELE04400
C------ PRINT DATA                                                     ELE04410
      HH= H                                                             ELE04420
      IF (PRINT) WRITE (6,492) IELNO,LOAD,TYPE(KTYPE),NODEI,NODEJ,F,R,HHELE04430
C                                                                       ELE04440
  430 CONTINUE                                                          ELE04450
C                                                                       ELE04460
C------ THIS IS FOR SOL05A USE ONLY                                    ELE04470
      IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                         ELE04480
        RSAF(1)=F                                                       ELE04490
      ENDIF                                                             ELE04500
C                                                                       ELE04510
C------ CALCULATE THE UNBALANCED MEMBER FORCE ON A JOINT.              ELE04520
            UBAL= F+FT-P                                                ELE04530
            F   = F                                                     ELE04540
            FT  = FT                                                    ELE04550
            P   = P                                                     ELE04560
            IF (BTEST(IBUG,7)) WRITE (6,493) F,FT,P                     ELE04570
            DO 440 I=1,IELDOF                                           ELE04580
               J=LM(I)                                                  ELE04590
               AI1= A(I,1)                                              ELE04600
               AI2= A(I,2)                                              ELE04610
               FUB(J)= FUB(J) + (A(I,1)-A(I,2)) * UBAL                  ELE04620
C              WRITE(60,*)UBAL,FUB(J), A(I,1),A(I,2)                    ELE04630
C              IF (BTEST(IBUG,7))                                       ELE04640
               IF (BTEST(IBUG,7))                                       ELE04650
     &         WRITE (6,494) (II,LM(II),FUB(II),II=1,IELDOF)            ELE04660
  440       CONTINUE                                                    ELE04670
C                                                                       ELE04680
  491 FORMAT (//'       GOEL FORCES...',5X,A/,                          ELE04690
     & '   ELEMENT LOAD',T17,'      TYPE',' NODEI',' NODEJ',           ELE04700
     & '      FORCE     ',' DISPLACMENT  ',' STIFFNESS   ')            ELE04710
  492 FORMAT (I10,I5,2X,A9,2I6,1P,3G15.6)                               ELE04720
  493 FORMAT ('  F:',1P,G16.8,'  FT:',G16.8,'  P:',G16.8)               ELE04730
  494 FORMAT ('  I:',I5,'  LM(I):',I6,'  FUB(I):',1P,G16.8)             ELE04740
C                                                                       ELE04750
C------ ADJUST F FOR UNBALANCED LOAD P                                 ELE04760
            F=P-FT                                                      ELE04770
      IOPT=IOPT                                                         ELE04780
      RETURN                                                            ELE04790
C------------------------ DETERMINE TOTAL FORCES, ENERGIES ECT --------ELE04800
  500 CONTINUE                                                          ELE04810
      IF( ICOMN .NE. 1 .AND. ICOMN .NE. 2 ) THEN                        ELE04820
       IF (PRINT.AND.FIRST) WRITE (6,491) STEPID                        ELE04830
C                                                                       ELE04840
C                                                                       ELE04850
C------ TOTAL FORCES, DISPLACEMENTS AND VELOCITIES                      ELE04860
      FT=FT+F                                                           ELE04870
      RT=RT+R                                                           ELE04880
      VT=VT+V                                                           ELE04890
      IF ( ABS(FT).GT.ABS(FMAX(1)) ) THEN                               ELE04900
        FMAX(1)=FT                                                      ELE04910
      ENDIF                                                             ELE04920
      IF ( ABS(RT).GT.ABS(FMAX(2)) ) THEN                               ELE04930
        FMAX(2)=RT                                                      ELE04940
      ENDIF                                                             ELE04950
C                                                                       ELE04960
C------ PRINT DATA                                                     ELE04970
      HH= H                                                             ELE04980
      IF (PRINT) WRITE(6,492) IELNO,1,TYPE(KTYPE),NODEI,NODEJ,FT,RT,HH  ELE04990
      ELSE IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                    ELE05000
       IF (PRINT.AND.FIRST) WRITE (6,497) STEPID                        ELE05010
       FT=ASAF(1)                                                       ELE05020
       IF (PRINT) WRITE(6,492) IELNO,1,TYPE(KTYPE),NODEI,NODEJ,FT,HH    ELE05030
      ENDIF                                                             ELE05040
  497 FORMAT (//'       GOEL FORCES...',5X,A/,                          ELE05050
     & '   ELEMENT LOAD',T17,'      TYPE',' NODEI',' NODEJ',           ELE05060
     & '      FORCE     ',' STIFFNESS    ')                           ELE05070
C                                                                       ELE05080
C------ TRANSFER TEMPROARY HYSTERESIS DATA TO PERMANENT STORAGE...      ELE05090
      ISE2=ISE/2                                                        ELE05100
      IF (.NOT. ELSTIC) THEN                                            ELE05110
        DO 510 I=1,ISE2                                                 ELE05120
  510     SE(I)=SE(I+ISE2)                                              ELE05130
      ENDIF                                                             ELE05140
C                                                                       ELE05150
      RETURN                                                            ELE05160
C                                                                       ELE05170
C------------------------ DETERMINE FIXED END FORCES, AND ADD TO FORCE-ELE05180
  600 RETURN                                                            ELE05190
C                                                                       ELE05200
C-------- FIXED END FORCES ARE NOT AVAILABLE FOR THE STRUT              ELE05210
C                                                                       ELE05220
C---------------------- WRITE MEMBER FORCES TO OUTPUT FILE ------------ELE05230
  700 CONTINUE                                                          ELE05240
C                                                                       ELE05250
      CALL MATLIB                                                       ELE05260
     & ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE, DUCT,EXCR,     ELE05270
     &   FT   ,F      ,RT    ,R    ,0.    ,0.    ,LELEM ,LMAT  ,       ELE05280
     &   MAT  ,MATTYP,SE,IELNO,DAMAGE)                                  ELE05290
C                                                                       ELE05300
      WRITE (IUNIT,710)  NODEI,NODEJ,KTYPE,FT,RT,H,EESE,EPSE,          ELE05310
     &                   (DUCT(I),EXCR(I),I=1,3)                        ELE05320
  710 FORMAT (3I6,1P,5G12.4/6G12.4)                                     ELE05330
C                                                                       ELE05340
      RETURN                                                            ELE05350
C                                                                       ELE05360
C-------------------- READ AND PRINT MEMBER FORCES FROM OUTPUT FILE ----ELE05370
  800 CONTINUE                                                          ELE05380
      BACKSPACE(IUNIT)                                                  ELE05390
      READ (IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT                        ELE05400
      ISTEP=-IWRITE                                                     ELE05410
C                                                                       ELE05420
  810 READ (IUNIT,815,END=830) NODEI,NODEJ,KTYPE,FT,RT,H,ESE,PSE,       ELE05430
     &                         (WORK(I),I=1,6)                        ELE05440
  815 FORMAT (3I6,5G12.4/6G12.4)                                        ELE05450
        ISTEP=ISTEP+IWRITE                                              ELE05460
        T=TO+DT*ISTEP                                                   ELE05470
        IF (ISTEP.EQ.0) WRITE (6,805) NODEI,NODEJ,TYPE(KTYPE)           ELE05480
        IF (MOD(ISTEP,INC).EQ.0)                                        ELE05490
     &  WRITE (6,820) ISTEP,T,FT,RT,H,ESE,PSE                           ELE05500
        GO TO 810                                                       ELE05510
C                                                                       ELE05520
  805 FORMAT (//' GOEL MEMBER FORCES...',/                              ELE05530
     & 5X,'NODEI:',I6,5X,'NODEJ:',I6,5X,'GOEL'S TYPE:',A,//             ELE05540
     & '      STEP     TIME       ',                                   ELE05550
     & '   FORCE     ',' DISPLACMENT ',' STIFFNESS',                    ELE05560
     & '      ESE       ',' PSE      ')                                ELE05570
  820 FORMAT (I10,1P,G15.5,6G13.5)                                      ELE05580
C                                                                       ELE05590
  830 WRITE (6,840) (WORK(I),I=1,6)                                     ELE05600
  840 FORMAT (/T10,' MAXIMUN DUCTILITIES AND EXCURSION RATIOS'/         ELE05610
     &        T10,'========================================'/          ELE05620
     &  T15,'DISPLACEMENT DEFINATION: U1=',1P,G12.4,5X,'E1=',G12.4/     ELE05630
     &  T15,'ENERGY DEFINATION #1:    U2=',1P,G12.4,5X,'E2=',G12.4/     ELE05640
     &  T15,'ENERGY DEFINATION #2:    U3=',1P,G12.4,5X,'E3=',G12.4)     ELE05650
      RETURN                                                            ELE05660
C                                                                       ELE05670
```

```
C********************* DETERMINE THE LENGTH OF THE MATERIAL DATA          ELE05680
  900 CONTINUE                                                           ELE05690
      ISE=0                                                              ELE05700
C                                                                        ELE05710
        MAT   = RINPUT( 1)                                               ELE05720
        CALL MATLIB                                                      ELE05730
     &  ( 0    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,  DUCT,EXCR,    ELE05740
     &    FT     ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,      ELE05750
     &    MAT    ,MATTYP,SE,IELNO,DAMAGE)                                ELE05760
      ISE=ISE+LELEM                                                      ELE05770
C                                                                        ELE05780
C------- DOUBLE THE STORAGE, TO ALLOW FOR TEMPROARY VALUES...            ELE05790
      ISE=ISE*2                                                          ELE05800
      RETURN                                                             ELE05810
C                                                                        ELE05820
C********************* DETERMINE THE STRAIN ENERGY                      ELE05830
 1000 CONTINUE                                                           ELE05840
C                                                                        ELE05850
C********************* DUCTILITIES AND EXCURSION RATIO'S                ELE05860
 1100 CONTINUE                                                           ELE05870
C.... OPTIONS 10 AND 11 HAVE THE SAME CODE FOR THIS ELEMENT...           ELE05880
C.... THE ENERGY WAS CALCULATED WITH THE LAST CALL FOR INCRIMENTAL FORCE ELE05890
C     THE TOTAL FORCES HAVE NOT BEEN CALLED YET,THUS THE ENERGIES RESIDE ELE05900
C     IN THE MATRIX SE BEGINING AT ADDRESS ISE21.                        ELE05910
C                                                                        ELE05920
      ISE21=ISE/2 +1                                                     ELE05930
      CALL MATLIB                                                        ELE05940
     &  ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,  DUCT,EXCR,    ELE05950
     &    FT     ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,      ELE05960
     &    MAT    ,MATTYP,SE(ISE21),IELNO,DAMAGE)                         ELE05970
C                                                                        ELE05980
      IF(ABS(DUCT(1)) .LT. 1 ) DUCT(1)=0                                 ELE05990
      IF( DUCFAG .EQ. 0 ) THEN                                           ELE06000
         IF( ABS(DUCT(1)) .LT. 1 ) DUCFAG=0                              ELE06010
         IF( ABS(DUCT(1)) .GE. 1 ) DUCFAG=ABS(DUCT(1))                   ELE06020
      ENDIF                                                              ELE06030
C                                                                        ELE06040
      IF (ELSTIC) EESE=(FT+F)*(RT+R)*0.50                                ELE06050
C                                                                        ELE06060
      IF (PRINT.AND.FIRST) WRITE (6,1191)                                ELE06070
C     IF (PRINT) WRITE (6,1192) IELNO,(DUCT(I),EXCR(I),I=1,3)            ELE06080
C1191 FORMAT (//' STRUT DUCTILITIES AND EXCURSION RATIOS....',5X/        ELE06090
C     &  '   ELEMENT',T12,'|- DUCTILITY DEFINATION #1 -| ',              ELE06100
C     &                  '1234567890ABCDE1234567890ABCDE                 ELE06110
C     &                  '|- DUCTILITY DEFINATION #2 -| ',               ELE06120
C     &                  '|- DUCTILITY DEFINATION #3 -| ',/ T12,         ELE06130
C     &  '   DUCTILITY    EXCURSION     ',                              ELE06140
C     &  '   DUCTILITY    EXCURSION     ',                              ELE06150
C     &  '   DUCTILITY    EXCURSION     ')                              ELE06160
C       1234567890ABCDE1234567890ABCDE                                   ELE06170
      IF (PRINT) WRITE (6,1192) IELNO,DUCT(1),EXCR(1)                    ELE06180
 1191 FORMAT (//' STRUT DUCTILITIES AND EXCURSION RATIOS....',5X/        ELE06190
     &  '   ELEMENT',T12,'|- DUCTILITY DEFINATION #1 -| ',/ T12,         ELE06200
     &  '   DUCTILITY    EXCURSION     ')                              ELE06210
 1192 FORMAT (I10,1P,6G15.6)                                             ELE06220
C                                                                        ELE06230
      RETURN                                                             ELE06240
C                                                                        ELE06250
C********************* MAXIMUM FORCES                                   ELE06260
 1200 CONTINUE                                                           ELE06270
      IF (FIRST) WRITE (6,491) 'MAXIMUM LOAD AND DISPL AT MAXIMUM LOAD ' ELE06280
     &           //' NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'  ELE06290
      WRITE(6,492) IELNO,0,TYPE(KTYPE),NODEI,NODEJ,FMAX(1),FMAX(2)       ELE06300
      RETURN                                                             ELE06310
C                                                                        ELE06320
C                                                                        ELE06330
C********************* RESET INTERNAL FORCES                            ELE06340
 1300 CONTINUE                                                           ELE06350
C                                                                        ELE06360
C----- ZERO MATRICIES                                                    ELE06370
        F =0                                                             ELE06380
        FT=0                                                             ELE06390
        R =0                                                             ELE06400
        RT=0                                                             ELE06410
        V =0                                                             ELE06420
        VT=0                                                             ELE06430
        FMAX(1)=0                                                        ELE06440
        FMAX(2)=0                                                        ELE06450
      SLK=SE(2)                                                          ELE06460
      DO 1310 I=1,ISE                                                    ELE06470
 1310   SE(I)=0                                                          ELE06480
      SE(2)=SLK                                                          ELE06490
      SE(5)=LENGTH                                                       ELE06500
C                                                                        ELE06510
      WRITE (6,1330) IELNO                                               ELE06520
 1330 FORMAT (' GOEL ........... ELEMENT #',I6,                         ELE06530
     &  ' INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO')      ELE06540
      CALL MATLIB                                                        ELE06550
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,  DUCT,EXCR,    ELE06560
     &    FT     ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,      ELE06570
     &    MAT    ,MATTYP,SE,IELNO,DAMAGE)                                ELE06580
      RETURN                                                             ELE06590
C                                                                        ELE06600
C                                                                        ELE06610
C********************* DAMAGE INDEX ***********************             ELE06620
 1400 IF (ELSTIC) RETURN                                                 ELE06630
C                                                                        ELE06640
      FMAX1=FMAX(1)                                                      ELE06650
      FMAX2=FMAX(2)                                                      ELE06660
      CALL MATLIB                                                        ELE06670
     &  ( 5    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,  DUCT,EXCR,    ELE06680
     &    FMAX1,0.    ,FMAX2,   0.,  0.  ,0.    ,LELEM ,LMAT  ,          ELE06690
     &    MAT    ,MATTYP,SE,IELNO,DAMAGE)                                ELE06700
      RETURN                                                             ELE06710
C                                                                        ELE06720
      IF (PRINT.AND.FIRST) WRITE (6,1410)                                ELE06730
      IF (PRINT) WRITE (6,1420) IELNO,DAMAGE,FMAX1,FMAX2,EESE,EPSE       ELE06740
 1410 FORMAT (/                                                          ELE06750
     &  '   ELEMENT',T12,' DAMAGE INDEX       F MAX       ',           ELE06760
     &                  '1234567890ABCDE1234567890ABCDE                 ELE06770
     &                  '     D MAX          ESE         ',             ELE06780
     &                  '     PSE         ')                            ELE06790
 1420 FORMAT (I10,1P,6G15.6)                                             ELE06800
C                                                                        ELE06810
      DAMAGE=DAMAGE*(EESE+EPSE)                                          ELE06820
      RETURN                                                             ELE06830
C                                                                        ELE06840
      END                                                                ELE06850
@PROCESS SDUMP GOSTMT XREF                                               ELE00010
C*************************************************************************ELE00020
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                     ELE00030
C           END DEBUG                                                    ELE00040
C*************************************************************************ELE00050
      SUBROUTINE ELE09                                                   ELE00060
     &  (IOPT       ,IUNIT      ,INC,NEWK   ,FIRST      ,PRINT      ,    ELE00070
     &   NDOF       ,NLOAD      ,MAXNOD     ,NNODE      ,NCOS       ,    ELE00080
     &   ID,ELSTIC  ,IDOF       ,COORD      ,COSINE     ,JTCOS      ,    ELE00090
     &   CONST      ,DISPL      ,FORCE      ,IREL       ,BUG        ,    ELE00100
     &   IBUG       ,ACCEL      ,KGDATA     ,GRAV       ,PGEOM      ,    ELE00110
     &   MAXELD     ,NELD       ,IELD       ,ELD        ,NAME       ,    ELE00120
     &   STEPID     ,AXIAL      ,IELNO      ,RINPUT     ,PUB        ,    ELE00130
     &   EESE       , EPSE,DAMAGE,DUCFAG    ,IELTYP     ,IELDOF     ,    ELE00140
     &   MTA        ,MTB        ,MZA        ,MZB        ,MXA        ,    ELE00150
     &   MFXA       ,STIF1Y     ,STIF2Y     ,STIF1Z     ,STIF2Z     ,    ELE00160
     &   YILDGJ,YILDEA,YILD1Y,YILD2Y,YILD1Z,YILD2Z,                      ELE00170
     &   ISEMTA,ISEMTB,ISEMZA,ISEMZB,ISEMXA,ISEFXA,PKG  ,                ELE00180
     &   R,RT ,F,FT   ,LM,CTS,CTE    ,BS     ,BE     ,                  ELE00190
     &   NODEI      ,NODEJ      ,JOINTI     ,JOINTJ     ,               ELE00200
     &   REL        ,LENGTH     ,KGDOF      ,LKG        ,LMKG       ,    ELE00210
C    &   ISE        ,ENGDAT     ,FMAX       ,                           ELE00220
     &   ISE        ,ENGDAT     ,FMAX       ,IAFAG      ,STBFAG     ,    ELE00230
     &   MATMTA     ,MATMTB     ,MATM3A     ,MATM3B     ,MATMXA     ,    ELE00240
     &   MATFXA,H,S22,S26,S212,S33,S35,S311,Q,BT,DT,AZ,CZ,BYY,AZZ   ,    ELE00250
     &   ROIETA,ROIETB,ROIEZA,ROIEZB,                                    ELE00260
     &   A          ,SE          ,STIFF      ,FEM        )               ELE00270
C                                                                        ELE00280
      IMPLICIT REAL(A-H,O-Z)                                             ELE00290
      COMMON /IDSTEP/ JSTEP                                              ELE00300
C                                                                        ELE00310
      COMMON /RSA/ICOMN,RSAF(12)                                         ELE00320
      REAL ICOMN                                                         ELE00330
C                                                                        ELE00340
      LOGICAL ERR,FIRST,PRINT,BUG,FEMPLG,FALSE,AXIAL,BTEST              ELE00350
      LOGICAL NEWK,ELSTIC                                                ELE00360
      INTEGER IAFAG,STBFAG                                               ELE00370
      CHARACTER*80 NAME                                                  ELE00380
      CHARACTER*(*) STEPID                                               ELE00390
      CHARACTER*6  IRELC                                                 ELE00400
      DIMENSION KGDATA(3)                                                ELE00410
      DIMENSION  IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS)         ELE00420
      DIMENSION CONST(MAXNOD,6),   ID(MAXNOD)  ,JTCOS(MAXNOD)            ELE00430
      DIMENSION ASAT(12,12),SAT(12,12),STIFF(156)                        ELE00440
C     DIMENSION A(12,12),  S(12,12),LM(12),LMT(12),LMKG(12),FMAX(12)     ELE00450
      DIMENSION A(12,12),  S(12,12),LM(12),LMT(12),LMKG(12),FMAX(144)    ELE00460
      DIMENSION CTS( 3, 3),CTE( 3, 3),COSLOC(3,3),VX(3),VT(3),VZ(3)      ELE00470
      DIMENSION BS( 3, 3),BE( 3, 3),SE(ISE),ENGDAT(2)                    ELE00480
      DIMENSION R(12),DISPL(NDOF,NLOAD),F(12),RT(12),FT(12),FUB(NDOF)    ELE00490
      DIMENSION FORCE(NDOF,NLOAD),IELD(5,MAXELD),ELD(12,MAXELD)          ELE00500
      DIMENSION ASA(12,12),RINPUT(100),FEM(12,NLOAD),GFEM(12)            ELE00510
      DIMENSION DUCT(3,6),EXCR(6,6),P(6)                                 ELE00520
      DIMENSION V(12),VL(12),EFUB(12),UP(12)                             ELE00530
      REAL LENGTH                                                        ELE00540
      INTEGER REL,RELT                                                   ELE00550
      DIMENSION MD(13)                                                   ELE00560
      DATA MD/1,2,4,7,11,16,22,29,37,46,56,67,79/                        ELE00570
C                                                                        ELE00580
C------------------------------------------------------------------------ELE00590
C VARIABLES:                                                             ELE00600
C------------------------------------------------------------------------ELE00610
C---------- GLOBAL VARIABLES                                            ELE00620
C IOPT   = 1, INITIALIZE BEAM COLUMN ELEMENT                             ELE00630
C        = 2, CALCULATE GEOMETRIC STIFFNESS                              ELE00640
C        = 3, FORM STIFFNESS                                             ELE00650
C        = 4, CALCULATE INCREMENTAL FORCES                               ELE00660
C        = 5, CALCULATE TOTAL FORCES AND ENERGIES                        ELE00670
C IUNIT  = OUTPUT FILE UNIT # FOR PRINTING OUTPUT                        ELE00680
C ERR    = ERR OR FLAG, IF AN ERR OR HAS OCCURED, ERR=.TRUE.             ELE00690
C FIRST  = FLAG, FIRST=.TRUE., PRINT HEADERS                             ELE00700
C PRINT  = FLAG, PRINT=.TRUE., PRINT DATA                                ELE00710
C NDOF   = # OF GLOBAL DOF                                               ELE00720
C NLOAD  = # OF LOAD COMBINATIONS                                        ELE00730
C MAXNOD = # ROW DIMENSION OF NODE ARRAY                                 ELE00740
C NNODE  = # OF NODES                                                    ELE00750
C ID     = ARRAY OF EXTERNAL NODE NUMBERS                                ELE00760
C IDOF   = ARRAY OF DEGREES OF FREEDOM                                   ELE00770
C COORD  = ARRAY OF COORDINATES                                          ELE00780
C COSINE = ARRAY OF DIRECTION COSINES OF NODES                           ELE00790
C CONST  = ARRAY OF CONSTRAINT TRANSFORMATIONS                           ELE00800
C DISPL  = GLOBAL DISPLACEMENT MATRIX                                    ELE00810
C IZMAT  = LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR            ELE00820
C NAME   = ELEMENT NAME                                                  ELE00830
C IELNO  = ELEMENT NUMBER                                                ELE00840
C RINPUT = INPUT DATA                                                    ELE00850
C STIFF  = OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)              ELE00860
C LM     = OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX               ELE00870
C---------- ELEMENT VARIABLES                                           ELE00880
C H,Q,AI,AJ,C,BI,BJ,S22,S26,S212,S33,S35,S311,D                          ELE00890
C        = INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX          ELE00900
C PKG    = INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX...          ELE00910
C R      = INCRIMENTAL DISPLACEMENTS                                     ELE00920
C RT     = TOTAL DISPLACEMENTS                                           ELE00930
C F      = INCRIMENTAL FORCES                                            ELE00940
C FT     = TOTAL FORCES                                                  ELE00950
C LM     = LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                      ELE00960
C CTS    = LOCAL TO GLOBAL ROTATION MATRIX START                         ELE00970
C CTE    = LOCAL TO GLOBAL ROTATION MATRIX END                           ELE00980
C BS     = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX START ELE00990
C BE     = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END   ELE01000
C STIFF  = ELEMENT STIFFNESS                                             ELE01010
C IELTYP = ELEMENT TYPE = 9                                              ELE01020
C NODEI  = EXTERNAL JOINT NUMBER, END I                                  ELE01030
C NODEJ  = EXTERNAL JOINT NUMBER, END J                                  ELE01040
C JOINTI = INTERNAL JOINT NUMBER, END I                                  ELE01050
C JOINTJ = INTERNAL JOINT NUMBER, END J                                  ELE01060
C---------- TEMPROARY VARIABLES                                          ELE01070
C VX     = VECTOR DEFINING THE ELEMENT X AXIS                            ELE01080
C VY     = VECTOR DEFINING THE ELEMENT Y AXIS                            ELE01090
C COSLOC = LOCAL COSINE MATRIX                                           ELE01100
C A      = GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                   ELE01110
C S      = ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART    ELE01120
C SAT    = PRODUCT OF S*A'                                               ELE01130
C ASAT   = PRODUCT OF A*SAT                                              ELE01140
C IAFAG  = 1 : INDEX FOR PINO-SUAREZ COLUMN CONSIDERING LOCAL BUCKLING   ELE01150
C          2 : PINO-SUAREZ COLUMN A END LOCAL BUCKLING OCCURRED          ELE01160
C          3 : PINO-SUAREZ COLUMN B END LOCAL BUCKLING OCCURRED          ELE01170
C          4 : PINO-SUAREZ COLUMN A AND B ENDS LOCAL BUCKLING OCCURRED   ELE01180
C          5 : INDEX FOR CHECKING BOX COLUMN STRENGH CRITERIA            ELE01190
C          6 : BOX COLUMN A END REACH TO FULLY PLASTIC                   ELE01200
C          7 : BOX COLUMN B END REACH TO FULLY PLASTIC                   ELE01210
C          8 : BOX COLUMN A AND B ENDS REACH TO FULLY PLASTIC            ELE01220
C------------------------------------------------------------------------ELE01230
C                                                                        ELE01240
C-------- CHOOSE OPTION                                                  ELE01250
        IF (IOPT.LT.1 .OR.IOPT.GT.14)                                    ELE01260
     &  WRITE (6,*) 'INVALID OPTION IN ELE-09, IOPT=',IOPT               ELE01270
C                                                                        ELE01280
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,         ELE01290
     &                                          1300,1400),IOPT          ELE01300
C                                                                        ELE01310
  100 CONTINUE                                                           ELE01320
C                                                                        ELE01330
C------ ZERO STRAIN ENERGY                                              ELE01340
      ENGDAT(1) = 0                                                      ELE01350
      ENGDAT(2) = 0                                                      ELE01360
C                                                                        ELE01370
      IF ( BTEST(IBUG,7) ) THEN                                          ELE01380
         WRITE (6,*) NAME                                                ELE01390
         CALL WMATRX(RINPUT,14, 1,'RINPUT')                              ELE01400
      ENDIF                                                              ELE01410
C                                                                        ELE01420
C------ INTERPERTATE INPUT DATA                                         ELE01430
      IELTYP= 9                                                          ELE01440
      MATMTA = RINPUT( 1)                                                ELE01450
      MATMTB = RINPUT( 2)                                                ELE01460
      MATM3A = RINPUT( 3)                                                ELE01470
      MATM3B = RINPUT( 4)                                                ELE01480
      MATMXA = RINPUT( 5)                                                ELE01490
      MATFXA = RINPUT( 6)                                                ELE01500
      NODEI  = RINPUT( 7)                                                ELE01510
      NODEJ  = RINPUT( 8)                                                ELE01520
      VY(1)  = RINPUT( 9)                                                ELE01530
      VY(2)  = RINPUT(10)                                                ELE01540
      VY(3)  = RINPUT(11)                                                ELE01550
      XS     = RINPUT(12)                                                ELE01560
      XE     = RINPUT(13)                                                ELE01570
      PKG    = RINPUT(14)                                                ELE01580
C                                                                        ELE01590
      IAFAG =0                                                           ELE01600
      STBFAG=0                                                           ELE01610
C                                                                        ELE01620
C------ GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S           ELE01630
      JOINTI=IQUICK(NODEI,ID,NNODE)                                      ELE01640
      JOINTJ=IQUICK(NODEJ,ID,NNODE)                                      ELE01650
C     WRITE (6,*) 'ID=',ID                                               ELE01660
C                                                                        ELE01670
```

```
C----- PRINT DATA FOR PLOTTING                                          ELE01680
      IF (BTEST(IBUG,2)) THEN                                           ELE01690
        WRITE(7,10) NODEJ, NODEI,                                       ELE01700
     &              (COORD(JOINTJ,I),I=1,3),(COORD(JOINTI,J),J=1,3)     ELE01710
   10 FORMAT (I5,1X,I5,1P,6G12.4)                                       ELE01720
      ENDIF                                                             ELE01730
C                                                                       ELE01740
C-----------------------------------------------------------------      ELE01750
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                            ELE01760
C-----------------------------------------------------------------      ELE01770
C----- DEFINE VECTOR X, LENGTH                                          ELE01780
      VX(1)=COORD(JOINTJ,1)-COORD(JOINTI,1)                             ELE01790
      VX(2)=COORD(JOINTJ,2)-COORD(JOINTI,2)                             ELE01800
      VX(3)=COORD(JOINTJ,3)-COORD(JOINTI,3)                             ELE01810
      LENGTH=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)-XS-XE                     ELE01820
      IF (LENGTH.LE.0) THEN                                             ELE01830
         WRITE(6,101) IELNO,JOINTI,JOINTJ,LENGTH                        ELE01840
  101    FORMAT(1X,20('*'),'ER','ROR - LENGTH IS LE 0',/                ELE01850
     &    5X,'IELNO=',I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,              ELE01860
     &    5X,'LENGTH',1P,G15.6/                                         ELE01870
     &    5X,'REVISE INPUT.....................'/)                      ELE01880
         LENGTH=1.0                                                     ELE01890
      ENDIF                                                             ELE01900
      IF ( BTEST(IBUG,7) ) THEN                                         ELE01910
         WRITE (6,*) 'JOINTS',JOINTI,JOINTJ                             ELE01920
         WRITE (6,*) 'VX',VX                                            ELE01930
         WRITE (6,*) 'VY',VY                                            ELE01940
         WRITE (6,*) 'LENGTH  ',LENGTH                                  ELE01950
      ENDIF                                                             ELE01960
C----- GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES CT AND B            ELE01970
      ICS=JTCOS(JOINTI)                                                 ELE01980
      ICE=JTCOS(JOINTJ)                                                 ELE01990
C                                                                       ELE02000
      CALL ROTXYZ(VX,VY,COSINE(1,1,ICS),CTS, XS ,0.,0.,BS,              ELE02010
     &      CONST(JOINTI,1),MAXNOD)                                     ELE02020
      CALL ROTXYZ(VX,VY,COSINE(1,1,ICE),CTE,-XE ,0.,0.,BE,              ELE02030
     &      CONST(JOINTJ,1),MAXNOD)                                     ELE02040
      IF ( BTEST(IBUG,7) ) THEN                                         ELE02050
           CALL WMATRX( CTS   , 3, 3,'CTS   ')                          ELE02060
           CALL WMATRX( CTE   , 3, 3,'CTE   ')                          ELE02070
      ENDIF                                                             ELE02080
C                                                                       ELE02090
C----- GET GLOBAL CONSTRAINT MATRIX BS, BE                              ELE02100
C     BS(1,1)=CONST(JOINTI,1)*CTS(2,1)+CONST(JOINTI,2)*CTS(3,1)+BS(1,1) ELE02110
C     BS(1,2)=CONST(JOINTI,1)*CTS(2,2)+CONST(JOINTI,2)*CTS(3,2)+BS(1,2) ELE02120
C     BS(1,3)=CONST(JOINTI,1)*CTS(2,3)+CONST(JOINTI,2)*CTS(3,3)+BS(1,3) ELE02130
C     BS(2,1)=CONST(JOINTI,3)*CTS(1,1)+CONST(JOINTI,4)*CTS(3,1)+BS(2,1) ELE02140
C     BS(2,2)=CONST(JOINTI,3)*CTS(1,2)+CONST(JOINTI,4)*CTS(3,2)+BS(2,2) ELE02150
C     BS(2,3)=CONST(JOINTI,3)*CTS(1,3)+CONST(JOINTI,4)*CTS(3,3)+BS(2,3) ELE02160
C     BS(3,1)=CONST(JOINTI,5)*CTS(1,1)+CONST(JOINTI,6)*CTS(2,1)+BS(3,1) ELE02170
C     BS(3,2)=CONST(JOINTI,5)*CTS(1,2)+CONST(JOINTI,6)*CTS(2,2)+BS(3,2) ELE02180
C     BS(3,3)=CONST(JOINTI,5)*CTS(1,3)+CONST(JOINTI,6)*CTS(2,3)+BS(3,3) ELE02190
C                                                                       ELE02200
C     BE(1,1)=CONST(JOINTJ,1)*CTE(2,1)+CONST(JOINTJ,2)*CTE(3,1)+BE(1,1) ELE02210
C     BE(1,2)=CONST(JOINTJ,1)*CTE(2,2)+CONST(JOINTJ,2)*CTE(3,2)+BE(1,2) ELE02220
C     BE(1,3)=CONST(JOINTJ,1)*CTE(2,3)+CONST(JOINTJ,2)*CTE(3,3)+BE(1,3) ELE02230
C     BE(2,1)=CONST(JOINTJ,3)*CTE(1,1)+CONST(JOINTJ,4)*CTE(3,1)+BE(2,1) ELE02240
C     BE(2,2)=CONST(JOINTJ,3)*CTE(1,2)+CONST(JOINTJ,4)*CTE(3,2)+BE(2,2) ELE02250
C     BE(2,3)=CONST(JOINTJ,3)*CTE(1,3)+CONST(JOINTJ,4)*CTE(3,3)+BE(2,3) ELE02260
C     BE(3,1)=CONST(JOINTJ,5)*CTE(1,1)+CONST(JOINTJ,6)*CTE(2,1)+BE(3,1) ELE02270
C     BE(3,2)=CONST(JOINTJ,5)*CTE(1,2)+CONST(JOINTJ,6)*CTE(2,2)+BE(3,2) ELE02280
C     BE(3,3)=CONST(JOINTJ,5)*CTE(1,3)+CONST(JOINTJ,6)*CTE(2,3)+BE(3,3) ELE02290
C                                                                       ELE02300
      IF ( BTEST(IBUG,7) ) THEN                                         ELE02310
         CALL WMATRX(BS, 3, 3,'BS   ')                                  ELE02320
         CALL WMATRX(BE, 3, 3,'BE   ')                                  ELE02330
      ENDIF                                                             ELE02340
C                                                                       ELE02350
C----- ZERO MATRICIES                                                   ELE02360
       DO 40 I=1,12                                                     ELE02370
         FMAX(I)=0                                                      ELE02380
         F (I)=0                                                        ELE02390
         FT(I)=0                                                        ELE02400
         R (I)=0                                                        ELE02410
         RT(I)=0                                                        ELE02420
         V (I)=0                                                        ELE02430
         VL(I)=0                                                        ELE02440
         DO 40 J=1,12                                                   ELE02450
           S(I,J)=0.                                                    ELE02460
   40    A(I,J)=0.                                                      ELE02470
C                                                                       ELE02480
C----- ASSEMBLE TRANSFORMATION MATRIX A                                 ELE02490
       DO 50 I=1,3                                                      ELE02500
       DO 50 J=1,3                                                      ELE02510
         A(I  ,J  )=CTS(I,J)                                            ELE02520
         A(I+3,J+3)=CTS(I,J)                                            ELE02530
         A(I+6,J+6)=CTE(I,J)                                            ELE02540
         A(I+9,J+9)=CTE(I,J)                                            ELE02550
         A(I+3,J  )=BS(I,J)                                             ELE02560
   50    A(I+9,J+6)=BE(I,J)                                             ELE02570
      IF ( BTEST(IBUG,7) ) CALL WMATRX( A    ,12,12,' A    ')          ELE02580
C                                                                       ELE02590
C----- GET MATERIAL PROPERTIES                                          ELE02600
        FALSE=.FALSE.                                                   ELE02610
        LSTTP=0                                                         ELE02620
        CALL MATLIB                                                     ELE02630
     &(2 ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,1),EXCR(1,1), ELE02640
     &  FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,        ELE02650
     &  MATMYA,MATTYP,SE(ISEMTA),IELNO,DAMAGE)                          ELE02660
          MYA=MATTYP                                                    ELE02670
          IF( MATTYP .NE. 9 .AND. MATTYP .NE. 10 .AND. MATTYP .NE. 11   ELE02680
     &                      .AND. MATTYP .NE. 6 )                       ELE02690
     &         WRITE(6,190)IELNO,MATTYP,NODEI,NODEJ                     ELE02700
        CALL MATLIB                                                     ELE02710
     &(2 ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,2),EXCR(1,2), ELE02720
     &  FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,        ELE02730
     &  MATMYB,MATTYP,SE(ISEMTB),IELNO,DAMAGE)                          ELE02740
          MYB=MATTYP                                                    ELE02750
          IF( MATTYP .NE. 9 .AND. MATTYP .NE. 10 .AND. MATTYP .NE. 11   ELE02760
     &                      .AND. MATTYP .NE. 6 )                       ELE02770
     &         WRITE(6,190)IELNO,MATTYP,NODEI,NODEJ                     ELE02780
        CALL MATLIB                                                     ELE02790
     &(2 ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,3),EXCR(1,3), ELE02800
     &  FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,        ELE02810
     &  MATMZA,MATTYP,SE(ISEMZA),IELNO,DAMAGE)                          ELE02820
          MZA=MATTYP                                                    ELE02830
          IF( MATTYP .NE. 9 .AND. MATTYP .NE. 10 .AND. MATTYP .NE. 11   ELE02840
     &                      .AND. MATTYP .NE. 6 )                       ELE02850
     &         WRITE(6,190)IELNO,MATTYP,NODEI,NODEJ                     ELE02860
        CALL MATLIB                                                     ELE02870
     &(2 ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,4),EXCR(1,4), ELE02880
     &  FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,        ELE02890
     &  MATMZB,MATTYP,SE(ISEMZB),IELNO,DAMAGE)                          ELE02900
          MZB=MATTYP                                                    ELE02910
          IF( MATTYP .NE. 9 .AND. MATTYP .NE. 10 .AND. MATTYP .NE. 11   ELE02920
     &                      .AND. MATTYP .NE. 6 )                       ELE02930
     &         WRITE(6,190)IELNO,MATTYP,NODEI,NODEJ                     ELE02940
        CALL MATLIB                                                     ELE02950
     &(2 ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,5),EXCR(1,5), ELE02960
     &  FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,        ELE02970
     &  MATMXA,MATTYP,SE(ISEMXA),IELNO,DAMAGE)                          ELE02980
          MXA=MATTYP                                                    ELE02990
          IF( MATTYP .NE. 10 ) WRITE(6,190)IELNO,MATTYP,NODEI,NODEJ     ELE03000
C                                                                       ELE03010
       SE(ISEFXA+4)=LENGTH                                              ELE03020
        CALL MATLIB                                                     ELE03030
     &(2 ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,6),EXCR(1,6), ELE03040
     &  FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,        ELE03050
     &  MATFXA,MATTYP,SE(ISEFXA),IELNO,DAMAGE)                          ELE03060
          MFXA=MATTYP                                                   ELE03070
          IF( MATTYP .NE. 10 .AND. MATTYP .NE. 13)                      ELE03080
     &     WRITE(6,190)IELNO,MATTYP,NODEI,NODEJ                       ELE03090
C                                                                       ELE03100
  190   FORMAT (' ELEMENT: ',I6,' INCOMPATIBLE MATERIAL PROP.#',I5,     ELE03110
     &    '  START JOINT:',I5,'  END JOINT:',I5,/                       ELE03120
     &    ' THE ELEMENT is REJECTED AND ER','ROR CHECKING CONTINUES')   ELE03130
C                                                                       ELE03140
C ....CHECK INTERACTIVE OPTION                                          ELE03150
      IF( MYA .EQ. 10 .AND. MYB .EQ. 10 .AND.                           ELE03160
     &    MZA .EQ. 10 .AND. MZB .EQ. 10 .AND. MFXA .EQ. 10 ) THEN       ELE03170
          IF( SE(ISEMYA+4) .EQ. 2 .AND. SE(ISEMYB+4) .EQ. 2 .AND.       ELE03180
     &        SE(ISEMZA+4) .EQ. 2 .AND. SE(ISEMZB+4) .EQ. 2    ) THEN   ELE03190
              IAFAG=1                                                   ELE03200
          ENDIF                                                         ELE03210
          IF( SE(ISEMYA+4) .EQ. 3 .AND. SE(ISEMYB+4) .EQ. 3 .AND.       ELE03220
     &        SE(ISEMZA+4) .EQ. 3 .AND. SE(ISEMZB+4) .EQ. 3    ) THEN   ELE03230
              IAFAG=5                                                   ELE03240
          ENDIF                                                         ELE03250
      ENDIF                                                             ELE03260
C                                                                       ELE03270
      EESE=0                                                            ELE03280
      EPSE=0                                                            ELE03290
C                                                                       ELE03300
C--------- NO END RELEASE FOR THIS ELEMENT                              ELE03310
      REL=0                                                             ELE03320
C                                                                       ELE03330
C--------- SET STIFFNESS COIEFFICIENTS...                               ELE03340
C                                                                       ELE03350
C                                                                       ELE03360
C  FOR BENDING IN Y AXIS:                                               ELE03370
      IF( (MYA .EQ. 9 .AND. MYB .EQ. 9) .OR.                            ELE03380
     &    (MYA .EQ.11 .AND. MYB .EQ.11) .OR.                            ELE03390
     &    (MYA .EQ. 6 .AND. MYB .EQ. 6) ) THEN                          ELE03400
          STIF1Y=SE(ISEMYA)                                             ELE03410
          STIF2Y=SE(ISEMYB)                                             ELE03420
          IF( MYA .EQ. 9 ) THEN                                         ELE03430
             SA1=SE(ISEMYA+60)                                          ELE03440
             SA2=SE(ISEMYB+60)                                          ELE03450
          ELSE IF( MYA .EQ.11) THEN                                     ELE03460
             SA1=SE(ISEMYA+30)                                          ELE03470
             SA2=SE(ISEMYB+30)                                          ELE03480
          ELSE IF( MYA .EQ. 6) THEN                                     ELE03490
             SA1=SE(ISEMYA+47)/LENGTH                                   ELE03500
             SA2=SE(ISEMYB+47)/LENGTH                                   ELE03510
             STIF1Y=STIF1Y/LENGTH                                       ELE03520
             STIF2Y=STIF2Y/LENGTH                                       ELE03530
             EI=SE(ISEMYA+48)                                           ELE03540
             TERM1=(LENGTH**2)/(12*EI*EI)                               ELE03550
             TERM2=LENGTH/(3*EI)                                        ELE03560
          ENDIF                                                         ELE03570
          IF ( SA1 .EQ. 0 ) THEN                                        ELE03580
               FLX1=0                                                   ELE03590
          ELSE IF ( SA1 .NE. 0 ) THEN                                   ELE03600
               FLX1=(1/STIF1Y)-(1/SA1)                                  ELE03610
          ENDIF                                                         ELE03620
          IF ( SA2 .EQ. 0 ) THEN                                        ELE03630
               FLX2=0                                                   ELE03640
          ELSE IF ( SA2 .NE. 0 ) THEN                                   ELE03650
               FLX2=(1/STIF2Y)-(1/SA2)                                  ELE03660
          ENDIF                                                         ELE03670
          IF ( MYA .EQ. 6 ) THEN                                        ELE03680
             DCOF=TERM1 + TERM2*(FLX1+FLX2) + FLX1*FLX2                 ELE03690
          ELSE IF ( MYA .NE. 6 ) THEN                                   ELE03700
             DCOF=(3/(SA1**2)) + (2/SA1)*(FLX1+FLX2) + FLX1*FLX2        ELE03710
          ENDIF                                                         ELE03720
C                                                                       ELE03730
          DCOF=1/DCOF                                                   ELE03740
C                                                                       ELE03750
          IF ( MYA .EQ. 6 ) THEN                                        ELE03760
            BY =DCOF*(TERM2 + FLX2)                                     ELE03770
            BYY=DCOF*(TERM2 + FLX1)                                     ELE03780
            DY=DCOF*(0.5*TERM2)                                         ELE03790
          ELSE IF ( MYA .NE. 6 ) THEN                                   ELE03800
            BY =DCOF*((2/SA1)+FLX2)                                     ELE03810
            BYY=DCOF*((2/SA2)+FLX1)                                     ELE03820
            DY=DCOF*(1/SA1)                                             ELE03830
          ENDIF                                                         ELE03840
      ELSE IF( MYA .EQ. 10 .AND. MYB .EQ. 10 ) THEN                     ELE03850
          ELAS=SE(ISEMYA+4)                                             ELE03860
          SP  =SE(ISEMYA+5)                                             ELE03870
          EIY =SE(ISEMYA  )                                             ELE03880
          YILD1Y=SE(ISEMYA+3)                                           ELE03890
          YILD2Y=SE(ISEMYB+3)                                           ELE03900
          IF( YILD1Y .EQ. 1 .OR. YILD2Y .EQ. 1 ) STBFAG=1               ELE03910
C                                                                       ELE03920
          SE(ISEMYA+12)=SE(ISEMYA+1)*LENGTH/(6*EIY)                     ELE03930
          SE(ISEMYB+12)=SE(ISEMYB+1)*LENGTH/(6*EIY)                     ELE03940
C                                                                       ELE03950
          IF( ELAS .EQ. 0 ) THEN                                        ELE03960
               BY =4*EIY/LENGTH                                         ELE03970
               BYY=4*EIY/LENGTH                                         ELE03980
               DY =2*EIY/LENGTH                                         ELE03990
          ELSE IF( ELAS .NE. 0 ) THEN                                   ELE04000
               IF( YILD1Y .EQ. 0 .AND. YILD2Y .EQ. 0 ) THEN             ELE04010
                    BY =4*EIY/LENGTH                                    ELE04020
                    BYY=4*EIY/LENGTH                                    ELE04030
                    DY =2*EIY/LENGTH                                    ELE04040
               ELSE IF( YILD1Y .EQ.1 .AND. YILD2Y .EQ. 0 ) THEN         ELE04050
                    BY =SP*(4*EIY/LENGTH)                               ELE04060
                    BYY=SP*(4*EIY/LENGTH)+(1-SP)*(3*EIY/LENGTH)         ELE04070
                    DY =SP*(2*EIY/LENGTH)                               ELE04080
               ELSE IF( YILD1Y .EQ. 0 .AND. YILD2Y .EQ. 1 ) THEN        ELE04090
                    BY =SP*(4*EIY/LENGTH)+(1-SP)*(3*EIY/LENGTH)         ELE04100
                    BYY=SP*(4*EIY/LENGTH)                               ELE04110
                    DY =SP*(2*EIY/LENGTH)                               ELE04120
               ELSE IF( YILD1Y .EQ. 1 .AND. YILD2Y .EQ. 1 ) THEN        ELE04130
                    BY =SP*(4*EIY/LENGTH)                               ELE04140
                    BYY=SP*(4*EIY/LENGTH)                               ELE04150
                    DY =SP*(2*EIY/LENGTH)                               ELE04160
               ENDIF                                                    ELE04170
          ENDIF                                                         ELE04180
      ELSE                                                              ELE04190
          WRITE(6,69)MYA,MYB,IELNO                                      ELE04200
   69     FORMAT(1X,'** INVALID MATERIAL TYPE MYA=',I2,' MYB=',I2,      ELE04210
     &         ' FOR ELEMENT NO. ',I5,' **')                            ELE04220
      ENDIF                                                             ELE04230
C                                                                       ELE04240
C  FOR BENDING IN Z AXIS:                                               ELE04250
      IF( (MZA .EQ. 9 .AND. MZB .EQ. 9) .OR.                            ELE04260
     &    (MZA .EQ.11 .AND. MZB .EQ.11) .OR.                            ELE04270
     &    (MZA .EQ. 6 .AND. MZB .EQ. 6) ) THEN                          ELE04280
          STIF1Z=SE(ISEMZA)                                             ELE04290
          STIF2Z=SE(ISEMZB)                                             ELE04300
          IF( MZA .EQ. 9 ) THEN                                         ELE04310
            SA1=SE(ISEMZA+60)                                           ELE04320
            SA2=SE(ISEMZB+60)                                           ELE04330
          ELSE IF( MZA .EQ.11 ) THEN                                    ELE04340
            SA1=SE(ISEMZA+30)                                           ELE04350
            SA2=SE(ISEMZB+30)                                           ELE04360
          ELSE IF( MZA .EQ. 6) THEN                                     ELE04370
            SA1=SE(ISEMZA+47)/LENGTH                                    ELE04380
            SA2=SE(ISEMZB+47)/LENGTH                                    ELE04390
            STIF1Z=STIF1Z/LENGTH                                        ELE04400
            STIF2Z=STIF2Z/LENGTH                                        ELE04410
            EI=SE(ISEMZA+48)                                            ELE04420
            TERM1=(LENGTH**2)/(12*EI*EI)                                ELE04430
            TERM2=LENGTH/(3*EI)                                         ELE04440
          ENDIF                                                         ELE04450
          IF ( SA1 .EQ. 0 ) THEN                                        ELE04460
               FLX1=0                                                   ELE04470
          ELSE IF ( SA1 .NE. 0 ) THEN                                   ELE04480
               FLX1=(1/STIF1Z)-(1/SA1)                                  ELE04490
          ENDIF                                                         ELE04500
          IF ( SA2 .EQ. 0 ) THEN                                        ELE04510
```

```
            FLX2=0
         ELSE IF ( SA2 .NE. 0 ) THEN
            FLX2=(1/STIF2S)-(1/SA2)
         ENDIF
         IF ( MZA .EQ. 6 ) THEN
            DCOF=TERM1 + TERM2*(FLX1+FLX2) + FLX1*FLX2
         ELSE IF ( MZA .NE. 6 ) THEN
            DCOF=(3/(SA1**2)) + (2/SA1)*(FLX1+FLX2) + FLX1*FLX2
         ENDIF
C
         DCOF=1/DCOF
C
         IF ( MZA .EQ. 6 ) THEN
           A2 =DCOF*(TERM2 + FLX2)
           A2Z=DCOF*(TERM2 + FLX1)
           C2=DCOF*(0.5*TERM2)
         ELSE IF ( MZA .NE. 6 ) THEN
           A2 =DCOF*((2/SA1)+FLX2)
           A2Z=DCOF*((2/SA2)+FLX1)
           C2=DCOF*(1/SA1)
         ENDIF
      ELSE IF( MZA .EQ. 10 .AND. MZB .EQ. 10 ) THEN
         ELAS=SE(ISEM2A+4)
         SP  =SE(ISEM2A+5)
         EI2 =SE(ISEM3A  )
         YILD12=SE(ISEM2A+3)
         YILD22=SE(ISEM2B+3)
         IF( YILD12 .EQ. 1 .OR. YILD22 .EQ. 1 ) STBFAG=1
C
         SE(ISEM2A+12)=SE(ISEM2A+1)*LENGTH/(6*EI2)
         SE(ISEM2B+12)=SE(ISEM2B+1)*LENGTH/(6*EI2)
C
         IF( ELAS .EQ. 0 ) THEN
              A2 =4*EI2/LENGTH
              A2Z=4*EI2/LENGTH
              C2 =2*EI2/LENGTH
         ELSE IF( YILD12 .EQ. 1 .AND. YILD22 .EQ. 0 ) THEN
              A2 =SP*(4*EI2/LENGTH)
              A2Z=SP*(4*EI2/LENGTH)+(1-SP)*(3*EI2/LENGTH)
              C2 =SP*(2*EI2/LENGTH)
         ELSE IF( YILD12 .EQ. 0 .AND. YILD22 .EQ. 1 ) THEN
              A2 =SP*(4*EI2/LENGTH)+(1-SP)*(3*EI2/LENGTH)
              A2Z=SP*(4*EI2/LENGTH)
              C2 =SP*(2*EI2/LENGTH)
         ELSE IF( YILD12 .EQ. 1 .AND. YILD22 .EQ. 1 ) THEN
              A2 =SP*(4*EI2/LENGTH)
              A2Z=SP*(4*EI2/LENGTH)
              C2 =SP*(2*EI2/LENGTH)
         ENDIF
      ENDIF
      ELSE
         WRITE(6,79)MZA,MZB,IELNO
   79    FORMAT(1X,'** INVALID MATERIAL TYPE MZA=',I2,' MZB=',I2,
     &          ' FOR ELEMENT NO. ',I5,' **')
      ENDIF
C
C  FOR AXIAL IN X AXIS:
      IF( MFXA .EQ. 10 ) THEN
         ELAS=SE(ISEFXA+4)
         SP  =SE(ISEFXA+5)
         EA  =SE(ISEFXA  )
         YILDEA=SE(ISEFXA+3)
         IF( YILDEA .EQ. 1) STBFAG=1
C
         SE(ISEFXA+12)=SE(ISEFXA+1)*LENGTH/EA
C
         IF( ELAS .EQ. 0 ) THEN
             H=EA/LENGTH
         ELSE IF( ELAS .NE. 0 ) THEN
             IF( YILDEA .EQ. 0 ) THEN
                 H=EA/LENGTH
             ELSE IF( YILDEA .EQ. 1 ) THEN
                H=SP*(EA/LENGTH)
             ENDIF
         ENDIF
      ELSE IF( MFXA .EQ. 13 ) THEN
         H=SE(ISEFXA)
      ELSE IF( MFXA .NE. 10 .OR. MFXA .NE. 13 ) THEN
         WRITE(6,89)MFXA,IELNO
   89    FORMAT(1X,'** INVALID MATERIAL TYPE MFXA=',I2,
     &          ' FOR ELEMENT NO. ',I5,' **')
      ENDIF
C
C
C  FOR TORSION IN X AXIS:
      IF( MXA .EQ. 10 ) THEN
         ELAS=SE(ISEMXA+4)
         SP  =SE(ISEMXA+5)
         GJ  =SE(ISEMXA  )
         YILDGJ=SE(ISEMXA+3)
         IF( YILDGJ .EQ. 1) STBFAG=1
C
         SE(ISEMXA+12)=SE(ISEMXA+1)*LENGTH/GJ
C
         IF( ELAS .EQ. 0 ) THEN
             Q=GJ/LENGTH
         ELSE IF( ELAS .NE. 0 ) THEN
             IF( YILDGJ .EQ. 0 ) THEN
                 Q=GJ/LENGTH
             ELSE IF( YILDGJ .EQ. 1 ) THEN
                 Q=SP*(GJ/LENGTH)
             ENDIF
         ENDIF
      ELSE
         WRITE(6,99)MXA,IELNO
   99    FORMAT(1X,'** INVALID MATERIAL TYPE MXA=',I2,
     &          ' FOR ELEMENT NO. ',I5,' **')
      ENDIF
C
C
      S22=(A2+2*C2+A2Z)/(LENGTH**2)
      S26=(A2+C2)/LENGTH
      S212=(A2Z+C2)/LENGTH
      S33=(BT+2*DT+BTT)/(LENGTH**2)
      S35=(BT+DT)/LENGTH
      S311=(BTT+DT)/LENGTH
C
C---------- SET CONSISTENT MASS GEOMETRIC STIFFNESS COIEFFICIENTS...
       GAZ=  2 * LENGTH / 15.
       GBZ=  GAZ
       GCZ=  LENGTH / 30.
       GDZ=  0.10
       GEZ=  0.10
       GFZ=  1.2 / LENGTH
C
       GAY=  2 * LENGTH / 15.
       GBY=  GAY
       GCY=  LENGTH / 30.
       GDY=  0.10
       GEY=  0.10
       GFY= 1.2 / LENGTH
C
C-----------------------------------------------------------------------
C  ASSEMBLE LOCAL STIFFNESS MATRIX
C-----------------------------------------------------------------------
      S( 1, 1) = H
      S( 1, 7) =-H
      S( 2, 2) = S22
      S( 2, 6) = S26
      S( 2, 8) =-S22
      S( 2,12) = S212
      S( 3, 3) = S33
      S( 3, 5) =-S35
      S( 3, 9) =-S33
      S( 3,11) =-S311
      S( 4, 4) = Q
      S( 4,10) =-Q
      S( 5, 5) = BT
      S( 5, 9) = S35
      S( 5,11) = DT
      S( 6, 6) = A2
      S( 6, 8) =-S26
      S( 6,12) = C2
      S( 7, 7) = H
      S( 8, 8) = S22
      S( 8,12) =-S212
      S( 9, 9) = S33
      S( 9,11) = S311
      S(10,10) = Q
      S(11,11) = BTT
      S(12,12) = A2Z
      S( 5, 3) = S( 3, 5)
      S( 6, 2) = S( 2, 6)
      S( 7, 1) = S( 1, 7)
      S( 8, 2) = S( 2, 8)
      S( 8, 6) = S( 6, 8)
      S( 9, 3) = S( 3, 9)
      S( 9, 5) = S( 5, 9)
      S(10, 4) = S( 4,10)
      S(11, 3) = S( 3,11)
      S(11, 5) = S( 5,11)
      S(11, 9) = S( 9,11)
      S(12, 2) = S( 2,12)
      S(12, 6) = S( 6,12)
      S(12, 8) = S( 8,12)
C     IF (FIRST) CALL WMATRX( S     ,12,12,' S     ')
C
C----- CALCULATE SAT
      DO 60 I=1,12
        DO 60 J=1,12
        SAT(I,J)=0
          DO 60 K=1,12
   60       SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)
C
C----- CALCULATE ASAT
      DO 70 I=1,12
        DO 70 J=1,12
        ASAT(I,J)=0
          DO 70 K=1,12
   70       ASAT(I,J)=ASAT(I,J)+A(I,K)*SAT(K,J)
      IF ( BTEST(IBUG,8) ) CALL WMATRX(ASAT  ,12,12,' ASAT ')
C
C----- DEGREES OF FREEDOM
C       REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF...
      DO 75 I=1,6
        IF ( IDOF(JOINTI,I).NE.IDOF(JOINTJ,I) ) THEN
          LMT(I  )=IDOF(JOINTI,I)
          LMT(I+6)=IDOF(JOINTJ,I)
        ELSE
          LMT(I  )= 0
          LMT(I+6)= 0
        ENDIF
   75   CONTINUE
C
C----- CONVERT TO HALF STORAGE MODE  BY COLUMNS
C
C         1  3  6 10 15 21 28 36 45 55 66 78  NUMBER INDICATES
C            2  5  9 14 20 27 35 44 54 65 77  LOCATION OF DATA
C               4  8 13 19 26 34 43 53 64 76  IN ARRAY STIFF
C                  7 12 18 25 33 42 52 63 75
C                    11 17 24 32 41 51 62 74
C                       16 23 31 40 50 61 73
C                          22 30 39 49 60 72
C                             29 38 48 59 71
C                                37 47 58 70
C                                   46 57 69
C                                      56 68
C                                         67
C
      L=0
      IELDOF=0
      DO 90 I=1,12
        IF (LMT(I).EQ.0 .OR. ASAT(I,I).LE.0) GO TO 90
        IF (LMT(I).EQ.0                    ) GO TO 90
        IELDOF=IELDOF+1
        LM(IELDOF)=LMT(I)
        DO 80 J=I,1,-1
C         IF (LMT(J).EQ.0 .OR. ASAT(J,J).LE.0 ) GO TO 80
          IF (LMT(J).EQ.0                    ) GO TO 80
          L=L+1
          STIFF(L)=ASAT(I,J)
   80   CONTINUE
   90   CONTINUE
      IF ( BTEST(IBUG,7) ) THEN
        WRITE (6,*) 'LMT=',LMT
        WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)
        CALL LMATRX(STIFF,MD,IELDOF,(L+1),'ELEMENT STIFFNESS')
        WRITE (6,*) 'KGDATA:',KGDATA,' PKG:',PKG
      ENDIF
C-----------------------------------------------------------------------
C  ASSEMBLE GEOMETRIC STIFFNESS MATRIX FOR A UNIT LOAD...
C-----------------------------------------------------------------------
      LKG=L
      IF ( PKG.EQ.0 .AND. KGDATA(1).NE.2 ) THEN
        AXIAL=.FALSE.
        REL=IBCLR(REL,6)
        GO TO 180
      ENDIF
C
C------------ DETERMINE IF AXIAL DEFORMATION IS CONSIDERED
C             BIT #6 OF REL IS TRUE IF AXIAL DEFORMATION IS CONSIDERED
      DO 999 I=1,3
        CS=CTS(I,1)
        CE=CTE(I,1)
        IF (ABS(CS).LE. .0001 .AND. ABS(CE).LE. .0001 ) GO TO 999
        IS=IDOF(JOINTI,I)
        IE=IDOF(JOINTJ,I)
        IF (IS.EQ.IE) GO TO 999
          REL=IBSET(REL,6)
  999   CONTINUE
C
        AXIAL=BTEST(REL,6)
      IF ( BTEST(IBUG,7) ) WRITE (6,*) 'AXIAL:',AXIAL
C
      IF (KGDATA(1).NE.0 .AND. KGDATA(2).NE.0 .AND. BTEST(REL,6)) THEN
C
C---------- DETERMINE THE AXIAL LOAD TO BE USED FOR KG...
        IF (KGDATA(1).EQ.1) THEN
            PGEOM = PKG
        ELSE IF (KGDATA(1).EQ.2) THEN
            PGEOM =  0.50*(F(1)-F(7))
        ENDIF
C
C--ASSEMBLE LOCAL GEOMETRIC STIFFNESS MATRIX FOR A UNIT LOAD...
C         ZERO LOCAL KG MATRIX
          DO 998 I=1,12
          DO 998 J=1,12
  998       S(I,J)=0
C
```

```
C---------- LUMPED MASS FORM OF GEOMETRIC STIFFNESS                          ELE07360
          IF (KGDATA(2).EQ.1) THEN                                          ELE07370
               F1=1/LENGTH                                                  ELE07380
               S( 2, 2)=  F1                                                ELE07390
               S( 2, 8)= -F1                                                ELE07400
               S( 8, 2)= -F1                                                ELE07410
               S( 8, 8)=  F1                                                ELE07420
               S( 3, 3)=  F1                                                ELE07430
               S( 3, 9)= -F1                                                ELE07440
               S( 9, 3)= -F1                                                ELE07450
               S( 9, 9)=  F1                                                ELE07460
C                                                                           ELE07470
C---------- CONSISTENT MASS FORM OF GEOMETRIC STIFFNESS                      ELE07480
          ELSE IF (KGDATA(2).EQ.2) THEN                                     ELE07490
               S( 2, 2)=  GFZ                                               ELE07500
               S( 2, 6)=  GDZ                                               ELE07510
               S( 2, 8)= -GFZ                                               ELE07520
               S( 2,12)=  GEZ                                               ELE07530
                S( 6, 2)=  GDZ                                              ELE07540
                S( 6, 6)=  GAZ                                              ELE07550
                S( 6, 8)= -GDZ                                              ELE07560
                S( 6,12)=  GCZ                                              ELE07570
               S( 8, 2)= -GFZ                                               ELE07580
               S( 8, 6)= -GDZ                                               ELE07590
               S( 8, 8)=  GFZ                                               ELE07600
               S( 8,12)= -GEZ                                               ELE07610
                S(12, 2)=  GEZ                                              ELE07620
                S(12, 6)= -GCZ                                              ELE07630
                S(12, 8)= -GEZ                                              ELE07640
                S(12,12)=  GBZ                                              ELE07650
C                                                                           ELE07660
               S( 3, 3)=  GFY                                               ELE07670
               S( 3, 5)= -GDY                                               ELE07680
               S( 3, 9)= -GFY                                               ELE07690
               S( 3,11)= -GEY                                               ELE07700
                S( 5, 3)= -GDY                                              ELE07710
                S( 5, 5)=  GAY                                              ELE07720
                S( 5, 9)=  GDY                                              ELE07730
                S( 5,11)= -GCY                                              ELE07740
               S( 9, 3)= -GFY                                               ELE07750
               S( 9, 5)=  GDY                                               ELE07760
               S( 9, 9)=  GFY                                               ELE07770
               S( 9,11)=  GEY                                               ELE07780
                S(11, 3)= -GEY                                              ELE07790
                S(11, 5)= -GCY                                              ELE07800
                S(11, 9)=  GEY                                              ELE07810
                S(11,11)=  GBY                                              ELE07820
          ENDIF                                                             ELE07830
     IF (BTEST(IBUG,8)) CALL WMATRX( S     ,12,12,' KG -LOCAL COORD ')      ELE07840
C                                                                           ELE07850
C----- CALCULATE SAT                                                        ELE07860
          DO 1060 I=1,12                                                    ELE07870
            DO 1060 J=1,12                                                  ELE07880
            SAT(I,J)=0                                                      ELE07890
              DO 1060 K=1,12                                                ELE07900
 1060           SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)                            ELE07910
C                                                                           ELE07920
C----- CALCULATE ASAT                                                       ELE07930
          DO 1070 I=1,12                                                    ELE07940
            DO 1070 J=1,12                                                  ELE07950
            ASAT(I,J)=0                                                     ELE07960
              DO 1070 K=1,12                                                ELE07970
 1070           ASAT(I,J)=ASAT(I,J)+A(I,K)*SAT(K,J)                        ELE07980
          IF (BTEST(IBUG,8))  CALL WMATRX(ASAT  ,12,12,' A*KG*AT ')         ELE07990
C                                                                           ELE08000
C----- CONVERT TO HALF STORAGE MODE  BY COLUMNS                             ELE08010
          KGDOF=0                                                           ELE08020
          DO 1090 I=1,12                                                    ELE08030
C           IF (LMT(I).EQ.0 .OR. ASAT(I,I).LE.0) GO TO 1090                 ELE08040
            IF (LMT(I).EQ.0                      ) GO TO 1090               ELE08050
            KGDOF=KGDOF+1                                                   ELE08060
            LMKG(KGDOF)=LMT(I)                                              ELE08070
            DO 1080 J=I,1,-1                                                ELE08080
C             IF (LMT(J).EQ.0 .OR. ASAT(J,J).LE.0 ) GO TO 1080              ELE08090
              IF (LMT(J).EQ.0                      ) GO TO 1080             ELE08100
              L=L+1                                                         ELE08110
              STIFF(L)=ASAT(I,J)                                            ELE08120
 1080         CONTINUE                                                      ELE08130
 1090         CONTINUE                                                      ELE08140
     IF ( BTEST(IBUG,7) ) THEN                                              ELE08150
            WRITE (6,*) 'KGDOF',KGDOF,'LMKG =',(LMKG(I),I=1,KGDOF)          ELE08160
            CALL LMATRX(STIFF(LKG+1),MD,KGDOF,(L-LKG),                      ELE08170
     &       'ELEMENT GEOMETRIC STIFFNESS (UNIT LOAD)')                     ELE08180
          ENDIF                                                             ELE08190
     ENDIF                                                                  ELE08200
C                                                                           ELE08210
C----- RELEASE UNUSED STORAGE                                               ELE08220
  180 IREL=156-L                                                            ELE08230
C                                                                           ELE08240
C     PRINT COLUMN DATA                                                     ELE08250
      IF (FIRST .OR. BTEST(IBUG,7) .OR. BTEST(IBUG,8) ) WRITE (6,192)       ELE08260
  191 WRITE(6,193) NAME,IELNO,NODEI,NODEJ,LENGTH,                           ELE08270
     &    VT(1),VT(2),VT(3),XS,-XE,PKG                                      ELE08280
      WRITE(6,194) MATFXA,MATMXA,MATMTA,MATMTB,MATM3A,MATM3B                ELE08290
C                                                                           ELE08300
  192 FORMAT(//' ELEMENT 09,  IE3DBEAM ELEMENT'//                           ELE08310
     &  T22,'#   START   END',3X,'LENGTH',3X,                               ELE08320
     & 11('-'),' Y-AXIS ',                                                  ELE08330
     &  12('-'),'  START DIST    END DIST     PKG')                         ELE08340
  193 FORMAT(1X,A15,2I6,1X,I6,1X,1P,G10.4,0P,                               ELE08350
     &       F9.5,' I',5P,F9.5,' J',F9.5,' K',1P,5S,3G12.4)                 ELE08360
  194 FORMAT(16X,'MATERIAL # : (FX  )-',I2,1X,', (MX  )-',I2,1X,            ELE08370
     &                        ',(MY-A)-',I2,1X,', (MY-B)-',I2,1X,           ELE08380
     &                        ',(MZ-A)-',I2,1X,', (MZ-B)-',I2 )             ELE08390
      RETURN                                                                ELE08400
C------------------------- DETERMINE GEOMETRIC STIFF ----------------------ELE08410
  200 CONTINUE                                                              ELE08420
C- KG IS DETERMINED FOR A UNIT LOAD WITH IOPT=1,                            ELE08430
C- THE ACTUAL LOAD (PGEOM) IS DETERMINED HERE.                              ELE08440
C- KG IS MULTIPLIED BY PGEOM WHEN THE TOTAL STIFFNESS IS ASSEMBLED          ELE08450
C- NEGATIVE PGEOM IS COMPRESSION....                                        ELE08460
      AXIAL=BTEST(REL,6)                                                    ELE08470
C                                                                           ELE08480
C---------- DETERMINE THE AXIAL LOAD TO BE USED FOR KG...                   ELE08490
      IF (AXIAL) THEN                                                       ELE08500
        IF (KGDATA(1).EQ.1) THEN                                            ELE08510
            PGEOM =  PKG                                                    ELE08520
        ELSE IF (KGDATA(1).EQ.2) THEN                                       ELE08530
            PGEOM =  0.50*(F(1)-F(7))                                       ELE08540
        ENDIF                                                               ELE08550
      ENDIF                                                                 ELE08560
      IF (BTEST(IBUG,7)) WRITE (6,*) 'AXIAL ',AXIAL,'  PGEOM:',PGEOM        ELE08570
C                                                                           ELE08580
      RETURN                                                                ELE08590
C------------------------- DETERMINE STIFFNESS ----------------------------ELE08600
  300 CONTINUE                                                              ELE08610
C                                                                           ELE08620
C     STIF1Y,STIF2Y,STIF1Z,STIF2Z,YILDGJ,YILDEA,YILD1Y,YILD2Y,              ELE08630
C     YILD1Z, AND YILD2Z CONTAIN STIFFNESSES, WHICH WERE DETERMINED IN      ELE08640
C     THE LAST CALL TO MA_TLIB                                              ELE08650
C                                                                           ELE08660
C     IF STIF1Y = SE(ISEMTA)    FOR MAT09,MAT11,MAT06                       ELE08670
C        STIF2Y = SE(ISEMTB)    FOR MAT09,MAT11,MAT06                       ELE08680
C        STIF1Z = SE(ISEM3A)    FOR MAT09,MAT11,MAT06                       ELE08690
C        STIF2Z = SE(ISEM3B)    FOR MAT09,MAT11,MAT06                       ELE08700
C        YILDGJ = SE(ISEMXA+3) FOR MAT10                                    ELE08710
C        YILDEA = SE(ISEFXA+3) FOR MAT10                                    ELE08720
C     OR                                                                    ELE08730
C     IF YILD1Y = SE(ISEMTA+3) FOR MAT10                                    ELE08740
C        YILD2Y = SE(ISEMTB+3) FOR MAT10                                    ELE08750
C        YILD1Z = SE(ISEMZA+3) FOR MAT10                                    ELE08760
C        YILD2Z = SE(ISEMZB+3) FOR MAT10                                    ELE08770
C        YILDGJ = SE(ISEMXA+3) FOR MAT10                                    ELE08780
C        YILDEA = SE(ISEFXA+3) FOR MAT10                                    ELE08790
C     THEN,                                                                 ELE08800
C     THE STIFFNESS HAS NOT CHANGED, RETURN                                 ELE08810
C                                                                           ELE08820
      IF( MTA .EQ. 9 .AND. MTB .EQ. 9 ) THEN                                ELE08830
       IF(STIF1Y .NE. SE(ISEMTA) .OR. STIF2Y .NE. SE(ISEMTB))GO TO 3        ELE08840
      ELSE IF( MTA .EQ.11 .AND. MTB .EQ.11 ) THEN                           ELE08850
       IF(STIF1Y .NE. SE(ISEMTA) .OR. STIF2Y .NE. SE(ISEMTB))GO TO 3        ELE08860
      ELSE IF( MTA .EQ. 6 .AND. MTB .EQ. 6 ) THEN                           ELE08870
       IF(STIF1Y .NE. (SE(ISEMTA)/LENGTH) .OR.                              ELE08880
     &    STIF2Y .NE. (SE(ISEMTB)/LENGTH) )GO TO 3                          ELE08890
      ELSE IF( MTA .EQ. 10 .AND. MTB .EQ. 10 ) THEN                         ELE08900
       IF(YILD1Y .NE. SE(ISEMTA+3) .OR. YILD2Y .NE. SE(ISEMTB+3))GO TO 3    ELE08910
      ENDIF                                                                 ELE08920
      IF( MZA .EQ. 9 .AND. MZB .EQ. 9 ) THEN                                ELE08930
       IF(STIF1Z .NE. SE(ISEMZA) .OR. STIF2Z .NE. SE(ISEMZB))GO TO 3        ELE08940
      ELSE IF( MZA .EQ.11 .AND. MZB .EQ.11 ) THEN                           ELE08950
       IF(STIF1Z .NE. SE(ISEMZA) .OR. STIF2Z .NE. SE(ISEMZB))GO TO 3        ELE08960
      ELSE IF( MZA .EQ. 6 .AND. MZB .EQ. 6 ) THEN                           ELE08970
       IF(STIF1Z .NE. (SE(ISEMZA)/LENGTH) .OR.                              ELE08980
     &    STIF2Z .NE. (SE(ISEMZB)/LENGTH) )GO TO 3                          ELE08990
      ELSE IF( MZA .EQ. 10 .AND. MZB .EQ. 10 ) THEN                         ELE09000
       IF(YILD1Z .NE. SE(ISEMZA+3) .OR. YILD2Z .NE. SE(ISEMZB+3))GO TO 3    ELE09010
      ENDIF                                                                 ELE09020
      IF(YILDGJ .NE. SE(ISEMXA+3) ) GO TO 3                                 ELE09030
      IF( MFXA .EQ. 10 ) THEN                                               ELE09040
         IF(YILDEA .NE. SE(ISEFXA+3) ) GO TO 3                              ELE09050
      ELSE IF( MFXA .EQ. 13 ) THEN                                          ELE09060
         IF( H .NE. SE(ISEFXA) ) GO TO 3                                    ELE09070
      ENDIF                                                                 ELE09080
C                                                                           ELE09090
      RETURN                                                                ELE09100
    3 CONTINUE                                                              ELE09110
C                                                                           ELE09120
C---------- SET STIFFNESS COIEFFICIENTS...                                  ELE09130
C                                                                           ELE09140
C  FOR BENDING IN Y AXIS:                                                   ELE09150
      IF( (MTA .EQ. 9 .AND. MTB .EQ. 9) .OR.                                ELE09160
     &    (MTA .EQ.11 .AND. MTB .EQ.11) .OR.                                ELE09170
     &    (MTA .EQ. 6 .AND. MTB .EQ. 6) ) THEN                              ELE09180
         STIF1Y=SE(ISEMTA)                                                  ELE09190
         STIF2Y=SE(ISEMTB)                                                  ELE09200
         IF( MTA .EQ. 9 ) THEN                                              ELE09210
            SA1=SE(ISEMTA+60)                                               ELE09220
            SA2=SE(ISEMTB+60)                                               ELE09230
         ELSE IF( MTA .EQ.11 ) THEN                                         ELE09240
            SA1=SE(ISEMTA+30)                                               ELE09250
            SA2=SE(ISEMTB+30)                                               ELE09260
         ELSE IF( MTA .EQ. 6) THEN                                          ELE09270
            SA1=SE(ISEMTA+47)/LENGTH                                        ELE09280
            SA2=SE(ISEMTB+47)/LENGTH                                        ELE09290
            STIF1Y=STIF1Y/LENGTH                                            ELE09300
            STIF2Y=STIF2Y/LENGTH                                            ELE09310
            EI=SE(ISEMTA+48)                                                ELE09320
            TERM1=(LENGTH**2)/(12*EI*EI)                                    ELE09330
            TERM2=LENGTH/(3*EI)                                             ELE09340
         ENDIF                                                              ELE09350
         IF ( SA1 .EQ. 0 ) THEN                                             ELE09360
               FLX1=0                                                       ELE09370
         ELSE IF ( SA1 .NE. 0 ) THEN                                        ELE09380
               FLX1=(1/STIF1Y)-(1/SA1)                                      ELE09390
         ENDIF                                                              ELE09400
         IF ( SA2 .EQ. 0 ) THEN                                             ELE09410
               FLX2=0                                                       ELE09420
         ELSE IF ( SA2 .NE. 0 ) THEN                                        ELE09430
               FLX2=(1/STIF2Y)-(1/SA2)                                      ELE09440
         ENDIF                                                              ELE09450
         IF ( MTA .EQ. 6 ) THEN                                             ELE09460
            DCOF=TERM1 + TERM2*(FLX1+FLX2) + FLX1*FLX2                      ELE09470
         ELSE IF ( MTA .NE. 6 ) THEN                                        ELE09480
            DCOF=(3/(SA1**2)) + (2/SA1)*(FLX1+FLX2) + FLX1*FLX2             ELE09490
         ENDIF                                                              ELE09500
C                                                                           ELE09510
         DCOF=1/DCOF                                                        ELE09520
C                                                                           ELE09530
         IF ( MTA .EQ. 6 ) THEN                                             ELE09540
           BY =DCOF*(TERM2 + FLX2)                                          ELE09550
           BYY=DCOF*(TERM2 + FLX1)                                          ELE09560
           DY=DCOF*(0.5*TERM2)                                              ELE09570
         ELSE IF ( MTA .NE. 6 ) THEN                                        ELE09580
           BY =DCOF*((2/SA1)+FLX2)                                          ELE09590
           BYY=DCOF*((2/SA2)+FLX1)                                          ELE09600
           DY=DCOF*(1/SA1)                                                  ELE09610
         ENDIF                                                              ELE09620
      ELSE IF( MTA .EQ. 10 .AND. MTB .EQ. 10 ) THEN                         ELE09630
         ELAS=SE(ISEMTA+4)                                                  ELE09640
         SP  =SE(ISEMTA+5)                                                  ELE09650
         EIY =SE(ISEMTA   )                                                 ELE09660
         YILD1Y=SE(ISEMTA+3)                                                ELE09670
         YILD2Y=SE(ISEMTB+3)                                                ELE09680
         IF( YILD1Y .EQ. 1 .OR. YILD2Y .EQ. 1) STBFAG=1                     ELE09690
         IF( ELAS .EQ. 0 ) THEN                                             ELE09700
               BY =4*EIY/LENGTH                                             ELE09710
               BYY=4*EIY/LENGTH                                             ELE09720
               DY =2*EIY/LENGTH                                             ELE09730
         ELSE IF( ELAS .NE. 0 ) THEN                                        ELE09740
               IF( YILD1Y .EQ. 0 .AND. YILD2Y .EQ. 0 ) THEN                 ELE09750
                     BY =4*EIY/LENGTH                                       ELE09760
                     BYY=4*EIY/LENGTH                                       ELE09770
                     DY =2*EIY/LENGTH                                       ELE09780
               ELSE IF( YILD1Y .EQ.1 .AND. YILD2Y .EQ. 0 ) THEN             ELE09790
                     BY =SP*(4*EIY/LENGTH)                                  ELE09800
                     BYY=SP*(4*EIY/LENGTH)+(1-SP)*(3*EIY/LENGTH)            ELE09810
                     DY =SP*(2*EIY/LENGTH)                                  ELE09820
               ELSE IF( YILD1Y .EQ. 0 .AND. YILD2Y .EQ. 1 ) THEN            ELE09830
                     BY =SP*(4*EIY/LENGTH)+(1-SP)*(3*EIY/LENGTH)            ELE09840
                     BYY=SP*(4*EIY/LENGTH)                                  ELE09850
                     DY =SP*(2*EIY/LENGTH)                                  ELE09860
               ELSE IF( YILD1Y .EQ. 1 .AND. YILD2Y .EQ. 1 ) THEN            ELE09870
                     BY =SP*(4*EIY/LENGTH)                                  ELE09880
                     BYY=SP*(4*EIY/LENGTH)                                  ELE09890
                     DY =SP*(2*EIY/LENGTH)                                  ELE09900
               ENDIF                                                        ELE09910
         ENDIF                                                              ELE09920
      ELSE                                                                  ELE09930
         WRITE(6,69)MTA,MTB,IELNO                                           ELE09940
      ENDIF                                                                 ELE09950
C                                                                           ELE09960
C  FOR BENDING IN Z AXIS:                                                   ELE09970
      IF( (MZA .EQ. 9 .AND. MZB .EQ. 9) .OR.                                ELE09980
     &    (MZA .EQ.11 .AND. MZB .EQ.11) .OR.                                ELE09990
     &    (MZA .EQ. 6 .AND. MZB .EQ. 6) ) THEN                              ELE10000
         STIF1Z=SE(ISEM3A)                                                  ELE10010
         STIF2Z=SE(ISEM3B)                                                  ELE10020
         IF( MZA .EQ. 9 ) THEN                                              ELE10030
           SA1=SE(ISEM3A+60)                                                ELE10040
           SA2=SE(ISEM3B+60)                                                ELE10050
         ELSE IF( MZA .EQ.11 ) THEN                                         ELE10060
           SA1=SE(ISEM3A+30)                                                ELE10070
           SA2=SE(ISEM3B+30)                                                ELE10080
         ELSE IF( MZA .EQ. 6) THEN                                          ELE10090
           SA1=SE(ISEM3A+47)/LENGTH                                         ELE10100
           SA2=SE(ISEM3B+47)/LENGTH                                         ELE10110
           STIF1Z=STIF1Z/LENGTH                                             ELE10120
           STIF2Z=STIF2Z/LENGTH                                             ELE10130
           EI=SE(ISEM3A+48)                                                 ELE10140
           TERM1=(LENGTH**2)/(12*EI*EI)                                     ELE10150
           TERM2=LENGTH/(3*EI)                                              ELE10160
         ENDIF                                                              ELE10170
         IF ( SA1 .EQ. 0 ) THEN                                             ELE10180
               FLX1=0                                                       ELE10190
```

```
         ELSE IF ( SA1 .NE. 0 ) THEN
              FLX1=(1/STIF13)-(1/SA1)
         ENDIF
         IF ( SA2 .EQ. 0 ) THEN
              FLX2=0
         ELSE IF ( SA2 .NE. 0 ) THEN
              FLX2=(1/STIF23)-(1/SA2)
         ENDIF
C
C        WRITE(6,*)'1.',STIF13, STIF23
C        WRITE(6,*)'2.',SA1,SA2
C        WRITE(6,*)'3.',FLX1,FLX2
C
         IF ( MZA .EQ. 6 ) THEN
           DCOF=TERM1 + TERM2*(FLX1+FLX2) + FLX1*FLX2
         ELSE IF ( MZA .NE. 6 ) THEN
           DCOF=(3/(SA1**2)) + (2/SA1)*(FLX1+FLX2) + FLX1*FLX2
         ENDIF
C
         DCOF=1/DCOF
C
         IF ( MZA .EQ. 6 ) THEN
           AZ =DCOF*(TERM2 + FLX2)
           AZZ=DCOF*(TERM1 + FLX1)
           CZ=DCOF*(0.5*TERM2)
         ELSE IF ( MZA .NE. 6 ) THEN
           AZ =DCOF*((2/SA1)+FLX2)
           AZZ=DCOF*((2/SA2)+FLX1)
           CZ=DCOF*(1/SA1)
         ENDIF
       ELSE IF( MZA .EQ. 10 .AND. MZB .EQ. 10 ) THEN
         ELAS=SE(ISEMZA+4)
         SP  =SE(ISEMZA+5)
         EIZ =SE(ISEMZA  )
         YILD1Z=SE(ISEMZA+3)
         YILD2Z=SE(ISEMZB+3)
         IF( YILD1Z .EQ. 1 .OR. YILD2Z .EQ. 1) STBFAG=1
         IF( ELAS .EQ. 0 ) THEN
              AZ =4*EIZ/LENGTH
              AZZ=4*EIZ/LENGTH
              CZ =2*EIZ/LENGTH
         ELSE IF( ELAS .NE. 0 ) THEN
              IF( YILD1Z .EQ. 0 .AND. YILD2Z .EQ. 0 ) THEN
                   AZ =4*EIZ/LENGTH
                   AZZ=4*EIZ/LENGTH
                   CZ =2*EIZ/LENGTH
              ELSE IF( YILD1Z .EQ. 1 .AND. YILD2Z .EQ. 0 ) THEN
                   AZ =SP*(4*EIZ/LENGTH)
                   AZZ=SP*(4*EIZ/LENGTH)+(1-SP)*(3*EIZ/LENGTH)
                   CZ =SP*(2*EIZ/LENGTH)
              ELSE IF( YILD1Z .EQ. 0 .AND. YILD2Z .EQ. 1 ) THEN
                   AZ =SP*(4*EIZ/LENGTH)+(1-SP)*(3*EIZ/LENGTH)
                   AZZ=SP*(4*EIZ/LENGTH)
                   CZ =SP*(2*EIZ/LENGTH)
              ELSE IF( YILD1Z .EQ. 1 .AND. YILD2Z .EQ. 1 ) THEN
                   AZ =SP*(4*EIZ/LENGTH)
                   AZZ=SP*(4*EIZ/LENGTH)
                   CZ =SP*(2*EIZ/LENGTH)
              ENDIF
         ENDIF
       ELSE
         WRITE(6,79)MZA,MZB,IELNO
       ENDIF
C
C  FOR AXIAL IN X AXIS:
       IF( MFXA .EQ. 10 ) THEN
         ELAS=SE(ISEFXA+4)
         SP  =SE(ISEFXA+5)
         EA  =SE(ISEFXA  )
         YILDEA=SE(ISEFXA+3)
         IF( YILDEA .EQ. 1 ) STBFAG=1
         IF( ELAS .EQ. 0 ) THEN
              H=EA/LENGTH
         ELSE IF( ELAS .NE. 0 ) THEN
              IF( YILDEA .EQ. 0 ) THEN
                   H=EA/LENGTH
              ELSE IF( YILDEA .EQ. 1 ) THEN
                   H=SP*(EA/LENGTH)
              ENDIF
         ENDIF
       ELSE IF( MFXA .EQ. 13 ) THEN
         H=SE(ISEFXA)
       ELSE IF( MFXA .NE. 10 .OR. MFXA .NE. 13 ) THEN
         WRITE(6,89)MFXA,IELNO
       ENDIF
C
C
C  FOR TORSION IN X AXIS:
       IF( MXA .EQ. 10 ) THEN
         ELAS=SE(ISEMXA+4)
         SP  =SE(ISEMXA+5)
         GJ  =SE(ISEMXA  )
         YILDGJ=SE(ISEMXA+3)
         IF( YILDGJ .EQ. 1 ) STBFAG=1
         IF( ELAS .EQ. 0 ) THEN
              Q=GJ/LENGTH
         ELSE IF( ELAS .NE. 0 ) THEN
            IF( YILDGJ .EQ. 0 ) THEN
                 Q=GJ/LENGTH
            ELSE IF( YILDGJ .EQ. 1 ) THEN
                 Q=SP*(GJ/LENGTH)
            ENDIF
         ENDIF
       ELSE
         WRITE(6,99)MXA,IELNO
       ENDIF
C
       S22=(AZ+2*CZ+AZZ)/(LENGTH**2)
       S26=(AZ+CZ)/LENGTH
       S212=(AZZ+CZ)/LENGTH
       S33=(BY+2*DY+BYY)/(LENGTH**2)
       S35=(BY+DY)/LENGTH
       S311=(BYY+DY)/LENGTH
C
C-----------------------------------------------------------------------
C  ASSEMBLE LOCAL STIFFNESS MATRIX
C-----------------------------------------------------------------------
       DO 894 I=1,12
       DO 894 J=1,12
  894    S(I,J)=0
C
       S( 1, 1) = H
       S( 1, 7) =-H
       S( 2, 2) = S22
       S( 2, 6) = S26
       S( 2, 8) =-S22
       S( 2,12) = S212
       S( 3, 3) = S33
       S( 3, 5) =-S35
       S( 3, 9) =-S33
       S( 3,11) =-S311
       S( 4, 4) = Q
       S( 4,10) =-Q
       S( 5, 5) = BY
       S( 5, 9) = S35
       S( 5,11) = DY
       S( 6, 6) = AZ
       S( 6, 8) =-S26
       S( 6,12) = CZ
       S( 7, 7) = H
       S( 8, 8) = S22
       S( 8,12) =-S212
       S( 9, 9) = S33
       S( 9,11) = S311
       S(10,10) = Q
       S(11,11) = BYY
       S(12,12) = AZZ
       S( 5, 3) = S( 3, 5)
       S( 6, 2) = S( 2, 6)
       S( 7, 1) = S( 1, 7)
       S( 8, 2) = S( 2, 8)
       S( 8, 6) = S( 6, 8)
       S( 9, 3) = S( 3, 9)
       S( 9, 5) = S( 5, 9)
       S(10, 4) = S( 4,10)
       S(11, 3) = S( 3,11)
       S(11, 5) = S( 5,11)
       S(11, 9) = S( 9,11)
       S(12, 2) = S( 2,12)
       S(12, 6) = S( 6,12)
       S(12, 8) = S( 8,12)
C      IF (FIRST) CALL WMATRX( S    ,12,12,' S    ')
C
C----- CALCULATE SAT
       DO 62 I=1,12
         DO 62 J=1,12
         SAT(I,J)=0
           DO 62 K=1,12
   62        SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)
C
C----- CALCULATE ASAT
       DO 72 I=1,12
         DO 72 J=1,12
         ASAT(I,J)=0
           DO 72 K=1,12
   72        ASAT(I,J)=ASAT(I,J)+A(I,K)*SAT(K,J)
       IF ( BTEST(IBUG,8) ) CALL WMATRX(ASAT  ,12,12,' ASAT ')
C
C----- DEGREES OF FREEDOM
C      REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF...
       DO 77 I=1,6
         IF ( IDOF(JOINTI,I).NE.IDOF(JOINTJ,I) ) THEN
           LMT(I  )=IDOF(JOINTI,I)
           LMT(I+6)=IDOF(JOINTJ,I)
         ELSE
           LMT(I  )= 0
           LMT(I+6)= 0
         ENDIF
   77  CONTINUE
C
C----- CONVERT TO HALF STORAGE MODE  BY COLUMNS
C
       L=0
       IELDOF=0
       DO 92 I=1,12
C        IF (LMT(I).EQ.0 .OR. ASAT(I,I).LE.0) GO TO 92
         IF (LMT(I).EQ.0                    ) GO TO 92
         IELDOF=IELDOF+1
         LM(IELDOF)=LMT(I)
         DO 82 J=I,1,-1
C          IF (LMT(J).EQ.0 .OR. ASAT(J,J).LE.0 ) GO TO 82
           IF (LMT(J).EQ.0                    ) GO TO 82
           L=L+1
           STIFF(L)=ASAT(I,J)
   82    CONTINUE
   92  CONTINUE
       IF ( BTEST(IBUG,7) ) THEN
         WRITE (6,*) 'LMT=',LMT
         WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)
         CALL LMATRX(STIFF,MD,IELDOF,(L+1),'ELEMENT STIFFNESS')
         WRITE (6,*) 'KGDATA1',KGDATA,' PKG=',PKG
       ENDIF
C
       RETURN
C------------------------- DETERMINE INCRIMENTAL FORCES ----------------
  400 CONTINUE
        IF (PRINT.AND.FIRST) WRITE (6,491) STEP1D
C
C------- CALL BMLOA TO DETERMINE THE FIXED END FORCES
       IF (MAXELD.GT.0) THEN
         CALL BMLOA(IELNO,FEMFLG,LENGTH,REL,MAXELD,NELD,NLOAD,
     &   FEM,IELD,ELD)
       ELSE
         FEMFLG=.FALSE.
       ENDIF
C
C------ LOOP FOR EACH LOAD
       DO 430 LOAD=1,NLOAD
C
C------ TRANSFORM FRAME DISPL INTO LOCAL DISPL
        DO 410 I=1,3
         R(I  )=0.
         R(I+3)=0.
         R(I+6)=0.
         R(I+9)=0.
         DO 410 J=1,3
           D1=DISPL(IDOF(JOINTI,J  ),LOAD)
           D2=DISPL(IDOF(JOINTI,J+3),LOAD)
           D3=DISPL(IDOF(JOINTJ,J  ),LOAD)
           D4=DISPL(IDOF(JOINTJ,J+3),LOAD)
         R(I  )=R(I  )+CTS(J,I)*D1 + BS(J,I)*D2
         R(I+3)=R(I+3)+CTS(J,I)*D2
         R(I+6)=R(I+6)+CTE(J,I)*D3 + BE(J,I)*D4
  410    R(I+9)=R(I+9)+CTE(J,I)*D4
C
C------ INCRIMENTAL FORCES
       F( 1)=  H*(R( 1)-R( 7))
       F( 2)= S22*(R( 2)-R( 8))+S26*R( 6)+S212*R(12)
       F( 3)= S33*(R( 3)-R( 9))-S35*R( 5)-S311*R(11)
       F( 4)=   Q*(R( 4)-R(10))
       F( 5)=-S35*(R( 3)-R( 9))+BY*R( 5)+DY*R(11)
       F( 6)= S26*(R( 2)-R( 8))+AZ*R( 6)+CZ*R(12)
       F( 7)=  -H*(R( 1)-R( 7))
       F( 8)=-S22*(R( 2)-R( 8))-S26*R( 6)-S212*R(12)
       F( 9)=-S33*(R( 3)-R( 9))+S35*R( 5)+S311*R(11)
       F(10)=  -Q*(R( 4)-R(10))
       F(11)=-S311*(R( 3)-R( 9))+ DY*R( 5)+BYY*R(11)
       F(12)= S212*(R( 2)-R( 8))+ CZ*R( 6)+AZZ*R(12)
C
C------ RELATIVE ROTATION FOR BENDINGS
         R(5)=R(5)-(R(3)-R(9))/LENGTH
         R(6)=R(6)-(R(8)-R(2))/LENGTH
         R(11)=R(11)-(R(3)-R(9))/LENGTH
         R(12)=R(12)-(R(8)-R(2))/LENGTH
C
C      IF( (MTA .EQ. 9 .OR. MTA .EQ. 11) .AND.
C     &    (MTB .EQ. 9 .OR. MTB .EQ. 11) ) THEN
C           F( 5)=STIF1T*R(5)
C           F(11)=STIF2T*R(11)
C           F( 9)=(F( 5)+F(11))/LENGTH
C           F( 3)=-F( 9)
C      ENDIF
C
C      IF( (MZA .EQ. 9 .OR. MZA .EQ. 11) .AND.
C     &    (MZB .EQ. 9 .OR. MZB .EQ. 11) ) THEN
C           F( 6)=STIF1Z*R(6)
C           F(12)=STIF2Z*R(12)
C           F( 2)=(F( 5)+F(11))/LENGTH
C           F( 8)=-F( 2)
C      ENDIF
C
C
C------ SUBTRACT OF FIXED END FORCES
       IF (FEMFLG) THEN
```

```
      DO 420 I=1,12                                                     ELE13040
         IF( I .EQ. 5 .OR. I .EQ. 11 .OR.                               ELE13050
     &       I .EQ. 3 .OR. I .EQ.  9 ) THEN                             ELE13060
         F(I)=F(I)-FEM(I,LOAD)                                          ELE13070
C           IF( MYA .NE. 10 .OR. MYB .NE. 10 ) GO TO 420                ELE13080
         ENDIF                                                          ELE13090
         IF( I .EQ. 6 .OR. I .EQ. 12 .OR.                               ELE13100
     &       I .EQ. 2 .OR. I .EQ.  8 ) THEN                             ELE13110
C           IF( MZA .NE. 10 .OR. MZB .NE. 10 ) GO TO 420                ELE13120
         F(I)=F(I)-FEM(I,LOAD)                                          ELE13130
         ENDIF                                                          ELE13140
  420 CONTINUE                                                          ELE13150
      ENDIF                                                             ELE13160
C                                                                       ELE13170
C------ DETERMINE MAXIMUN VALUES FOR MULTIPLE LOAD CASES                ELE13180
      IF (NLOAD.GT.1) THEN                                              ELE13190
        DO 425 I=1,12                                                   ELE13200
          IF ( ABS(F(I)).GT.ABS(FMAX(I*12-12+I)) ) THEN                 ELE13210
            DO 426 J=0,11                                               ELE13220
  426         FMAX(I*12-11+J)=F(J+1)                                    ELE13230
          ENDIF                                                         ELE13240
  425   CONTINUE                                                        ELE13250
      ENDIF                                                             ELE13260
C                                                                       ELE13270
C------ PRINT FORCES AND DEFORMATIONS                                   ELE13280
C     IF (BTEST(IBUG,0).AND.PRINT)                                      ELE13290
      IF (         PRINT)                                               ELE13300
     &                    WRITE (6,493) IELNO,LOAD,NODEI,(R(J),J=1,6),  ELE13310
     &                                  NODEJ,(R(J),J=7,12)             ELE13320
      IF (         PRINT) WRITE (6,492) IELNO,LOAD,NODEI,(F(J),J=1,6),  ELE13330
     &                                  NODEJ,(F(J),J=7,12),            ELE13340
     &                                  STEFAG                          ELE13350
C                                                                       ELE13360
  430 CONTINUE                                                          ELE13370
C                                                                       ELE13380
C ------THIS IS FOR SOL05A USE ONLY                                     ELE13390
      IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                         ELE13400
        DO 431 I=1,12                                                   ELE13410
  431   RSAF(I)=F(I)                                                    ELE13420
      ENDIF                                                             ELE13430
C                                                                       ELE13440
C ------CHECK STIFFNESS                                                ELE13450
C       THIS IS PERFORMED ONLY FOR INELASTIC ANALYSIS                   ELE13460
        P(1)=FT( 5) + F( 5)                                             ELE13470
        P(2)=FT(11)+  F(11)                                             ELE13480
        P(3)=FT( 6) + F( 6)                                             ELE13490
        P(4)=FT(12)+  F(12)                                             ELE13500
        P(5)=FT(10)+  F(10)                                             ELE13510
        P(6)=FT( 7)+  F( 7)                                             ELE13520
      IF(.NOT. ELSTIC) THEN                                             ELE13530
C                                                                       ELE13540
C.....CHECK INTERACTIVE STRENGTH CRITERIA                               ELE13550
        PT  = SE(ISEFXA+11)                                             ELE13560
        TMPY= SE(ISEMYA+11)                                             ELE13570
        TMPZ= SE(ISEMZA+11)                                             ELE13580
        TMPYA=SE(ISEMYA+1)                                              ELE13590
        TMPYB=SE(ISEMYB+1)                                              ELE13600
        TMPZA=SE(ISEMZA+1)                                              ELE13610
        TMPZB=SE(ISEMZB+1)                                              ELE13620
      IF ( IAFAG .EQ. 1 ) THEN                                          ELE13630
        CALL ITERS(P(6),P(1),P(2),TMPT,PT,TMPYA,TMPYB,                  ELE13640
     &                    P(3),P(4),TMPZ,   TMPZA,TMPZB,IAFAG)          ELE13650
C       IF( IAFAG .EQ. 1 ) GO TO 377                                    ELE13660
        IF( IAFAG .EQ. 2 ) THEN                                         ELE13670
          SE(ISEMYA+1)=ABS(P(1))*SE(ISEMYA+8)                           ELE13680
          SE(ISEMZA+1)=ABS(P(3))*SE(ISEMZA+8)                           ELE13690
          SE(ISEMYA+2)=-SE(ISEMYA+1)                                    ELE13700
          SE(ISEMZA+2)=-SE(ISEMZA+1)                                    ELE13710
        ELSE IF( IAFAG .EQ. 3 ) THEN                                    ELE13720
          SE(ISEMYB+1)=ABS(P(2))*SE(ISEMYB+8)                           ELE13730
          SE(ISEMZB+1)=ABS(P(4))*SE(ISEMZB+8)                           ELE13740
          SE(ISEMYB+2)=-SE(ISEMYB+1)                                    ELE13750
          SE(ISEMZB+2)=-SE(ISEMZB+1)                                    ELE13760
        ELSE IF( IAFAG .EQ. 4 ) THEN                                    ELE13770
          SE(ISEMYA+1)=TMPYA*SE(ISEMYA+8)                               ELE13780
          SE(ISEMZA+1)=TMPZA*SE(ISEMZA+8)                               ELE13790
          SE(ISEMYB+1)=TMPYB*SE(ISEMYB+8)                               ELE13800
          SE(ISEMZB+1)=TMPZB*SE(ISEMZB+8)                               ELE13810
          SE(ISEMYA+2)=-SE(ISEMYA+1)                                    ELE13820
          SE(ISEMZA+2)=-SE(ISEMZA+1)                                    ELE13830
          SE(ISEMYB+2)=-SE(ISEMYB+1)                                    ELE13840
          SE(ISEMZB+2)=-SE(ISEMZB+1)                                    ELE13850
        ENDIF                                                           ELE13860
C                                                                       ELE13870
C... ASSIGN AXIAL RESIDUAL STRENGTH                                     ELE13880
C                                                                       ELE13890
        IF( SE(ISEFXA+1) .EQ. -SE(ISEFXA+2) .AND. IAFAG .NE. 1 ) THEN   ELE13900
              SE(ISEFXA+1)=SE(ISEFXA+1)                                 ELE13910
              SE(ISEFXA+2)=-ABS(P(6))*SE(ISEFXA+8)                      ELE13920
              WRITE(6,705)IELNO,JSTEP,IAFAG                             ELE13930
  705         FORMAT(1X,'* ELEMENT ',I5,' LOCAL BUCKLING OCCURRED ',    ELE13940
     &                  'AT STEP ',I5,' IAFAG=',I5,' *')                ELE13950
        ENDIF                                                           ELE13960
      ELSE IF( IAFAG .EQ. 2 ) THEN                                      ELE13970
        CALL ITERS(P(6),P(1),P(2),TMPT,PT,TMPYA,TMPYB,                  ELE13980
     &                    P(3),P(4),TMPZ,   TMPZA,TMPZB,IAFAG)          ELE13990
        IF( IAFAG .EQ. 4 ) THEN                                         ELE14000
          SE(ISEMYB+1)=ABS(P(2))*SE(ISEMYB+8)                           ELE14010
          SE(ISEMZB+1)=ABS(P(4))*SE(ISEMZB+8)                           ELE14020
          SE(ISEMYB+2)=-SE(ISEMYB+1)                                    ELE14030
          SE(ISEMZB+2)=-SE(ISEMZB+1)                                    ELE14040
          WRITE(6,705)IELNO,JSTEP,IAFAG                                 ELE14050
        ENDIF                                                           ELE14060
      ELSE IF( IAFAG .EQ. 3 ) THEN                                      ELE14070
        CALL ITERS(P(6),P(1),P(2),TMPT,PT,TMPYA,TMPYB,                  ELE14080
     &                    P(3),P(4),TMPZ,   TMPZA,TMPZB,IAFAG)          ELE14090
        IF( IAFAG .EQ. 4 ) THEN                                         ELE14100
          SE(ISEMYA+1)=ABS(P(1))*SE(ISEMYA+8)                           ELE14110
          SE(ISEMZA+1)=ABS(P(3))*SE(ISEMZA+8)                           ELE14120
          SE(ISEMYA+2)=-SE(ISEMYA+1)                                    ELE14130
          SE(ISEMZA+2)=-SE(ISEMZA+1)                                    ELE14140
          WRITE(6,705)IELNO,JSTEP,IAFAG                                 ELE14150
        ENDIF                                                           ELE14160
      ELSE IF( IAFAG .EQ. 5 ) THEN                                      ELE14170
        CALL ITERS2(P(6),P(1),P(2),TMPT,PT,TMPYA,TMPYB,                 ELE14180
     &                    P(3),P(4),TMPZ,   TMPZA,TMPZB,IAFAG)          ELE14190
        IF( IAFAG .GT. 5 ) THEN                                         ELE14200
          WRITE(6,706)IELNO,JSTEP,IAFAG                                 ELE14210
  706       FORMAT(1X,'* ELEMENT ',I5,' MODEL_OVERSTRENGTH ',           ELE14220
     &                  'AT STEP ',I5,' IAFAG=',I5,' *')                ELE14230
        ENDIF                                                           ELE14240
      ENDIF                                                             ELE14250
C                                                                       ELE14260
C     ----- TRANSFER PERMENANT HYSTERESIS DATA TO TEMPERARY STORAGE     ELE14270
  377   ISE2=ISE/2                                                      ELE14280
        DO 522 I=1,ISE2                                                 ELE14290
  522     SE(I+ISE2)=SE(I)                                              ELE14300
C                                                                       ELE14310
C---- CALCULATE INELASTIC ROTATION                                      ELE14320
      IF( (MYA .EQ.  9 .AND. MYB .EQ.  9) .OR.                          ELE14330
     &    (MYA .EQ. 11 .AND. MYB .EQ. 11) .OR.                          ELE14340
     &    (MYA .EQ.  6 .AND. MYB .EQ.  6) ) THEN                        ELE14350
        STIF1Y=SE(ISEMYA)                                               ELE14360
        STIF2Y=SE(ISEMYB)                                               ELE14370
        IF( MYA .EQ. 9 ) THEN                                           ELE14380
          SA1=SE(ISEMYA+60)                                             ELE14390
          SA2=SE(ISEMYB+60)                                             ELE14400
        ELSE IF( MYA .EQ.11 ) THEN                                      ELE14410
          SA1=SE(ISEMYA+30)                                             ELE14420
          SA2=SE(ISEMYB+30)                                             ELE14430
        ELSE IF( MYA .EQ.  6 ) THEN                                     ELE14440
          SA1=SE(ISEMYA+47)/LENGTH                                      ELE14450
          SA2=SE(ISEMYB+47)/LENGTH                                      ELE14460
          STIF1Y=STIF1Y/LENGTH                                          ELE14470
          STIF2Y=STIF2Y/LENGTH                                          ELE14480
        ENDIF                                                           ELE14490
        IF ( SA1 .EQ. 0 ) THEN                                          ELE14500
             FLX1=0                                                     ELE14510
        ELSE IF ( SA1 .NE. 0 ) THEN                                     ELE14520
             FLX1=(1/STIF1Y)-(1/SA1)                                    ELE14530
        ENDIF                                                           ELE14540
        IF ( SA2 .EQ. 0 ) THEN                                          ELE14550
             FLX2=0                                                     ELE14560
        ELSE IF ( SA2 .NE. 0 ) THEN                                     ELE14570
             FLX2=(1/STIF2Y)-(1/SA2)                                    ELE14580
        ENDIF                                                           ELE14590
C       FLX1=(1/STIF1Y)-(1/SA1)                                         ELE14600
C       FLX2=(1/STIF2Y)-(1/SA2)                                         ELE14610
        ROTYA=ROTYA + FLX1*F(5)                                         ELE14620
        ROTYB=ROTYB + FLX2*F(11)                                        ELE14630
      ENDIF                                                             ELE14640
C                                                                       ELE14650
      IF( (MZA .EQ.  9 .AND. MZB .EQ.  9) .OR.                          ELE14660
     &    (MZA .EQ. 11 .AND. MZB .EQ. 11) .OR.                          ELE14670
     &    (MZA .EQ.  6 .AND. MZB .EQ.  6) ) THEN                        ELE14680
        STIF1Z=SE(ISEMZA)                                               ELE14690
        STIF2Z=SE(ISEMZB)                                               ELE14700
        IF( MZA .EQ. 9 ) THEN                                           ELE14710
          SA1=SE(ISEMZA+60)                                             ELE14720
          SA2=SE(ISEMZB+60)                                             ELE14730
        ELSE IF( MZA .EQ.11 ) THEN                                      ELE14740
          SA1=SE(ISEMZA+30)                                             ELE14750
          SA2=SE(ISEMZB+30)                                             ELE14760
        ELSE IF( MZA .EQ.  6 ) THEN                                     ELE14770
          SA1=SE(ISEMZA+47)/LENGTH                                      ELE14780
          SA2=SE(ISEMZB+47)/LENGTH                                      ELE14790
          STIF1Z=STIF1Z/LENGTH                                          ELE14800
          STIF2Z=STIF2Z/LENGTH                                          ELE14810
        ENDIF                                                           ELE14820
        IF ( SA1 .EQ. 0 ) THEN                                          ELE14830
             FLX1=0                                                     ELE14840
        ELSE IF ( SA1 .NE. 0 ) THEN                                     ELE14850
             FLX1=(1/STIF1Z)-(1/SA1)                                    ELE14860
        ENDIF                                                           ELE14870
        IF ( SA2 .EQ. 0 ) THEN                                          ELE14880
             FLX2=0                                                     ELE14890
        ELSE IF ( SA2 .NE. 0 ) THEN                                     ELE14900
             FLX2=(1/STIF2Z)-(1/SA2)                                    ELE14910
        ENDIF                                                           ELE14920
C       FLX1=(1/STIF1Z)-(1/SA1)                                         ELE14930
C       FLX2=(1/STIF2Z)-(1/SA2)                                         ELE14940
        ROTZA=ROTZA + FLX1*F(6)                                         ELE14950
        ROTZB=ROTZB + FLX2*F(12)                                        ELE14960
      ENDIF                                                             ELE14970
C                                                                       ELE14980
C                                                                       ELE14990
C---- BENDING IN A END Y DIRECTION                                      ELE15000
C     SET UP TEMTERARY - TOTAL LOADS,DISP'S                             ELE15010
        PL=FT(5)                                                        ELE15020
        D =RT(5)+R(5)                                                   ELE15030
        DL=RT(5)                                                        ELE15040
        IF( MYA .EQ. 9 ) THEN                                           ELE15050
          D=RIETA + P(1)/SE(ISEMYA+60)                                  ELE15060
          DL=ROIETA + PL/SE(ISEMYA+60)                                  ELE15070
        ELSE IF( MYA .EQ. 11) THEN                                      ELE15080
          D=RIETA + P(1)/SE(ISEMYA+30)                                  ELE15090
          DL=ROIETA + PL/SE(ISEMYA+30)                                  ELE15100
        ELSE IF( MYA .EQ.  6) THEN                                      ELE15110
          IF( SE(ISEMYA+47) .NE. 0 ) THEN                               ELE15120
             D=(RIETA/LENGTH) + P(1)/SE(ISEMYA+47)                      ELE15130
             DL=(ROIETA/LENGTH) + PL/SE(ISEMYA+47)                      ELE15140
          ENDIF                                                         ELE15150
        ENDIF                                                           ELE15160
        MOPT=3                                                          ELE15170
        CALL MATLIB                                                     ELE15180
     &(MOPT,LSTTTP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,1),EXCR(1,1), ELE15190
     &   P(1)  ,PL    ,D     ,DL    ,V     ,VL    ,LELEM ,LMAT  ,       ELE15200
     &   MATMYA,MATTTP,SE(ISE2+ISEMYA),IELNO,DAMAGE)                    ELE15210
        IF( MATTTP .EQ. 9 ) THEN                                        ELE15220
          IF(SE(ISE2+ISEMYA) .NE.SE(ISEMYA) ) NEWK=.TRUE.               ELE15230
          ROIETA=D-P(1)/SE(ISEMYA+60)                                   ELE15240
        ELSE IF( MATTTP .EQ.11 ) THEN                                   ELE15250
          IF(SE(ISE2+ISEMYA) .NE.SE(ISEMYA) ) NEWK=.TRUE.               ELE15260
          ROIETA=D-P(1)/SE(ISEMYA+30)                                   ELE15270
        ELSE IF( MATTTP .EQ. 6 ) THEN                                   ELE15280
          IF(SE(ISE2+ISEMYA) .NE.SE(ISEMYA) ) NEWK=.TRUE.               ELE15290
          IF(SE(ISEMYA+47) .NE. 0 ) THEN                                ELE15300
            ROIETA=(D-P(1)/SE(ISEMYA+47))*LENGTH                        ELE15310
          ENDIF                                                         ELE15320
        ELSE IF( MATTTP .EQ. 10 ) THEN                                  ELE15330
          IF(SE(ISE2+ISEMYA+3) .NE.SE(ISEMYA+3) ) NEWK=.TRUE.           ELE15340
        ENDIF                                                           ELE15350
C---- BENDING IN B END Y DIRECTION                                      ELE15360
C     SET UP TEMTERARY - TOTAL LOADS,DISP'S                             ELE15370
        PL=FT(11)                                                       ELE15380
        D =RT(11)+R(11)                                                 ELE15390
        DL=RT(11)                                                       ELE15400
        IF( MYB .EQ. 9 ) THEN                                           ELE15410
          D=RIETB + P(2)/SE(ISEMYB+60)                                  ELE15420
          DL=ROIETB + PL/SE(ISEMYB+60)                                  ELE15430
        ELSE IF( MYB .EQ. 11) THEN                                      ELE15440
          D=RIETB + P(2)/SE(ISEMYB+30)                                  ELE15450
          DL=ROIETB + PL/SE(ISEMYB+30)                                  ELE15460
        ELSE IF( MYB .EQ.  6) THEN                                      ELE15470
          IF( SE(ISEMYB+47) .NE. 0 ) THEN                               ELE15480
             D=(RIETB/LENGTH) + P(2)/SE(ISEMYB+47)                      ELE15490
             DL=(ROIETB/LENGTH) + PL/SE(ISEMYB+47)                      ELE15500
          ENDIF                                                         ELE15510
        ENDIF                                                           ELE15520
        CALL MATLIB                                                     ELE15530
     &(MOPT,LSTTTP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,2),EXCR(1,2), ELE15540
     &   P(2)  ,PL    ,D     ,DL    ,V     ,VL    ,LELEM ,LMAT  ,       ELE15550
     &   MATMYB,MATTTP,SE(ISE2+ISEMYB),IELNO,DAMAGE)                    ELE15560
        IF( MATTTP .EQ. 9 ) THEN                                        ELE15570
          IF(SE(ISE2+ISEMYB) .NE.SE(ISEMYB) ) NEWK=.TRUE.               ELE15580
          ROIETB=D-P(2)/SE(ISEMYB+60)                                   ELE15590
        ELSE IF( MATTTP .EQ.11 ) THEN                                   ELE15600
          IF(SE(ISE2+ISEMYB) .NE.SE(ISEMYB) ) NEWK=.TRUE.               ELE15610
          ROIETB=D-P(2)/SE(ISEMYB+30)                                   ELE15620
        ELSE IF( MATTTP .EQ. 6 ) THEN                                   ELE15630
          IF(SE(ISE2+ISEMYB) .NE.SE(ISEMYB) ) NEWK=.TRUE.               ELE15640
          IF(SE(ISEMYB+47) .NE. 0 ) THEN                                ELE15650
            ROIETB=(D-P(2)/SE(ISEMYB+47))*LENGTH                        ELE15660
          ENDIF                                                         ELE15670
        ELSE IF( MATTTP .EQ. 10 ) THEN                                  ELE15680
          IF(SE(ISE2+ISEMYB+3) .NE.SE(ISEMYB+3) ) NEWK=.TRUE.           ELE15690
        ENDIF                                                           ELE15700
C---- BENDING IN A END Z DIRECTION                                      ELE15710
C     SET UP TEMTERARY - TOTAL LOADS,DISP'S                             ELE15720
        PL=FT(6)                                                        ELE15730
        D =RT(6)+R(6)                                                   ELE15740
        DL=RT(6)                                                        ELE15750
        IF( MZA .EQ. 9 ) THEN                                           ELE15760
          D=RIEZA + P(3)/SE(ISEMZA+60)                                  ELE15770
          DL=ROIEZA + PL/SE(ISEMZA+60)                                  ELE15780
        ELSE IF( MZA .EQ. 11) THEN                                      ELE15790
          D=RIEZA + P(3)/SE(ISEMZA+30)                                  ELE15800
          DL=ROIEZA + PL/SE(ISEMZA+30)                                  ELE15810
        ELSE IF( MZA .EQ.  6) THEN                                      ELE15820
          IF( SE(ISEMZA+47) .NE. 0 ) THEN                               ELE15830
             D=(RIEZA/LENGTH) + P(3)/SE(ISEMZA+47)                      ELE15840
             DL=(ROIEZA/LENGTH) + PL/SE(ISEMZA+47)                      ELE15850
          ENDIF                                                         ELE15860
        ENDIF                                                           ELE15870
```

```
      CALL MATLIB                                                       ELE15880
     &(MOPT,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,3),EXCR(1,3), ELE15890
     &   P(3)  ,PL    ,D     ,DL    ,V     ,VL    ,LELEM ,LMAT  ,      ELE15900
     &   MATM3A,MATTYP,SE(ISE2+ISEM3A),IELNO,DAMAGE)                    ELE15910
        IF( MATTYP .EQ. 9 ) THEN                                        ELE15920
           IF(SE(ISE2+ISEM3A) .NE.SE(ISEM3A) ) NEWK=.TRUE.              ELE15930
           ROIE3A=D-P(3)/SE(ISEM3A+60)                                  ELE15940
        ELSE IF( MATTYP .EQ.11 ) THEN                                   ELE15950
           IF(SE(ISE2+ISEM3A) .NE.SE(ISEM3A) ) NEWK=.TRUE.              ELE15960
           ROIE3A=D-P(3)/SE(ISEM3A+30)                                  ELE15970
        ELSE IF( MATTYP .EQ. 6 ) THEN                                   ELE15980
           IF(SE(ISE2+ISEM3A) .NE.SE(ISEM3A) ) NEWK=.TRUE.              ELE15990
           IF( SE(ISEM3A+47) .NE. 0 ) THEN                              ELE16000
              ROIE3A=(D-P(3)/SE(ISEM3A+47))*LENGTH                      ELE16010
           ENDIF                                                        ELE16020
        ELSE IF( MATTYP .EQ. 10 ) THEN                                  ELE16030
           IF(SE(ISE2+ISEM3A+3) .NE.SE(ISEM3A+3) ) NEWK=.TRUE.          ELE16040
        ENDIF                                                           ELE16050
C---- BENDING IN B END Z DIRECTION                                      ELE16060
C     SET UP TEMTERARY - TOTAL LOADS,DISP'S                             ELE16070
        PL=FT(12)                                                       ELE16080
        D =RT(12)+R(12)                                                 ELE16090
        DL=RT(12)                                                       ELE16100
        IF( M2B .EQ. 9 ) THEN                                           ELE16110
           D=ROIE2B + P(4)/SE(ISEM2B+60)                                ELE16120
           DL=ROIE2B + PL/SE(ISEM2B+60)                                 ELE16130
        ELSE IF( M2B .EQ. 11) THEN                                      ELE16140
           D=ROIE2B + P(4)/SE(ISEM2B+30)                                ELE16150
           DL=ROIE2B + PL/SE(ISEM2B+30)                                 ELE16160
        ELSE IF( M2B .EQ.  6) THEN                                      ELE16170
           IF( SE(ISEM2B+47) .NE. 0 ) THEN                              ELE16180
              D=(ROIE2B/LENGTH) + P(4)/SE(ISEM2B+47)                    ELE16190
              DL=(ROIE2B/LENGTH) + PL/SE(ISEM2B+47)                     ELE16200
           ENDIF                                                        ELE16210
        ENDIF                                                           ELE16220
        CALL MATLIB                                                     ELE16230
     &(MOPT,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,4),EXCR(1,4), ELE16240
     &   P(4)  ,PL    ,D     ,DL    ,V     ,VL    ,LELEM ,LMAT  ,      ELE16250
     &   MATM2B,MATTYP,SE(ISE2+ISEM3B),IELNO,DAMAGE)                    ELE16260
        IF( MATTYP .EQ. 9 ) THEN                                        ELE16270
           IF(SE(ISE2+ISEM3B) .NE.SE(ISEM3B) ) NEWK=.TRUE.              ELE16280
           ROIE2B=D-P(4)/SE(ISEM3B+60)                                  ELE16290
        ELSE IF( MATTYP .EQ.11 ) THEN                                   ELE16300
           IF(SE(ISE2+ISEM2B) .NE.SE(ISEM3B) ) NEWK=.TRUE.              ELE16310
           ROIE2B=D-P(4)/SE(ISEM3B+30)                                  ELE16320
        ELSE IF( MATTYP .EQ. 6 ) THEN                                   ELE16330
           IF(SE(ISE2+ISEM2B) .NE.SE(ISEM2B) ) NEWK=.TRUE.              ELE16340
           IF(SE(ISEM2B+47) .NE. 0 ) THEN                               ELE16350
              ROIE2B=(D-P(4)/SE(ISEM3B+47))*LENGTH                      ELE16360
           ENDIF                                                        ELE16370
        ELSE IF( MATTYP .EQ. 10 ) THEN                                  ELE16380
           IF(SE(ISE2+ISEM2B+3) .NE.SE(ISEM2B+3) ) NEWK=.TRUE.          ELE16390
        ENDIF                                                           ELE16400
C---- TORSION IN A END X DIRECTION                                      ELE16410
C     SET UP TEMTERARY - TOTAL LOADS,DISP'S                             ELE16420
        PL=FT(10)                                                       ELE16430
        D =RT(10)+R(10)-RT(4)-R(4)                                      ELE16440
        DL=RT(10)-RT(4)                                                 ELE16450
        CALL MATLIB                                                     ELE16460
     &(MOPT,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,5),EXCR(1,5), ELE16470
     &   P(5)  ,PL    ,D     ,DL    ,V     ,VL    ,LELEM ,LMAT  ,      ELE16480
     &   MATM3A,MATTYP,SE(ISE2+ISEM3A),IELNO,DAMAGE)                    ELE16490
        IF(SE(ISE2+ISEM3A+3) .NE.SE(ISEM3A+3) ) NEWK=.TRUE.             ELE16500
C---- AXIAL IN A END X DIRECTION                                        ELE16510
C     SET UP TEMTERARY - TOTAL LOADS,DISP'S                             ELE16520
        PL=FT(1)                                                        ELE16530
        D =RT(7)+R(7)-RT(1)-R(1)                                        ELE16540
        DL=RT(7)-RT(1)                                                  ELE16550
        CALL MATLIB                                                     ELE16560
     &(MOPT,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,6),EXCR(1,6), ELE16570
     &   P(6)  ,PL    ,D     ,DL    ,V     ,VL    ,LELEM ,LMAT  ,      ELE16580
     &   MATFXA,MATTYP,SE(ISE2+ISEFXA),IELNO,DAMAGE)                    ELE16590
        IF( MFXA .EQ. 10 ) THEN                                         ELE16600
           IF( SE(ISE2+ISEFXA+3) .NE. SE(ISEFXA+3) ) NEWK=.TRUE.        ELE16610
        ELSE IF( MFXA .EQ. 13 ) THEN                                    ELE16620
           IF( SE(ISE2+ISEFXA) .NE. SE(ISEFXA) ) NEWK=.TRUE.            ELE16630
        ENDIF                                                           ELE16640
      ENDIF                                                             ELE16650
C                                                                       ELE16660
C                                                                       ELE16670
C------ CALCULATE THE UNBALACED MEMBER FORCES ON THE JOINT              ELE16680
C                                                                       ELE16690
        UF( 5)= -( FT( 5)+F( 5)-P(1) )                                  ELE16700
        UF(11)= -( FT(11)+F(11)-P(2) )                                  ELE16710
        UF( 6)= -( FT( 6)+F( 6)-P(3) )                                  ELE16720
        UF(12)= -( FT(12)+F(12)-P(4) )                                  ELE16730
        UF(10)= -( FT(10)+F(10)-P(5) )                                  ELE16740
        UF( 7)= -( FT( 7)+F( 7)-P(6) )                                  ELE16750
        UF( 9)=(UF( 5)+UF(11))/LENGTH                                   ELE16760
        UF( 3)= -UF( 9)                                                 ELE16770
        UF( 2)=(UF( 6)+UF(12))/LENGTH                                   ELE16780
        UF( 8)= -UF( 2)                                                 ELE16790
        UF( 1)= -UF( 7)                                                 ELE16800
        UF( 4)= -UF(10)                                                 ELE16810
C                                                                       ELE16820
        DO 440 I=1,IELDOF                                               ELE16830
           K=LM(I)                                                      ELE16840
           EFUB(I)=0                                                    ELE16850
           DO 460 J=1,12                                                ELE16860
  460         EFUB(I)=EFUB(I) + A(I,J)*UF(J)                            ELE16870
           FUB(K)=FUB(K)+EFUB(I)                                        ELE16880
  440   CONTINUE                                                        ELE16890
C                                                                       ELE16900
        IF (BTEST(IBUG,7)) THEN                                         ELE16910
          WRITE(6,*)' UNBALANCE FORCES FOR ELEMENT ',IELNO,':'          ELE16920
          WRITE(6,497) (I,P(I),I=1,6)                                   ELE16930
          WRITE(6,495) (I,F(I),FT(I),UF(I),I=1,12)                      ELE16940
          WRITE(6,496) (I,LM(I),EFUB(I),I=1,IELDOF)                     ELE16950
        ENDIF                                                           ELE16960
C                                                                       ELE16970
C------ ADJUST F FOR UNBALANCED LOAD P                                  ELE16980
      DO 445 I=1,12                                                     ELE16990
  445   F(I)=F(I)+UF(I)                                                 ELE17000
C                                                                       ELE17010
  491 FORMAT (//'     IE3D BEAM FORCES...',5X/7X,A/,                    ELE17020
     & ' ',' ELEMENT LOAD         NODE',7X,'AXIAL',11X,'FY',13X,'FZ',  ELE17030
     & 11X,'TORSION',10X,'MY',13X,'MZ   STBFAG')                        ELE17040
  492 FORMAT (/,I10,I5,2X,'FORCE',I6,1P,6G15.6,                         ELE17050
     &        /15X    ,2X,'FORCE',I6,   6G15.6,3X,I1)                  ELE17060
  493 FORMAT (/,I10,I5,2X,'DISPL',I6,1P,6G15.6,                         ELE17070
     &        /15X    ,2X,'DISPL',I6,   6G15.6)                        ELE17080
  494 FORMAT (7X,A)                                                     ELE17090
  495 FORMAT(' I:',I2,' F:',1P,G12.4,' FT:',G12.4,' UF:',G12.4)         ELE17100
  496 FORMAT(' I:',I2,' LM(I):',1P,G12.4,' EFUB(I):',G12.4)             ELE17110
  497 FORMAT(' I:',I2,' P(I):',1P,G12.4)                                ELE17120
      RETURN                                                            ELE17130
C------------------------- DETERMINE TOTAL FORCES ----------------------ELE17140
  500 CONTINUE                                                          ELE17150
      IF (PRINT.AND.FIRST) WRITE (6,491) STEPID                         ELE17160
      IF(ICOMN .NE. 1 .AND. ICOMN .NE. 2 ) THEN                         ELE17170
C                                                                       ELE17180
C------ TOTAL FORCES AND DISPLACEMENTS                                  ELE17190
      DO 510 I=1,12                                                     ELE17200
C         IF (BTEST(IBUG,7)) WRITE(6,520) I,R(I),RT(I),F(I),FT(I)       ELE17210
          FT(I)=FT(I)+F(I)                                              ELE17220
          RT(I)=RT(I)+R(I)                                              ELE17230
  510 CONTINUE                                                          ELE17240
C                                                                       ELE17250
      DO 512 I=1,12                                                     ELE17260
          IF ( ABS(FT(I)).GT.ABS(FMAX(I*12-12+I)) ) THEN                ELE17270
             DO 511 J=0,11                                              ELE17280
  511           FMAX(I*12-11+J)=FT(J+1)                                 ELE17290
          ENDIF                                                         ELE17300
  512 CONTINUE                                                          ELE17310
      ELSE IF(ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                     ELE17320
         DO 513 I=1,12                                                  ELE17330
  513      FT(I)=RSAF(I)                                                ELE17340
      ENDIF                                                             ELE17350
C                                                                       ELE17360
  520 FORMAT (' I:',I3,' R:',1P,G12.4,' RT:',G12.4,                     ELE17370
     &                ' F:',1P,G12.4,' FT:',G12.4)                     ELE17380
      LOAD=1                                                            ELE17390
C     IF (BTEST(IBUG,0).AND.PRINT)                                      ELE17400
      IF (        PRINT)                                                ELE17410
     &                  WRITE (6,493) IELNO,LOAD,NODEI,(RT(J),J=1,6),   ELE17420
     &                                         NODEJ,(RT(J),J=7,12)     ELE17430
      IF (        PRINT) WRITE (6,492) IELNO,LOAD,NODEI,(FT(J),J=1,6),  ELE17440
     &                                         NODEJ,(FT(J),J=7,12),    ELE17450
     &                                         STBFAG                   ELE17460
C                                                                       ELE17470
C     ----- TRANSFER TEMPERARY HYSTERESIS DATA TO PERMANENT STORAGE     ELE17480
      ISE2=ISE/2                                                        ELE17490
      IF( .NOT. ELSTIC) THEN                                            ELE17500
      DO 415 I=1,ISE2                                                   ELE17510
  415   SE(I)=SE(I+ISE2)                                                ELE17520
      ENDIF                                                             ELE17530
C                                                                       ELE17540
      RETURN                                                            ELE17550
C                                                                       ELE17560
C------------------------- DETERMINE FIXED END FORCES, AND ADD TO FORCE-ELE17570
  600 IF (MAXELD.LE.0) RETURN                                           ELE17580
C                                                                       ELE17590
C------- CALL BMLOA TO DETERMINE THE FIXED END FORCES                   ELE17600
      CALL BMLOA(IELNO,FEMFLG,LENGTH,REL,MAXELD,NELD,NLOAD,            ELE17610
     &     FEM,IELD,ELD)                                               ELE17620
      FEMFLG=FEMFLG                                                     ELE17630
      IF (.NOT. FEMFLG) RETURN                                          ELE17640
C                                                                       ELE17650
C------- ADD FIXED END FORCES TO LOAD MATRIX                            ELE17660
C     IF( MTA .EQ. 9 .OR. MTA .EQ. 11 ) THEN                            ELE17670
C        DO 743 LOAD=1,NLOAD                                            ELE17680
C           IF( FEM(5,LOAD) .NE. 0 .OR. FEM(11,LOAD) .NE. 0 ) THEN      ELE17690
C           WRITE(6,*)' *** INVALID FIXED END FORCES INPUT IN ',        ELE17700
C    &              'ELEMENT NO. ',IELNO,' ***'                         ELE17710
C           ENDIF                                                       ELE17720
C 743    CONTINUE                                                       ELE17730
C     ENDIF                                                             ELE17740
C                                                                       ELE17750
C     IF( MZA .EQ. 9 .OR. MZA .EQ. 11 ) THEN                            ELE17760
C        DO 744 LOAD=1,NLOAD                                            ELE17770
C           IF( FEM(6,LOAD) .NE. 0 .OR. FEM(12,LOAD) .NE. 0 ) THEN      ELE17780
C           WRITE(6,*)' *** INVALID FIXED END FORCES INPUT IN ',        ELE17790
C    &              'ELEMENT NO. ',IELNO,' ***'                         ELE17800
C           ENDIF                                                       ELE17810
C 744    CONTINUE                                                       ELE17820
C     ENDIF                                                             ELE17830
C                                                                       ELE17840
      DO 620 LOAD=1,NLOAD                                               ELE17850
        DO 610 I=1,3                                                    ELE17860
        GFEM(I  )=0.                                                    ELE17870
        GFEM(I+3)=0.                                                    ELE17880
        GFEM(I+6)=0.                                                    ELE17890
        GFEM(I+9)=0.                                                    ELE17900
        DO 610 J=1,3                                                    ELE17910
           D1=FEM(J  ,LOAD)                                             ELE17920
           D2=FEM(J+3,LOAD)                                             ELE17930
           D3=FEM(J+6,LOAD)                                             ELE17940
           D4=FEM(J+9,LOAD)                                             ELE17950
        GFEM(I  )=GFEM(I  )+CTS(I,J)*D1                                 ELE17960
        GFEM(I+3)=GFEM(I+3)+ BS(I,J)*D1 + CTS(I,J)*D2                   ELE17970
        GFEM(I+6)=GFEM(I+6)+CTE(I,J)*D3                                 ELE17980
  610   GFEM(I+9)=GFEM(I+9)+ BE(I,J)*D3 + CTE(I,J)*D4                   ELE17990
      IF (BTEST(IBUG,7)) THEN                                           ELE18000
        WRITE(6,*) 'IELNO',IELNO                                        ELE18010
        CALL WMATRX( FEM  ,1 ,12,'LOCAL  FIXED END FORCES')             ELE18020
        CALL WMATRX(GFEM  ,1 ,12,'GLOBAL FIXED END FORCES')             ELE18030
      ENDIF                                                             ELE18040
C                                                                       ELE18050
      DO 620 J=1,6                                                      ELE18060
        JI=IDOF(JOINTI,J)                                               ELE18070
        JJ=IDOF(JOINTJ,J)                                               ELE18080
        FORCE(JI,LOAD)=FORCE(JI,LOAD)+GFEM(J  )                         ELE18090
  620   FORCE(JJ,LOAD)=FORCE(JJ,LOAD)+GFEM(J+6)                         ELE18100
      IF (BTEST(IBUG,7) ) THEN                                          ELE18110
          CALL WMATRX(FORCE ,NDOF,NLOAD,'GLOBAL LOAD MATRIX')           ELE18120
      ENDIF                                                             ELE18130
C                                                                       ELE18140
      RETURN                                                            ELE18150
C                                                                       ELE18160
C---------------------- WRITE MEMBER FORCES TO OUTPUT FILE --------------ELE18170
  700 CONTINUE                                                          ELE18180
C                                                                       ELE18190
      WRITE (IUNIT,710) NODEI,(FT(J),J=1,6),NODEJ,(FT(J),J=7,12),STBFAG ELE18200
      WRITE (IUNIT,710) NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12)        ELE18210
C                                                                       ELE18220
C     WRITE (IUNIT,715) SE(ISEMTA),SE(ISEMTB),SE(ISEMTA+4),SE(ISEMTB+4),ELE18230
C    &                  SE(ISEMTA+64),SE(ISEMTA+65),                   ELE18240
C    &                  SE(ISEMTB+64),SE(ISEMTB+65)                    ELE18250
C    &                  SE(ISEM3A),SE(ISEM3B),SE(ISEM3A+4),SE(ISEM3B+4) ELE18260
C                                                                       ELE18270
  710 FORMAT (I6,1P,6G12.5/,                                            ELE18280
     &        I6,1P,6G12.5,I1)                                          ELE18290
C                                                                       ELE18300
C 715 FORMAT (6X,1P,4G12.5/,                                            ELE18310
C    &        6X,1P,4G12.5)                                             ELE18320
C                                                                       ELE18330
      RETURN                                                            ELE18340
C                                                                       ELE18350
C-------------------- READ AND PRINT MEMBER FORCES FROM OUTPUT FILE ----ELE18360
  800 CONTINUE                                                          ELE18370
      BACKSPACE(IUNIT)                                                  ELE18380
      READ (IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT                        ELE18390
      WRITE (6,805)                                                     ELE18400
      ISTEP=-IWRITE                                                     ELE18410
  810 READ (IUNIT,815,END=830)                                          ELE18420
     &   NODEI,(FT(J),J=1,6),NODEJ,(FT(J),J=7,12),STBFAG               ELE18430
      READ (IUNIT,815,END=830)                                          ELE18440
     &   NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12)                      ELE18450
  815 FORMAT (I6,6G12.5/,                                               ELE18460
     &        I6,6G12.5,I1)                                             ELE18470
C                                                                       ELE18480
C     READ (IUNIT,716,END=830) SE(ISEMTA+4),SE(ISEMTB+4),               ELE18490
C    &                         SE(ISEM3A+4),SE(ISEM3B+4)                ELE18500
C 716 FORMAT (54X,1P,2G12.5/,                                           ELE18510
C    &        54X,1P,2G12.5)                                            ELE18520
C                                                                       ELE18530
C                                                                       ELE18540
          ISTEP=ISTEP+IWRITE                                            ELE18550
          T=TO+DT*ISTEP                                                 ELE18560
CCC       MODU=MOD(ISTEP,INC)                                           ELE18570
          IF (MOD(ISTEP,INC).EQ.0) THEN                                 ELE18580
            WRITE (6,820) ISTEP,T,NODEI,(FT(J),J=1,6),                  ELE18590
     &                            NODEJ,(FT(J),J=7,12),STBFAG           ELE18600
            WRITE (6,920)           (RT(J),J=1,12)                      ELE18610
            WRITE (6,717) SE(ISEMTA+4),SE(ISEMTB+4),                    ELE18620
C    &              SE(ISEM3A+4),SE(ISEM3B+4)                          ELE18630
          ENDIF                                                         ELE18640
          GO TO 810                                                     ELE18650
C                                                                       ELE18660
  805 FORMAT (//'     IE3D BEAM FORCES...',/                            ELE18670
     &  ' ','   STEP    TIME          NODE',7X,'AXIAL',11X,'FY',13X,'FZ', ELE18680
     &  11X,'TORSION',10X,'MY',13X,'MZ   STBFAG')                       ELE18690
  820 FORMAT (/,I10,1P,G15.5,0P,I6,1P,6G15.6,                           ELE18700
     &        /25X,           I6,   6G15.6,3X,I1)                      ELE18710
```

```
  920 FORMAT ( 25X,'DISP  ',       1P,6G15.6,
     &         /31X,                 6G15.6)
C 717 FORMAT ( 25X,'RULE  ',       1P,6GX,2G15.6,
C    &         /31X,                 6GX,2G15.6)
C
      830 RETURN
C
C--------------------- DETERMINE THE LENGTH OF THE MATERIAL DATA
  900 CONTINUE
      ISE=0
C---- LOOP FOR EACH MATERIAL -------
      DO 910 I=1,6
        IF( I.EQ.1) THEN
           ISEMTA=ISE+1
        ELSE IF( I.EQ.2) THEN
           ISEMTB=ISE+1
        ELSE IF( I.EQ.3) THEN
           ISEMZA=ISE+1
        ELSE IF( I.EQ.4) THEN
           ISEMZB=ISE+1
        ELSE IF( I.EQ.5) THEN
           ISEMXA=ISE+1
        ELSE IF( I.EQ.6) THEN
           ISEFXA=ISE+1
        ENDIF
        MATI  = RINPUT( I)
        CALL MATLIB
     &  ( 0    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,
     &    FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &    MATI  ,MATTYP,SE,IELNO,DAMAGE)
  910   ISE=ISE+LELEM
C
C----- DOUBLE THE STORAGE, TO ALLOW FOR TEMPORARY VALUES
       ISE=ISE*2
C
      RETURN
C
C--------------------- DETERMINE THE TOTAL STRAIN ENERGY
 1000 CONTINUE
      EESE=0
      EPSE=0
C
C     /* ENERGY FORMULATION IS NOT AVAILABLE FOR ELE09 ELEMENT */
C     IF( ELSTIC) THEN
        DO 1010 I=1,12
 1010     EESE=EESE + (FT(I)+F(I))*(R(I)+RT(I))/2.
C     ENDIF
C
      ENGDAT(1) = EESE
      ENGDAT(2) = 0
C
      RETURN
C
C--------------------- DUCTILITIES(BASED ON DEFINITION ONE ONLY
 1100 CONTINUE
C     ... NOTE ELE09 WAS LAST CALLED TO CALC INCIRMENTAL DISPL,
C              THE ENERGIES ASSOCIATED WITH INCR DISPL ARE STORED IN
C              THE ADDRESSES: ISEMTA+ISE/2
C                             ISEMTB+ISE/2
C                             ISEMZA+ISE/2
C                             ISEMZB+ISE/2
C                             ISEMXA+ISE/2
C                             ISEFXA+ISE/2
C.......... BENDING ENERGY, DUCTILITY AND EXCURSION .....
      ISE2=ISE/2
C---- BENDING IN A END Y DIRECTION
      MOPT=4
C      WRITE(40,*)' * ---MEMBER ', IELNO,' ---------------- *'
       CALL MATLIB
     &(MOPT,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,1),EXCR(1,1),
     &   FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &   MATMTA,MATTYP,SE(ISE2+ISEMTA),IELNO,DAMAGE)
C---- BENDING IN B END Y DIRECTION
       CALL MATLIB
     &(MOPT,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,2),EXCR(1,2),
     &   FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &   MATMTB,MATTYP,SE(ISE2+ISEMTB),IELNO,DAMAGE)
C---- BENDING IN A END Z DIRECTION
       CALL MATLIB
     &(MOPT,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,3),EXCR(1,3),
     &   FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &   MATMZA,MATTYP,SE(ISE2+ISEMZA),IELNO,DAMAGE)
C---- BENDING IN B END Z DIRECTION
       CALL MATLIB
     &(MOPT,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,4),EXCR(1,4),
     &   FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &   MATMZB,MATTYP,SE(ISE2+ISEMZB),IELNO,DAMAGE)
C---- TORSION IN A END X DIRECTION
       CALL MATLIB
     &(MOPT,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,5),EXCR(1,5),
     &   FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &   MATMXA,MATTYP,SE(ISE2+ISEMXA),IELNO,DAMAGE)
C---- AXIAL IN A END X DIRECTION
       CALL MATLIB
     &(MOPT,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DUCT(1,6),EXCR(1,6),
     &   FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &   MATFXA,MATTYP,SE(ISE2+ISEFXA),IELNO,DAMAGE)
C
C       WRITE(40,*)' * ------------------------------------------ *'
        DO 1109 I=1,6
          IF( ABS(DUCT(1,I)) .LT. 1 ) DUCT(1,I)=0
 1109   CONTINUE
C
        IF( DUCFAG .EQ. 0 ) THEN
        IF(  ABS(DUCT(1,1)) .GE. 1 .OR. ABS(DUCT(1,2)) .GE. 1
     &  .OR. ABS(DUCT(1,3)) .GE. 1 .OR. ABS(DUCT(1,4)) .GE. 1
     &  .OR. ABS(DUCT(1,5)) .GE. 1 .OR. ABS(DUCT(1,6)) .GE. 1 ) THEN
             DUCFAG=ABS(DUCT(1,1))
             IF( DUCFAG .LT. ABS(DUCT(1,2)) ) DUCFAG=ABS(DUCT(1,2))
             IF( DUCFAG .LT. ABS(DUCT(1,3)) ) DUCFAG=ABS(DUCT(1,3))
             IF( DUCFAG .LT. ABS(DUCT(1,4)) ) DUCFAG=ABS(DUCT(1,4))
             IF( DUCFAG .LT. ABS(DUCT(1,5)) ) DUCFAG=ABS(DUCT(1,5))
             IF( DUCFAG .LT. ABS(DUCT(1,6)) ) DUCFAG=ABS(DUCT(1,6))
        ENDIF
        ENDIF
C
       IF (PRINT.AND.FIRST) WRITE (6,1191)
       IF (PRINT) WRITE (6,1192)IELNO,(DUCT(1,J),J=1,6)
 1192 FORMAT (6X,I5,1X,6F11.5)
 1191 FORMAT (//'    IE3DBEAM  DUCTILITY(BASED ON DEFINITION 1):'/6X,
     &  ' ELEM#',      '     |MTA|       |MTB|       |MZA|',
     &  '     |MZB|       |MX |       |FX |')
      RETURN
C
C
C--------------------- PRINT MAXIMUM ELEMENT LOADS
C
 1200 CONTINUE
      LOAD2=0
      IF (FIRST) WRITE(6,494)' COLUMN ( LOAD) REPRESENTS THE DOF WHICH'
     &          //' HAS MAXIMUM VALUE'
      IF (FIRST) WRITE (6,491) ' MAXIMUM VALUES FOR ALL STEPS '
     &          //' NOTE: MAXIMUM VALUES WITH THE OTHER DOFS FORCES'
     &          //' ARE PRINT OUT AT THE SAME TIME'
      DO 309 I=0,11
           LOAD2=I+1
           IF( FMAX(I*12+1+I) .EQ. 0. ) GO TO 309
           WRITE (6,492) IELNO,LOAD2,NODEI,(FMAX(I*12+J),J=1,6),
     &                               NODEJ,(FMAX(I*12+J),J=7,12),STBFAG
  309 CONTINUE
C
      RETURN
C
C
C--------------------- RESET ELEMENT FORCES
C
 1300 CONTINUE
C----- ZERO MATRICIES
        DO 1340 I=1,12
          FMAX(I*12-11)=0
          F (I)=0
          FT(I)=0
          R (I)=0
 1340     RT(I)=0
        DO 1310 I=1,ISE
 1310     SE(I)=0
C
      WRITE (6,1330) IELNO
 1330 FORMAT (' IE3D BEAM..... ELEMENT #',I6,
     &  ' INTERNAL FORCES AND HISTERESIS MODELS ARE RESET TO ZERO')
        FALSE=.FALSE.
        LSTTYP=0
        CALL MATLIB
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,
     &    FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &    MATMTA,MATTYP,SE(ISEMTA),IELNO,DAMAGE)
        CALL MATLIB
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,
     &    FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &    MATMTB ,MATTYP,SE(ISEMTB),IELNO,DAMAGE)
        CALL MATLIB
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,
     &    FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &    MATMZA ,MATTYP,SE(ISEMZA),IELNO,DAMAGE)
        CALL MATLIB
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,
     &    FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &    MATMZB ,MATTYP,SE(ISEMZB),IELNO,DAMAGE)
        CALL MATLIB
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,
     &    FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &    MATMXA ,MATTYP,SE(ISEMXA),IELNO,DAMAGE)
        CALL MATLIB
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,
     &    FT    ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &    MATFXA ,MATTYP,SE(ISEFXA),IELNO,DAMAGE)
C
       RETURN
C
C--------------------- DAMAGE INDEX
C
 1400 DAMAGE = 0.
C
      RETURN
C
      END
C-----------------------------------------------------------------------
C          DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE
C          END DEBUG
C-----------------------------------------------------------------------
      SUBROUTINE ELE12
     &  (IOPT      ,IUNIT     ,INC,NEWK    ,FIRST     ,PRINT     ,
     &   NDOF      ,NLOAD     ,MAXNOD      ,NNODE     ,NCOS      ,
     &   ID,ELSTIC ,IDOF      ,COORD       ,COSINE    ,JTCOS     ,
     &   CONST     ,DISPL     ,FORCE       ,IREL      ,BUG       ,
     &   IBUG      ,ACCEL     ,KGDATA      ,GRAV      ,PGEOM     ,
     &   MAXELD    ,NELD      ,IELD        ,ELD       ,NAME      ,
     &   STEPID    ,IELNO     ,RINPUT      ,FUB       ,          ,
     &   EESE      ,EPSE,DAMAGE,DUCFAG     ,IELTYP    ,IELDOF    ,
     &   R         ,RT        ,F           ,FT        ,          ,
     &   LM        ,CTS       ,CTE         ,BS        ,BE        ,
     &   NODEI     ,NODEJ     ,JOINTI      ,JOINTJ    ,          ,
     &   REL       ,LENGTH    ,            ,          ,          ,
     &   ISE       ,ENGDAT    ,FMAX        ,S         ,A         ,
     &   MAT       ,SE        ,STIFF       )
C
      IMPLICIT REAL(A-H,O-Z)
C
      COMMON /RSA/ICOMN,RSAF(12)
      REAL ICOMN
C
      LOGICAL ERR,FIRST,PRINT,BUG,FALSE,BTEST
      LOGICAL NEWK,ELSTIC
      CHARACTER*80 NAME
      CHARACTER*(*) STEPID
      CHARACTER*6  IRELC
      DIMENSION KGDATA(3)
      DIMENSION  IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS)
      DIMENSION CONST(MAXNOD,6),    ID(MAXNOD)   ,JTCOS(MAXNOD)
      DIMENSION ASAT(12,12),SAT(12,12),STIFF(78)
C     DIMENSION A(12,12),  S(12,12),LM(12),LMT(12),LMKG(12),FMAX(12)
      DIMENSION A(12,12),  S(12,12),LM(12),LMT(12),LMKG(12),FMAX(144)
      DIMENSION CTS( 3, 3),CTE( 3, 3),COSLOC(3,3),VX(3),VY(3),VZ(3)
      DIMENSION BS( 3, 3),BE( 3, 3),SE(ISE),ENGDAT(2)
      DIMENSION R(12),DISPL(NDOF,NLOAD),F(12),RT(12),FT(12),FUB(NDOF)
      DIMENSION FORCE(NDOF,NLOAD),IELD(5,MAXELD),ELD(12,MAXELD)
      DIMENSION ASA(12,12),RINPUT(100)
      DIMENSION DUCT(3),EXCR(6)
      DIMENSION V(12),VL(12)
      DIMENSION P(12),PL(12),D(12),DL(12)
      REAL LENGTH
      INTEGER REL,RELT
      DIMENSION MD(13)
      DATA MD/1,2,4,7,11,16,22,29,37,46,56,67,79/
C
C-----------------------------------------------------------------------
C VARIABLES:
C-----------------------------------------------------------------------
C---------- GLOBAL VARIABLES
C IOPT    = 1, INITIALIZE BEAM COLUMN ELEMENT
C         = 2, CALCULATE GEOMETRIC STIFFNESS
C         = 3, FORM STIFFNESS
C         = 4, CALCULATE INCREMENTAL FORCES
C         = 5, CALCULATE TOTAL FORCES AND ENERGIES
C IUNIT   = OUTPUT FILE UNIT # FOR PRINTING OUTPUT
C ERR     = ERR OR FLAG, IF AN ERR OR HAS OCCURED, ERR=.TRUE.
C FIRST   = FLAG, FIRST=.TRUE., PRINT HEADERS
C PRINT   = FLAG, PRINT=.TRUE., PRINT DATA
C NDOF    = # OF GLOBAL DOF
C NLOAD   = # OF LOAD COMBINATIONS
C MAXNOD  = # ROW DIMENSION OF NODE ARRAY
C NNODE   = # OF NODES
C ID      = ARRAY OF EXTERNAL NODE NUMBERS
C IDOF    = ARRAY OF DEGREES OF FREEDOM
C COORD   = ARRAY OF COORDINATES
C COSINE  = ARRAY OF DIRECTION COSINES OF NODES
C CONST   = ARRAY OF CONSTRAINT TRANSFORMATIONS
C DISPL   = GLOBAL DISPLACEMENT MATRIX
C IZMAT   = LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR
C NAME    = ELEMENT NAME
C IELNO   = ELEMENT NUMBER
C RINPUT  = INPUT DATA
C STIFF   = OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)
C LM      = OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX
C---------- ELEMENT VARIABLES
C H,Q,AI,AJ,C,BI,BJ,S22,S26,S212,S33,S35,S311,D
C         = INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX
C PKG     = INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX...
C R       = INCRIMENTAL DISPLACEMENTS
C RT      = TOTAL DISPLACEMENTS
C F       = INCRIMENTAL FORCES
C FT      = TOTAL FORCES
C LM      = LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX
C CTS     = LOCAL TO GLOBAL ROTATION MATRIX START
```

```
C CTE     = LOCAL TO GLOBAL ROTATION MATRIX END                                 ELE00890
C BS      = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX STARTELE00900
C BE      = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END  ELE00910
C STIFF   = ELEMENT STIFFNESS                                           ELE00920
C IELTYP  = ELEMENT TYPE = 12                                           ELE00930
C NODEI   = EXTERNAL JOINT NUMBER, END I                                ELE00940
C NODEJ   = EXTERNAL JOINT NUMBER, END J                                ELE00950
C JOINTI  = INTERNAL JOINT NUMBER, END I                                ELE00960
C JOINTJ  = INTERNAL JOINT NUMBER, END J                                ELE00970
C---------- TEMPROARY VARIABLES                                         ELE00980
C VX      = VECTOR DEFINING THE ELEMENT X AXIS                          ELE00990
C VY      = VECTOR DEFINING THE ELEMENT Y AXIS                          ELE01000
C COSLOC  = LOCAL COSINE MATRIX                                         ELE01010
C A       = GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                 ELE01020
C S       = ELEMENT STIFFNESS AT LOCAL COORD. AT ENDS OF FLEXABLE PART  ELE01030
C SAT     = PRODUCT OF S*A'                                             ELE01040
C ASAT    = PRODUCT OF A*SAT                                            ELE01050
C-------------------------------------------------------------------------ELE01060
C                                                                       ELE01070
C------- CHOOSE OPTION                                                  ELE01080
      IF (IOPT.LT.1 .OR.IOPT.GT.14)                                     ELE01090
     & WRITE (6,*) 'INVALID OPTION IN ELE-09, IOPT=',IOPT               ELE01100
C                                                                       ELE01110
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,        ELE01120
     &                                          1300,1400),IOPT         ELE01130
C                                                                       ELE01140
  100 CONTINUE                                                          ELE01150
C                                                                       ELE01160
C----- ZERO STRAIN ENERGY                                              ELE01170
      ENGDAT(1) = 0                                                     ELE01180
      ENGDAT(2) = 0                                                     ELE01190
C                                                                       ELE01200
      IF ( BTEST(IBUG,7) ) THEN                                         ELE01210
         WRITE (6,*) NAME                                               ELE01220
         CALL WMATRX(RINPUT,8, 1,'RINPUT')                              ELE01230
      ENDIF                                                             ELE01240
C                                                                       ELE01250
C----- INTERPERTATE INPUT DATA                                         ELE01260
      IELTYP= 12                                                        ELE01270
      MAT     = RINPUT( 1)                                              ELE01280
      NODEI   = RINPUT( 2)                                              ELE01290
      NODEJ   = RINPUT( 3)                                              ELE01300
      VY(1)   = RINPUT( 4)                                              ELE01310
      VY(2)   = RINPUT( 5)                                              ELE01320
      VY(3)   = RINPUT( 6)                                              ELE01330
      XS      = RINPUT( 7)                                              ELE01340
      XE      = RINPUT( 8)                                              ELE01350
C                                                                       ELE01360
C                                                                       ELE01370
C----- GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S            ELE01380
      JOINTI=IQUICK(NODEI,ID,NNODE)                                     ELE01390
      JOINTJ=IQUICK(NODEJ,ID,NNODE)                                     ELE01400
C     WRITE (6,*) 'ID:',ID                                              ELE01410
C                                                                       ELE01420
C----- PRINT DATA FOR PLOTTING                                         ELE01430
      IF (BTEST(IBUG,2)) THEN                                           ELE01440
       WRITE(7,10) NODEJ, NODEI,                                        ELE01450
     &              (COORD(JOINTJ,I),I=1,3),(COORD(JOINTI,J),J=1,3)     ELE01460
   10  FORMAT (I5,1X,I5,1P,6G12.4)                                      ELE01470
      ENDIF                                                             ELE01480
C                                                                       ELE01490
C-------------------------------------------------------------------    ELE01500
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                            ELE01510
C-------------------------------------------------------------------    ELE01520
C----- DEFINE VECTOR X, LENGTH                                         ELE01530
      VX(1)=COORD(JOINTJ,1)-COORD(JOINTI,1)                             ELE01540
      VX(2)=COORD(JOINTJ,2)-COORD(JOINTI,2)                             ELE01550
      VX(3)=COORD(JOINTJ,3)-COORD(JOINTI,3)                             ELE01560
      LENGTH=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)-XS-XE                     ELE01570
C                                                                       ELE01580
      SE(79)=LENGTH                                                     ELE01590
C                                                                       ELE01600
      IF (LENGTH.LE.0) THEN                                             ELE01610
         WRITE(6,101) IELNO,JOINTI,JOINTJ,LENGTH                        ELE01620
  101    FORMAT(1X,20('*'),'ER','ROR - LENGTH IS LE 0',/                ELE01630
     &    5X,'IELNO=',I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,              ELE01640
     &    5X,'LENGTH',1P,G15.6/                                         ELE01650
     &    5X,'REVISE INPUT.............................')              ELE01660
         LENGTH=1.0                                                     ELE01670
      ENDIF                                                             ELE01680
      IF ( BTEST(IBUG,7) ) THEN                                         ELE01690
         WRITE (6,*) 'JOINTS',JOINTI,JOINTJ                             ELE01700
         WRITE (6,*) 'VX',VX                                            ELE01710
         WRITE (6,*) 'VY',VY                                            ELE01720
         WRITE (6,*) 'LENGTH  ',LENGTH                                  ELE01730
      ENDIF                                                             ELE01740
C----- GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES CT AND B            ELE01750
      ICS=JTCOS(JOINTI)                                                 ELE01760
      ICE=JTCOS(JOINTJ)                                                 ELE01770
C                                                                       ELE01780
      CALL ROTXYZ(VX,VY,COSINE(1,1,ICS),CTS, XS ,0.,0.,BS,              ELE01790
     &     CONST(JOINTI,1),MAXNOD)                                      ELE01800
      CALL ROTXYZ(VX,VY,COSINE(1,1,ICE),CTE,-XE ,0.,0.,BE,              ELE01810
     &     CONST(JOINTJ,1),MAXNOD)                                      ELE01820
      IF ( BTEST(IBUG,7) ) THEN                                         ELE01830
         CALL WMATRX( CTS  , 3, 3,'CTS   ')                             ELE01840
         CALL WMATRX( CTE  , 3, 3,'CTE   ')                             ELE01850
      ENDIF                                                             ELE01860
C                                                                       ELE01870
C----- GET GLOBAL CONSTRAINT MATRIX BS, BE                              ELE01880
C     BS(1,1)=CONST(JOINTI,1)*CTS(2,1)+CONST(JOINTI,2)*CTS(3,1)+BS(1,1) ELE01890
C     BS(1,2)=CONST(JOINTI,1)*CTS(2,2)+CONST(JOINTI,2)*CTS(3,2)+BS(1,2) ELE01900
C     BS(1,3)=CONST(JOINTI,1)*CTS(2,3)+CONST(JOINTI,2)*CTS(3,3)+BS(1,3) ELE01910
C     BS(2,1)=CONST(JOINTI,3)*CTS(1,1)+CONST(JOINTI,4)*CTS(3,1)+BS(2,1) ELE01920
C     BS(2,2)=CONST(JOINTI,3)*CTS(1,2)+CONST(JOINTI,4)*CTS(3,2)+BS(2,2) ELE01930
C     BS(2,3)=CONST(JOINTI,3)*CTS(1,3)+CONST(JOINTI,4)*CTS(3,3)+BS(2,3) ELE01940
C     BS(3,1)=CONST(JOINTI,5)*CTS(1,1)+CONST(JOINTI,6)*CTS(2,1)+BS(3,1) ELE01950
C     BS(3,2)=CONST(JOINTI,5)*CTS(1,2)+CONST(JOINTI,6)*CTS(2,2)+BS(3,2) ELE01960
C     BS(3,3)=CONST(JOINTI,5)*CTS(1,3)+CONST(JOINTI,6)*CTS(2,3)+BS(3,3) ELE01970
C                                                                       ELE01980
C     BE(1,1)=CONST(JOINTJ,1)*CTE(2,1)+CONST(JOINTJ,2)*CTE(3,1)+BE(1,1) ELE01990
C     BE(1,2)=CONST(JOINTJ,1)*CTE(2,2)+CONST(JOINTJ,2)*CTE(3,2)+BE(1,2) ELE02000
C     BE(1,3)=CONST(JOINTJ,1)*CTE(2,3)+CONST(JOINTJ,2)*CTE(3,3)+BE(1,3) ELE02010
C     BE(2,1)=CONST(JOINTJ,3)*CTE(1,1)+CONST(JOINTJ,4)*CTE(3,1)+BE(2,1) ELE02020
C     BE(2,2)=CONST(JOINTJ,3)*CTE(1,2)+CONST(JOINTJ,4)*CTE(3,2)+BE(2,2) ELE02030
C     BE(2,3)=CONST(JOINTJ,3)*CTE(1,3)+CONST(JOINTJ,4)*CTE(3,3)+BE(2,3) ELE02040
C     BE(3,1)=CONST(JOINTJ,5)*CTE(1,1)+CONST(JOINTJ,6)*CTE(2,1)+BE(3,1) ELE02050
C     BE(3,2)=CONST(JOINTJ,5)*CTE(1,2)+CONST(JOINTJ,6)*CTE(2,2)+BE(3,2) ELE02060
C     BE(3,3)=CONST(JOINTJ,5)*CTE(1,3)+CONST(JOINTJ,6)*CTE(2,3)+BE(3,3) ELE02070
C                                                                       ELE02080
      IF ( BTEST(IBUG,7) ) THEN                                         ELE02090
         CALL WMATRX(BS, 3, 3,'BS    ')                                 ELE02100
         CALL WMATRX(BE, 3, 3,'BE    ')                                 ELE02110
      ENDIF                                                             ELE02120
C                                                                       ELE02130
C----- ZERO MATRICIES                                                  ELE02140
       DO 40 I=1,12                                                     ELE02150
         FMAX(I)=0                                                      ELE02160
         F (I)=0                                                        ELE02170
         FT(I)=0                                                        ELE02180
         R (I)=0                                                        ELE02190
         RT(I)=0                                                        ELE02200
         DO 40 J=1,12                                                   ELE02210
           S(I,J)=0.                                                    ELE02220
   40      A(I,J)=0.                                                    ELE02230
C                                                                       ELE02240
C----- ASSEMBLE TRANSFORMATION MATRIX A                                 ELE02250
       DO 50 I=1,3                                                      ELE02260
       DO 50 J=1,3                                                      ELE02270
         A(I  ,J  )=CTS(I,J)                                            ELE02280
         A(I+3,J+3)=CTS(I,J)                                            ELE02290
         A(I+6,J+6)=CTE(I,J)                                            ELE02300
         A(I+9,J+9)=CTE(I,J)                                            ELE02310
         A(I+3,J  )=BS(I,J)                                             ELE02320
   50    A(I+9,J+6)=BE(I,J)                                             ELE02330
      IF ( BTEST(IBUG,7) ) CALL WMATRX( A    ,12,12,' A    ')           ELE02340
C                                                                       ELE02350
C----- GET MATERIAL PROPERTIES                                         ELE02360
        FALSE=.FALSE.                                                   ELE02370
        LSTTYP=0                                                        ELE02380
        CALL MATLIB                                                     ELE02390
     &  ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,     ELE02400
     &    FT     ,F     ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,     ELE02410
     &    MAT,MATTYP,SE,IELNO,DAMAGE)                                    ELE02420
        IF( MATTYP .NE. 12) THEN                                        ELE02430
          WRITE(6,190)IELNO,MATTYP,NODEI,NODEJ                          ELE02440
  190     FORMAT (' ELEMENT: ',I6,' INCOMPATIBLE MATERIAL PROP.#',I5,   ELE02450
     &    '  START JOINT:',I5,'  END JOINT:',I5,/                       ELE02460
     &    ' THE ELEMENT IS REJECTED AND ER','ROR CHECKING CONTINUES')    ELE02470
          ERR=.TRUE.                                                    ELE02480
          GO TO 191                                                     ELE02490
        ENDIF                                                           ELE02500
C                                                                       ELE02510
      EESE=0                                                            ELE02520
      EPSE=0                                                            ELE02530
C                                                                       ELE02540
C---------- NO END RELEASE FOR THIS ELEMENT                             ELE02550
      REL=0                                                             ELE02560
C                                                                       ELE02570
C---------- SET STIFFNESS COIEFFICIENTS...                              ELE02580
C                                                                       ELE02590
C-------------------------------------------------------------------    ELE02600
C  ASSEMBLE LOCAL STIFFNESS MATRIX                                      ELE02610
C-------------------------------------------------------------------    ELE02620
      K=0                                                               ELE02630
      DO 77 I=1,12                                                      ELE02640
      DO 77 J=I,12                                                      ELE02650
         K=K+1                                                          ELE02660
         IF( ABS(SE(K)) .LT. 1.E-6) SE(K)=0                             ELE02670
         S(I,J)=SE(K)                                                   ELE02680
         S(J,I)=S(I,J)                                                  ELE02690
   77 CONTINUE                                                          ELE02700
C                                                                       ELE02710
C     IF (FIRST) CALL WMATRX( S    ,12,12,' S    ')                   ELE02720
C                                                                       ELE02730
C----- CALCULATE SAT                                                   ELE02740
      DO 60 I=1,12                                                      ELE02750
        DO 60 J=1,12                                                    ELE02760
        SAT(I,J)=0                                                      ELE02770
          DO 60 K=1,12                                                  ELE02780
   60       SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)                            ELE02790
C                                                                       ELE02800
C----- CALCULATE ASAT                                                  ELE02810
      DO 70 I=1,12                                                      ELE02820
        DO 70 J=1,12                                                    ELE02830
        ASAT(I,J)=0                                                     ELE02840
          DO 70 K=1,12                                                  ELE02850
   70     ASAT(I,J)=ASAT(I,J)+A(I,K)*SAT(K,J)                          ELE02860
      IF ( BTEST(IBUG,8) ) CALL WMATRX(ASAT  ,12,12,' ASAT ')           ELE02870
C                                                                       ELE02880
C----- DEGREES OF FREEDOM                                              ELE02890
C      REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF...             ELE02900
      DO 75 I=1,6                                                       ELE02910
        IF ( IDOF(JOINTI,I).NE.IDOF(JOINTJ,I) ) THEN                    ELE02920
           LMT(I  )=IDOF(JOINTI,I)                                      ELE02930
           LMT(I+6)=IDOF(JOINTJ,I)                                      ELE02940
        ELSE                                                            ELE02950
           LMT(I  )= 0                                                  ELE02960
           LMT(I+6)= 0                                                  ELE02970
        ENDIF                                                           ELE02980
   75   CONTINUE                                                        ELE02990
C                                                                       ELE03000
C----- CONVERT TO HALF STORAGE MODE  BY COLUMNS                        ELE03010
C                                                                       ELE03020
C          1  3  6 10 15 21 28 36 45 55 66 78  NUMBER INDICATES        ELE03030
C             2  5  9 14 20 27 35 44 54 65 77  LOCATION OF DATA        ELE03040
C                4  8 13 19 26 34 43 53 64 76  IN ARRAY STIFF          ELE03050
C                   7 12 18 25 33 42 52 63 75                          ELE03060
C                  11 17 24 32 41 51 62 74                             ELE03070
C                     16 23 31 40 50 61 73                             ELE03080
C                        22 30 39 49 60 72                             ELE03090
C                           29 38 48 59 71                             ELE03100
C                              37 47 58 70                             ELE03110
C                                 46 57 69                             ELE03120
C                                    56 68                             ELE03130
C                                       67                             ELE03140
C                                                                       ELE03150
      L=0                                                               ELE03160
      IELDOF=0                                                          ELE03170
      DO 90 I=1,12                                                      ELE03180
C       IF (LMT(I).EQ.0 .OR. ASAT(I,I).LE.0) GO TO 90                   ELE03190
        IF (LMT(I).EQ.0                     ) GO TO 90                  ELE03200
        IELDOF=IELDOF+1                                                 ELE03210
        LM(IELDOF)=LMT(I)                                               ELE03220
        DO 80 J=I,1,-1                                                  ELE03230
C         IF (LMT(J).EQ.0 .OR. ASAT(J,J).LE.0 ) GO TO 80                ELE03240
          IF (LMT(J).EQ.0                      ) GO TO 80               ELE03250
          L=L+1                                                         ELE03260
          STIFF(L)=ASAT(I,J)                                            ELE03270
   80     CONTINUE                                                      ELE03280
   90   CONTINUE                                                        ELE03290
      IF ( BTEST(IBUG,7) ) THEN                                         ELE03300
         WRITE (6,*) 'LMT=',LMT                                         ELE03310
         WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)          ELE03320
         CALL LMATRX(STIFF,MD,IELDOF,(L+1),'ELEMENT STIFFNESS')         ELE03330
      ENDIF                                                             ELE03340
C                                                                       ELE03350
C                                                                       ELE03360
C----- RELEASE UNUSED STORAGE                                          ELE03370
  180 IREL= 78-L                                                        ELE03380
C                                                                       ELE03390
C     PRINT COLUMN DATA                                                 ELE03400
      IF (FIRST .OR. BTEST(IBUG,7) .OR. BTEST(IBUG,8) ) WRITE (6,192)   ELE03410
  191 WRITE(6,193) NAME,IELNO,MAT,NODEI,NODEJ,LENGTH,                   ELE03420
     &    VY(1),VY(2),VY(3),XS,-XE                                      ELE03430
C                                                                       ELE03440
  192 FORMAT(/' ELEMENT 12,  STABILITY ELEMENT'//                       ELE03450
     &  T22,'#   MATL   START   END',3X,'LENGTH',3X,                   ELE03460
     &  11('-'),'  Y-AXIS  ',                                           ELE03470
     &  12('-'),'  START DIST    END DIST     ')                       ELE03480
  193 FORMAT(1X,A15,3I6,1X,I6,2X,1P,G10.4,0P,                           ELE03490
     &        F9.5,' I',SP,F9.5,' J',F9.5,' K',1P,SS,2G12.4)          ELE03500
      RETURN                                                            ELE03510
C------------------------- DETERMINE GEOMETRIC STIFF ------------------ELE03520
  200 CONTINUE                                                          ELE03530
C----GEOMETRIC STIFFNESS IS INCLUDED IN THE ELEMENT STIFFNESS.          ELE03540
C    NO NEED TO CALCULATED                                            ELE03550
C                                                                       ELE03560
      RETURN                                                            ELE03570
C------------------------- DETERMINE STIFFNESS ------------------------ELE03580
  300 CONTINUE                                                          ELE03590
C                                                                       ELE03600
C---- STIFFNESS CHANGES AT EACH STEP FOR STABILITY ELEMENT              ELE03610
C                                                                       ELE03620
C-------------------------------------------------------------------    ELE03630
C  ASSEMBLE LOCAL STIFFNESS MATRIX                                      ELE03640
C-------------------------------------------------------------------    ELE03650
      K=0                                                               ELE03660
      DO 87 I=1,12                                                      ELE03670
      DO 87 J=I,12                                                      ELE03680
         K=K+1                                                          ELE03690
         IF( ABS(SE(K)) .LT. 1.E-6) SE(K)=0                             ELE03700
         S(I,J)=SE(K)                                                   ELE03710
         S(J,I)=S(I,J)                                                  ELE03720
```

```
   87 CONTINUE                                                          ELE03730
C     IF (FIRST) CALL WMATRX( S    ,12,12,' S    ')                    ELE03740
C                                                                       ELE03750
C----- CALCULATE SAT                                                   ELE03760
      DO 62 I=1,12                                                      ELE03770
        DO 62 J=1,12                                                    ELE03780
        SAT(I,J)=0                                                      ELE03790
          DO 62 K=1,12                                                  ELE03800
   62       SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)                            ELE03810
C                                                                       ELE03820
C----- CALCULATE ASAT                                                  ELE03830
      DO 72 I=1,12                                                      ELE03840
        DO 72 J=1,12                                                    ELE03850
        ASAT(I,J)=0                                                     ELE03860
          DO 72 K=1,12                                                  ELE03870
   72       ASAT(I,J)=ASAT(I,J)+A(I,K)*SAT(K,J)                        ELE03880
      IF ( BTEST(IBUG,8) ) CALL WMATRX(ASAT  ,12,12,' ASAT ')           ELE03890
C                                                                       ELE03900
C----- DEGREES OF FREEDOM                                              ELE03910
C      REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF...             ELE03920
      DO 12 I=1,6                                                       ELE03930
        IF ( IDOF(JOINTI,I).NE.IDOF(JOINTJ,I) ) THEN                    ELE03940
          LMT(I  )=IDOF(JOINTI,I)                                       ELE03950
          LMT(I+6)=IDOF(JOINTJ,I)                                       ELE03960
        ELSE                                                            ELE03970
          LMT(I  )= 0                                                   ELE03980
          LMT(I+6)= 0                                                   ELE03990
        ENDIF                                                           ELE04000
   12   CONTINUE                                                        ELE04010
C                                                                       ELE04020
C----- CONVERT TO HALF STORAGE MODE  BY COLUMNS                        ELE04030
C                                                                       ELE04040
      L=0                                                               ELE04050
      IELDOF=0                                                          ELE04060
      DO 92 I=1,12                                                      ELE04070
C       IF (LMT(I).EQ.0 .OR. ASAT(I,I).LE.0) GO TO 92                   ELE04080
        IF (LMT(I).EQ.0                      ) GO TO 92                 ELE04090
        IELDOF=IELDOF+1                                                 ELE04100
        LM(IELDOF)=LMT(I)                                               ELE04110
        DO 82 J=1,I-1                                                   ELE04120
C         IF (LMT(J).EQ.0 .OR. ASAT(J,J).LE.0 ) GO TO 82                ELE04130
          IF (LMT(J).EQ.0                    ) GO TO 82                 ELE04140
          L=L+1                                                         ELE04150
          STIFF(L)=ASAT(I,J)                                            ELE04160
   82     CONTINUE                                                      ELE04170
   92     CONTINUE                                                      ELE04180
      IF ( BTEST(IBUG,7) ) THEN                                         ELE04190
        WRITE (6,*) 'LMT=',LMT                                          ELE04200
        WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)           ELE04210
        CALL LMATRX(STIFF,MD,IELDOF,(L+1),'ELEMENT STIFFNESS')          ELE04220
        WRITE (6,*) 'KGDATA:',KGDATA,' PKG:',PKG                        ELE04230
      ENDIF                                                             ELE04240
C                                                                       ELE04250
      RETURN                                                            ELE04260
C----------------------------- DETERMINE INCRIMENTAL FORCES ------------ELE04270
  400 CONTINUE                                                          ELE04280
      IF (PRINT.AND.FIRST) WRITE (6,491) STEPID                         ELE04290
C                                                                       ELE04300
C                                                                       ELE04310
C------ LOOP FOR EACH LOAD                                             ELE04320
      DO 430 LOAD=1,NLOAD                                               ELE04330
C                                                                       ELE04340
C------ TRANSFORM FRAME DISPL INTO LOCAL DISPL                          ELE04350
        DO 410 I=1,3                                                    ELE04360
          R(I  )=0.                                                     ELE04370
          R(I+3)=0.                                                     ELE04380
          R(I+6)=0.                                                     ELE04390
          R(I+9)=0.                                                     ELE04400
          DO 410 J=1,3                                                  ELE04410
            D1=DISPL(IDOF(JOINTI,J  ),LOAD)                             ELE04420
            D2=DISPL(IDOF(JOINTI,J+3),LOAD)                             ELE04430
            D3=DISPL(IDOF(JOINTJ,J  ),LOAD)                             ELE04440
            D4=DISPL(IDOF(JOINTJ,J+3),LOAD)                             ELE04450
          R(I  )=R(I  )+CTS(J,I)*D1 + BS(J,I)*D2                        ELE04460
          R(I+3)=R(I+3)+CTS(J,I)*D2                                     ELE04470
          R(I+6)=R(I+6)+CTE(J,I)*D3 + BE(J,I)*D4                        ELE04480
  410     R(I+9)=R(I+9)+CTE(J,I)*D4                                     ELE04490
C                                                                       ELE04500
C------ INCRIMENTAL FORCES                                             ELE04510
      DO 435 I=1,12                                                     ELE04520
C     DO 435 J=1,12                                                     ELE04530
        F(I)=0                                                          ELE04540
        DO 435 K=1,12                                                   ELE04550
  435     F(I)=F(I)+S(I,K)*R(K)                                         ELE04560
C                                                                       ELE04570
C                                                                       ELE04580
C------ DETERMINE MAXIMUN VALUES FOR MULTIPLE LOAD CASES               ELE04590
      IF (NLOAD.GT.1) THEN                                              ELE04600
        DO 425 I=1,12                                                   ELE04610
          IF ( ABS(F(I)).GT.ABS(FMAX(I*12-12+I)) ) THEN                 ELE04620
            DO 426 J=0,11                                               ELE04630
  426         FMAX(I*12-11+J)=F(J+1)                                    ELE04640
          ENDIF                                                         ELE04650
  425   CONTINUE                                                        ELE04660
      ENDIF                                                             ELE04670
C                                                                       ELE04680
C------ PRINT FORCES AND DEFORMATIONS                                  ELE04690
C     IF (BTEST(IBUG,0).AND.PRINT)                                      ELE04700
      SP=SE(84)                                                         ELE04710
      FLP=SE(98)                                                        ELE04720
      IF (         PRINT)                                               ELE04730
     &                   WRITE (6,493) IELNO,LOAD,NODEI,(R(J),J=1,6),  ELE04740
     &                                            NODEJ,(R(J),J=7,12),ELE04750
     &                                            FLP                 ELE04760
      IF (         PRINT) WRITE (6,492) IELNO,LOAD,NODEI,(F(J),J=1,6),ELE04770
     &                                            NODEJ,(F(J),J=7,12),ELE04780
     &                                            SP                  ELE04790
C                                                                       ELE04800
  430 CONTINUE                                                          ELE04810
C                                                                       ELE04820
C ------THIS IS FOR SOL05A USE ONLY                                    ELE04830
      IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                         ELE04840
        DO 431 I=1,12                                                   ELE04850
  431   ASAF(I)=F(I)                                                    ELE04860
      ENDIF                                                             ELE04870
C                                                                       ELE04880
C                                                                       ELE04890
C ------CALLL MATLIB TO GET STIFFNESS                                  ELE04900
C       THIS IS PERFORMED ONLY FOR INELASTIC ANALYSIS                   ELE04910
C     IF(.NOT. ELSTIC) THEN                                             ELE04920
          DO 93 I=1,12                                                  ELE04930
            P(I)=FT(I)+F(I)                                             ELE04940
            D(I)=RT(I)+R(I)                                             ELE04950
            PL(I)=FT(I)                                                 ELE04960
            DL(I)=RT(I)                                                 ELE04970
   93     CONTINUE                                                      ELE04980
C                                                                       ELE04990
C     ----- TRANSFER PERMENANT HYSTERESIS DATA TO TEMPERARY STORAGE     ELE05000
        ISE2=ISE/2                                                      ELE05010
        DO 522 I=1,ISE2                                                 ELE05020
  522     SE(I+ISE2)=SE(I)                                              ELE05030
C                                                                       ELE05040
        MOPT=3                                                          ELE05050
        CALL MATLIB                                                     ELE05060
     &  ( MOPT ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,     ELE05070
     &    P    ,PL   ,D    ,DL   ,V    ,VL   ,LELEM ,LMAT  ,           ELE05080
     &    MAT  ,MATTYP,SE(ISE2+1),IELNO,DAMAGE)                        ELE05090
        NEWK=.TRUE.                                                     ELE05100
C     ENDIF                                                             ELE05110
C                                                                       ELE05120
C                                                                       ELE05130
C------ CALCULATE THE UNBALACED MEMBER FORCES ON THE JOINT              ELE05140
C       THE EXACT MEMBER FORCES ARE DETERMINED FROM HYSTERESIS MODEL    ELE05150
C       BASED ON DISPLACEMENT D. BECAUSE NO HYSTERESIS MODEL FOR MAT12, ELE05160
C       THE MEMBER FUB ARE NOT CALCULATED IN THIS ELEMENT               ELE05170
        DO 440 I=1,3                                                    ELE05180
          I1=IDOF(JOINTI,I  )                                           ELE05190
          I2=IDOF(JOINTI,I+3)                                           ELE05200
          J1=IDOF(JOINTJ,I  )                                           ELE05210
          J2=IDOF(JOINTJ,I+3)                                           ELE05220
          DO 440 J=1,3                                                  ELE05230
            FUB(I1)=FUB(I1)+CTS(I,J)*0                                  ELE05240
            FUB(I2)=FUB(I2)+ BS(I,J)*0                                  ELE05250
            FUB(I2)=FUB(I2)+CTS(I,J)*0                                  ELE05260
            FUB(J1)=FUB(J1)+CTE(I,J)*0                                  ELE05270
            FUB(J2)=FUB(J2)+ BE(I,J)*0                                  ELE05280
  440       FUB(J2)=FUB(J2)+CTE(I,J)*0                                  ELE05290
C                                                                       ELE05300
  491 FORMAT (/'     STABILITY ELEMENT  FORCES...',5X/7X,A/,            ELE05310
     & '    ELEMENT LOAD         NODE',7X,'AXIAL',11X,'FT',13X,'FZ',    ELE05320
     & 11X,'TORSION',10X,'MY',13X,'MZ    FLP,SP ')                     ELE05330
  492 FORMAT (/,I10,I5,2X,'FORCE',I6,1P,6G15.6,                         ELE05340
     &        /15X    ,2X,'FORCE',I6,   6G15.6,1X,G10.3)               ELE05350
  493 FORMAT (/,I10,I5,2X,'DISPL',I6,1P,6G15.6,                         ELE05360
     &        /15X    ,2X,'DISPL',I6,   6G15.6,1X,G10.3)               ELE05370
  494 FORMAT (7X,A)                                                     ELE05380
      RETURN                                                            ELE05390
C----------------------- DETERMINE TOTAL FORCES ----------------------- ELE05400
  500 CONTINUE                                                          ELE05410
      IF (PRINT.AND.FIRST) WRITE (6,491) STEPID                         ELE05420
      IF( ICOMN .NE. 1 .AND. ICOMN .NE. 2 ) THEN                        ELE05430
C                                                                       ELE05440
C------ TOTAL FORCES AND DISPLACEMENTS                                 ELE05450
      DO 510 I=1,12                                                     ELE05460
C       IF (BTEST(IBUG,7)) WRITE(6,520) I,R(I),RT(I),F(I),FT(I)         ELE05470
        FT(I)=FT(I)+F(I)                                                ELE05480
        RT(I)=RT(I)+R(I)                                                ELE05490
  510 CONTINUE                                                          ELE05500
C                                                                       ELE05510
      DO 512 I=1,12                                                     ELE05520
        IF ( ABS(FT(I)).GT.ABS(FMAX(I*12-12+I)) ) THEN                  ELE05530
          DO 511 J=0,11                                                 ELE05540
  511       FMAX(I*12-11+J)=FT(J+1)                                     ELE05550
        ENDIF                                                           ELE05560
  512 CONTINUE                                                          ELE05570
  520 FORMAT ('   I:',I3,' R:',1P,G12.4,' RT:',G12.4,                   ELE05580
     &                   ' F:',1P,G12.4,' FT:',G12.4)                 ELE05590
      ELSE IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                    ELE05600
        DO 513 I=1,12                                                   ELE05610
  513     FT(I)=ASAF(I)                                                 ELE05620
      ENDIF                                                             ELE05630
C                                                                       ELE05640
      LOAD=1                                                            ELE05650
      ISE2=ISE/2                                                        ELE05660
      SP=SE(ISE2+84)                                                    ELE05670
      FLP=SE(ISE2+98)                                                   ELE05680
C     IF (BTEST(IBUG,0).AND.PRINT)                                      ELE05690
      IF (         PRINT)                                               ELE05700
     &                   WRITE (6,493) IELNO,LOAD,NODEI,(RT(J),J=1,6),  ELE05710
     &                                            NODEJ,(RT(J),J=7,12), ELE05720
     &                                            FLP                 ELE05730
      IF (         PRINT) WRITE (6,492) IELNO,LOAD,NODEI,(FT(J),J=1,6),ELE05740
     &                                            NODEJ,(FT(J),J=7,12), ELE05750
     &                                            SP                  ELE05760
C                                                                       ELE05770
C     ----- TRANSFER TEMPERARY HYSTERESIS DATA TO PERMANENT STORAGE     ELE05780
        ISE2=ISE/2                                                      ELE05790
C       IF( .NOT. ELSTIC) THEN                                          ELE05800
          DO 415 I=1,ISE2                                               ELE05810
  415       SE(I)=SE(I+ISE2)                                            ELE05820
C       ENDIF                                                           ELE05830
C                                                                       ELE05840
      RETURN                                                            ELE05850
C                                                                       ELE05860
C------------------------- DETERMINE FIXED END FORCES, AND ADD TO FORCE-ELE05870
  600 CONTINUE                                                          ELE05880
C-----FIXED END FORCES ARE NOT AVAILABLE FOR STABILITY ELEMENT          ELE05890
C                                                                       ELE05900
      RETURN                                                            ELE05910
C                                                                       ELE05920
C---------------------- WRITE MEMBER FORCES TO OUTPUT FILE ------------ELE05930
  700 CONTINUE                                                          ELE05940
C                                                                       ELE05950
      SP=SE(84)                                                         ELE05960
      FLP=SE(98)                                                        ELE05970
      WRITE (IUNIT,710) NODEI,(FT(J),J=1,6),NODEJ,(FT(J),J=7,12),SP     ELE05980
      WRITE (IUNIT,710) NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12),FLP    ELE05990
  710 FORMAT (I6,1P,6G12.5/,                                            ELE06000
     &        I6,1P,6G12.5,1X,G10.3)                                    ELE06010
C                                                                       ELE06020
      RETURN                                                            ELE06030
C                                                                       ELE06040
C--------------------- READ AND PRINT MEMBER FORCES FROM OUTPUT FILE ---ELE06050
  800 CONTINUE                                                          ELE06060
      BACKSPACE(IUNIT)                                                  ELE06070
      READ (IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT                        ELE06080
      WRITE (6,805)                                                     ELE06090
      ISTEP=-IWRITE                                                     ELE06100
  810 READ (IUNIT,815,END=830)                                          ELE06110
     &   NODEI,(FT(J),J=1,6),NODEJ,(FT(J),J=7,12),SP                   ELE06120
      READ (IUNIT,815,END=830)                                          ELE06130
     &   NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12),FLP                  ELE06140
  815 FORMAT (I6,6G12.5/,                                               ELE06150
     &        I6,6G12.5,1X,G10.3)                                       ELE06160
        ISTEP=ISTEP+IWRITE                                              ELE06170
        T=TO+DT*ISTEP                                                   ELE06180
CCC     MODU=MOD(ISTEP,INC)                                             ELE06190
        IF (MOD(ISTEP,INC).EQ.0) THEN                                   ELE06200
          WRITE (6,820) ISTEP,T,NODEI,(FT(J),J=1,6),                    ELE06210
     &                          NODEJ,(FT(J),J=7,12),SP                 ELE06220
          WRITE (6,920)           (RT(J),J=1,12),FLP                    ELE06230
        ENDIF                                                           ELE06240
        GO TO 810                                                       ELE06250
C                                                                       ELE06260
  805 FORMAT (/'     STABILITY ELEMENT FORCES...',/                     ELE06270
     & '    STEP    TIME         NODE',7X,'AXIAL',11X,'FY',13X,'FZ',    ELE06280
     & 11X,'TORSION',10X,'MY',13X,'MZ    FLP,SP')                      ELE06290
  820 FORMAT (/,I10,1P,G15.5,0P,I5,1P,6G15.6,                           ELE06300
     &        /25X,                 I6,   6G15.6,1X,G10.3)             ELE06310
  920 FORMAT ( 25X,'DISP  ',         1P,6G15.6,                         ELE06320
     &        /31X,                    6G15.6,1X,G10.3)               ELE06330
C                                                                       ELE06340
  830 RETURN                                                            ELE06350
C                                                                       ELE06360
C--------------------- DETERMINE THE LENGTH OF THE MATERIAL DATA        ELE06370
  900 CONTINUE                                                          ELE06380
      ISE=0                                                             ELE06390
        MAT  = RINPUT( 1)                                               ELE06400
        CALL MATLIB                                                     ELE06410
     &  ( 0    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,EESE,EPSE,DUCT,EXCR,     ELE06420
     &    FT   ,F    ,RT   ,R    ,V    ,VL   ,LELEM ,LMAT  ,           ELE06430
     &    MAT  ,MATTYP,SE,IELNO,DAMAGE)                                ELE06440
        ISE=ISE+LELEM                                                   ELE06450
C                                                                       ELE06460
C----- DOUBLE THE STORAGE, TO ALLOW FOR TEMPOARY VALUES                 ELE06470
        ISE=ISE*2                                                       ELE06480
C                                                                       ELE06490
      RETURN                                                            ELE06500
C                                                                       ELE06510
C--------------------- DETERMINE THE TOTAL STRAIN ENERGY                ELE06520
 1000 CONTINUE                                                          ELE06530
      EESE=0                                                            ELE06540
      EPSE=0                                                            ELE06550
C                                                                       ELE06560
```

```
C     /* ENERGY FORMULATION FOR ELE12 ELEMENT IS NOT AVAILABLE */
C     IF(ELSTIC) THEN
        DO 1010 I=1,12
 1010     EESE=EESE + (FT(I)+F(I))*(R(I)+RT(I))/2.
C     ENDIF
C
      ENGDAT(1) = EESE
      ENGDAT(2) = 0
C
      RETURN
C
C******************** DUCTILITIES AND EXCURSION RATIOS
C                     NOT AVAILABLE CURRENTLY
 1100 RETURN
C
C
C******************** PRINT MAXIMUM ELEMENT LOADS
C
 1200 CONTINUE
      LOAD2=0
      IF (FIRST) WRITE(6,494)' COLUMN ( LOAD) REPRESENTS THE DOF WHICH'
     &          //' HAS MAXIMUM VALUE'
      IF (FIRST) WRITE (6,491) ' MAXIMUM VALUES FOR ALL STEPS  '
     &          //' NOTE: MAXIMUM VALUES WITH THE OTHER DOFS FORCES'
     &          //' ARE PRINT OUT AT THE SAME TIME'
      DO 309 I=0,11
         LOAD2=I+1
         IF( FMAX(I*12+1+I) .EQ. 0. ) GO TO 309
         WRITE (6,492) IELNO,LOAD2,NODEI,(FMAX(I*12+J),J=1,6),
     &                             NODEJ,(FMAX(I*12+J),J=7,12)
  309 CONTINUE
C
      RETURN
C
C
C******************** RESET ELEMENT FORCES
C
 1300 CONTINUE
C----- ZERO MATRICIES
       DO 1340 I=1,12
         FMAX(I*12-11)=0
         F (I)=0
         FT(I)=0
         R (I)=0
 1340    RT(I)=0
       DO 1310 I=1,ISE
 1310    SE(I)=0
       SE(79)=LENGTH
C
      WRITE (6,1330) IELNO
 1330 FORMAT (' STABILITY ELEMENT ..... ELEMENT #',I6,
     &  '  INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO')
        FALSE=.FALSE.
        LSTTIP=0
        CALL MATLIB
     &  ( 2    ,LSTTIP ,FALSE ,IREL,FALSE,NAME,ESSE,EPSE,DUCT,EXCR,
     &    FT   ,F      ,RT    ,R     ,V     ,VL    ,LELEM ,LMAT  ,
     &    MAT  ,MATTIP,SE,IELNO,DAMAGE)
C
      RETURN
C
C******************** DAMAGE INDEX
C
 1400 DAMAGE = 0.
C
      RETURN
C
      END
@PROCESS OPT(0) XREF MAP GOSTMT
C*====================================================================*
C         DEBUG UNIT(8),TRACE,SUBCHK,INIT,SUBTRACE
C         END DEBUG
C*====================================================================*
      SUBROUTINE ELE14
     &  (IOPT       ,IUNIT      ,NEWK       ,FIRST      ,PRINT      ,
     &   NDOF       ,NLOAD      ,MAXNOD     ,NNODE      ,NCOS       ,
     &   ID         ,IDOF       ,COORD      ,COSINE     ,JTCOS      ,
     &   CONST      ,DISPL      ,VELOC      ,FORCE      ,FUB        ,
     &   IREL       ,BUG        ,IBUG       ,ACCEL      ,KGDATA     ,
     &   GRAV       ,PGEOM      ,ELSTIC     ,NAME       ,STEPID     ,
     &   IELNO      ,RINPUT     ,AXIAL      ,ESE        ,PSE        ,
     &   INC        ,DAMAGE     ,DUCFAG     ,
     &   IELTYP     ,IELDOF     ,ISES       ,ISEA1      ,ISEA2      ,
     &   ISE        ,J1         ,J2         ,J3         ,J4         ,
     &   MATS       ,MATA1      ,MATA2      ,BLANK      ,R          ,
     &   RT         ,F          ,FT         ,V          ,VT         ,
     &   PKG        ,LKG        ,LM         ,LMKG       ,A          ,
     &   SS         ,SA1        ,SA2        ,TEMP1      ,TEMP2      ,
     &   KGDOF      ,LENGTH     ,FMAX       ,SE         ,STIFF      )
      COMMON /RSA/ICOMN,RSAF(12)
      REAL ICOMN
      REAL LWP,MQD
C
      DOUBLE PRECISION SX,SXX,SY,SXY,SN
      LOGICAL ERR,FIRST,PRINT,BUG,FALSE,NEWK,ELSTIC ,AXIAL,BTEST
      CHARACTER*80 NAME
      CHARACTER*(*) STEPID
      DOUBLE PRECISION  SAT,DSTIFF(600)
      DIMENSION  IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS)
      DIMENSION CONST(MAXNOD,6),    ID(MAXNOD)  ,JTCOS(MAXNOD)
      DIMENSION SAT(3,24),STIFF(600),KGDATA(3) ,SAT2(6,24)
      DIMENSION AT(24,6),A(24,3),S(6,6),LM(24),AC(6,3),LMKG(24)
      DIMENSION CT(3,3),BS(3,3),VX(3),VT(3),VZB(3),VXT(3),VZT(3),VZB(3)
      DIMENSION SE(ISE),A11(6,3),A22(6,3),A44(6,3),A33(6,3)
      DIMENSION F(3),R(3),V(3),FT(3),RT(3),VT(3),P(3),AD(100),AV(50)
      DIMENSION DISPL(NDOF,NLOAD),FUB(NDOF),EFUB(24),FMAX(6)
      DIMENSION VELOC(NDOF)
      DIMENSION FORCE(NDOF,NLOAD)
      DIMENSION RINPUT(100),WORK(1),DUCT(3,3),EXCR(6,3)
      REAL LENGTH
      DIMENSION MD(25)
      DATA MD/ 1,  2,  4,  7, 11, 16, 22, 29, 37, 46, 56, 67,
     &        79, 92,106,121,137,154,172,191,211,232,254,277,301/
C
C*====================================================================*
C VARIABLES:
C--------------------------------------------------------------------*
C---------- GLOBAL VARIABLES
C IOPT   = 1, INITIALIZE BEAM COLUMN ELEMENT
C        = 2, CALCULATE GEOMETRIC STIFFNESS
C        = 3, FORM STIFFNESS
C        = 4, CALCULATE INCREMENTAL FORCES
C        = 5, CALCULATE TOTAL FORCES AND ENERGIES
C IUNIT  = OUTPUT FILE UNIT # FOR PRINTING OUTPUT
C ERR    = ERR OR FLAG, IF AN ERR OR HAS OCCURED, ERR=.TRUE.
C FIRST  = FLAG, FIRST=.TRUE., PRINT HEADERS
C PRINT  = FLAG, PRINT=.TRUE., PRINT DATA
C NDOF   = # OF GLOBAL DOF
C NLOAD  = # OF LOAD COMBINATIONS
C MAXNOD = # ROW DIMENSION OF NODE ARRAY
C NNODE  = # OF NODES
C ID     = ARRAY OF EXTERNAL NODE NUMBERS
C IDOF   = ARRAY OF DEGREES OF FREEDOM
C COORD  = ARRAY OF COORDINATES
C COSINE = ARRAY OF DIRECTION COSINES OF NODES
C CONST  = ARRAY OF CONSTRAINT TRANSFORMATIONS
C DISPL  = GLOBAL DISPLACEMENT MATRIX
C Z      = REAL GLOBAL STORTAG VECTOR
C NZ     = INTEGER GLOBAL STORTAG VECTOR
C IZMAT  = LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR
C NAME   = ELEMENT NAME
C IELNO  = ELEMENT NUMBER
C RINPUT = INPUT DATA
C STIFF  = OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)
C LM     = OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX
C---------- ELEMENT VARIABLES
C H,Q,AI,AJ,C,BI,BJ,S22,S26,S212,S33,S35,S311,D
C        = INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX
C PKG    = INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX...
C R      = INCRIMENTAL DISPLACEMENTS
C RT     = TOTAL DISPLACEMENTS
C F      = INCRIMENTAL FORCES
C FT     = TOTAL FORCES
C LM     = LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX
C CTS    = LOCAL TO GLOBAL ROTATION MATRIX START
C CTE    = LOCAL TO GLOBAL ROTATION MATRIX END
C BS     = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX START
C BE     = LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END
C STIFF  = ELEMENT STIFFNESS
C IELTYP = ELEMENT TYPE = 10
C NODEI  = EXTERNAL JOINT NUMBER, END I
C NODEJ  = EXTERNAL JOINT NUMBER, END J
C JOINTI = INTERNAL JOINT NUMBER, END I
C JOINTJ = INTERNAL JOINT NUMBER, END J
C---------- TEMPORARY VARIABLES
C VX     = VECTOR DEFINING THE ELEMENT X AXIS
C VT     = VECTOR DEFINING THE ELEMENT Y AXIS
C COSLOC = LOCAL COSINE MATRIX
C A      = GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX
C S      = ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART
C SAT    = PRODUCT OF S*A'
C ASAT   = PRODUCT OF A*SAT
C--------------------------------------------------------------------*
C
C
C-------- CHOOSE OPTION
        IF (IOPT.LT.1 .OR.IOPT.GT.14)
     &  WRITE (6,*) 'INVALID OPTION IN ELE-14, IOPT=',IOPT
C
        GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,
     &                                          1300,1400),IOPT
C
  100 CONTINUE
C
C----- INTERPERTATE INPUT DATA
      IELTYP= 14
      MATS  = RINPUT( 1)
      MATA1 = RINPUT( 2)
      MATA2 = RINPUT( 3)
      NODE1 = RINPUT( 4)
      NODE2 = RINPUT( 5)
      NODE3 = RINPUT( 6)
      NODE4 = RINPUT( 7)
      ALPIN = RINPUT( 8)
      PKG   = RINPUT( 9)
C
C----- GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S
      J1=IQUICK(NODE1,ID,NNODE)
      J2=IQUICK(NODE2,ID,NNODE)
      J3=IQUICK(NODE3,ID,NNODE)
      J4=IQUICK(NODE4,ID,NNODE)
C
C-------------------------------------------------------------------
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX
C-------------------------------------------------------------------
C----- DEFINE VECTOR X, AT BOTTOM OF THE WALL
      VXB(1)=COORD(J4,1)-COORD(J3,1)
      VXB(2)=COORD(J4,2)-COORD(J3,2)
      VXB(3)=COORD(J4,3)-COORD(J3,3)
C
C----- DEFINE VECTOR X, AT TOP OF THE WALL
      VXT(1)=COORD(J1,1)-COORD(J2,1)
      VXT(2)=COORD(J1,2)-COORD(J2,2)
      VXT(3)=COORD(J1,3)-COORD(J2,3)
C
C----- CALCULATE Y VECTOR ...
      VT(1)=(COORD(J1,1)+COORD(J2,1)-COORD(J3,1)-COORD(J4,1))/2.
      VT(2)=(COORD(J1,2)+COORD(J2,2)-COORD(J3,2)-COORD(J4,2))/2.
      VT(3)=(COORD(J1,3)+COORD(J2,3)-COORD(J3,3)-COORD(J4,3))/2.
C
C----- CALCULATE AVERAGE LENGTH,ALPHA AND BETA...
      LENGTH=SQRT(VT(1)**2+VT(2)**2+VT(3)**2)
      SE(ISES+77)=LENGTH
      IF (LENGTH.LE.0) THEN
         WRITE(6,101) IELNO,LENGTH
  101    FORMAT(1X,20('*'),'ERROR - LENGTH IS LE 0',/
     &    5X,'IELNO=',I5,5X,5X,'LENGTH=',1P,G15.6/
     &    5X,'REVISE INPUT.............................'/)
         LENGTH=1.0
      ENDIF
C
      ALPHA=ALPIN*LENGTH
      BETA = LENGTH-ALPHA
C
C----- NORMALISE Y VECTOR ...
      VT(1)=VT(1)/LENGTH
      VT(2)=VT(2)/LENGTH
      VT(3)=VT(3)/LENGTH
C
C----- CALCULATE THE WIDTH ...
      CALL VCROSS(VZT,VXT,VT)
      CALL VCROSS(VZB,VXB,VT)
      WT = SQRT(VZT(1)**2 + VZT(2)**2 + VZT(3)**2)
      WB = SQRT(VZB(1)**2 + VZB(2)**2 + VZB(3)**2)
      W  = (WT+WB)/2.0
C
C----- NORMALIZE Z VECTOR ...
      VZB(1)=VZB(1)/W
      VZB(2)=VZB(2)/W
      VZB(3)=VZB(3)/W
      SE(ISES+78)=W
C
C----- CHECK FOR TWIST
      TWIST=ACOS(VCOS(VZT,VZB))*57.2958
      IF (ABS(TWIST).GT. 5.0) WRITE (6,102) TWIST
  102 FORMAT(5X,10('*'),' WARNING THE WALL IS TWISTED=',1P,G12.4,
     &    ' DEGREES, REVISE INPUT AND RERUN')
C
C----- CALCULATE VX, AND NORMALIZE
      CALL VCROSS(VX,VT,VZB)
      WX=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)
      VX(1)=VX(1)/WX
      VX(2)=VX(2)/WX
      VX(3)=VX(3)/WX
C
C----- CHECK FOR TWIST
      IF ( BTEST(IBUG,7) ) THEN
         WRITE (6,*) 'JOINTS',J1,J2,J3,J4
         WRITE (6,*) '                VX',VX
         WRITE (6,*) '                VT',VT
         WRITE (6,*) '                VZ',VZB
         WRITE (6,*) 'LENGTH,ALPHA,BETA',LENGTH,ALPHA,BETA
      ENDIF
C
C----- GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES
C--- JOINT #1
         CALL ROTXYZ(VX,VT,COSINE(1,1,JTCOS(J1)),CT,
     &              0.,0.,0.,BS,  CONST(J1,1),MAXNOD)
         DO 21 I=1,3
         DO 21 J=1,3
           A11(I  ,J)=CT(I,J)
   21      A11(I+3,J)=BS(I,J)
C--- JOINT #2
         CALL ROTXYZ(VX,VT,COSINE(1,1,JTCOS(J2)),CT,
```

```
     &                    0.,0.,0..BS,  CONST(J2,1),MAXNOD)
          DO 22 I=1,3
          DO 22 J=1,3
            A22(I  ,J)=CT(I,J)
22        A22(I+3,J)=BS(I,J)
C--- JOINT #3
          CALL ROTXYZ(VX,VY,COSINE(1,1,JTCOS(J3)),CT,
     &                    0.,0.,0.,BS,  CONST(J3,1),MAXNOD)
          DO 23 I=1,3
          DO 23 J=1,3
            A33(I  ,J)=CT(I,J)
23        A33(I+3,J)=BS(I,J)
C--- JOINT #4
          CALL ROTXYZ(VX,VY,COSINE(1,1,JTCOS(J4)),CT,
     &                    0.,0.,0.,BS,  CONST(J4,1),MAXNOD)
          DO 24 I=1,3
          DO 24 J=1,3
            A44(I  ,J)=CT(I,J)
24        A44(I+3,J)=BS(I,J)
C
          IF ( BTEST(IBUG,7) ) THEN
               WRITE (6,*) 'JOINT #1,  J1',J1
               CALL WMATRX(A11  , 6, 3,'A11  ')
               WRITE (6,*) 'JOINT #2,  J2',J2
               CALL WMATRX(A22  , 6, 3,'A22  ')
               WRITE (6,*) 'JOINT #3,  J3',J3
               CALL WMATRX(A33  , 6, 3,'A33  ')
               WRITE (6,*) 'JOINT #4,  J4',J4
               CALL WMATRX(A44  , 6, 3,'A44  ')
          ENDIF
C
C----- ZERO MATRICIES
      DO 25 I=1,3
         FMAX(I  )= 0
         FMAX(I+3)= 0
         F (I)=0
         FT(I)=0
         R (I)=0
         RT(I)=0
         V (I)=0
25       VT(I)=0
      DO 27 I=1,24
         LM  (I)=0
         LMKG(I)=0
         DO 26 J=1,3
26          A(I,J)=0.00
         DO 27 K=1,6
27          AT(I,K)=0.00
C
C----- ASSEMBLE TRANSFORMATION MATRIX A = A5 X A4 X A3
C  NOTE: AT IS DIMMENSIONED AS A 24X6 MATRIX. BUT ONLY THE FIRST IELDOF
C        ROW'S CONTAIN INFORMATION. THE BALANCE OF THE MATRIX IS
C        UNDEFINED.  THE GLOBAL DOF COORESPONDING TO THE ROW'S OF A
C        ARE STORED IN VECTOR LM.  AGAIN, ONLY THE FIRST IELDOF ROW'S
C        OF LM ARE DEFINED.
      IELDOF=0
      DO 50 I=1,4
      DO 50 J=1,6
C
C               REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF...
      IF ( (IDOF(J1,J).EQ.IDOF(J4,J)) .OR.
     &     (IDOF(J1,J).EQ.IDOF(J3,J)) .OR.
     &     (IDOF(J2,J).EQ.IDOF(J4,J)) .OR.
     &     (IDOF(J2,J).EQ.IDOF(J3,J)) ) GO TO 50
C
          IELDOF=IELDOF+1
          IF (I.EQ.1) THEN
              LMTI=IDOF(J1,J)
              AT(IELDOF,1)=A11(J,1)
              AT(IELDOF,2)=A11(J,2)
          ELSE IF (I.EQ.2) THEN
              LMTI=IDOF(J2,J)
              AT(IELDOF,3)=A22(J,2)
          ELSE IF (I.EQ.3) THEN
              LMTI=IDOF(J3,J)
              AT(IELDOF,4)=A33(J,2)
          ELSE IF (I.EQ.4) THEN
              LMTI=IDOF(J4,J)
              AT(IELDOF,5)=A44(J,1)
              AT(IELDOF,6)=A44(J,2)
          ENDIF
C
          LM(IELDOF)=LMTI
          IF((AT(IELDOF,1).EQ.0).AND.(AT(IELDOF,2).EQ.0) .AND.
     &       (AT(IELDOF,3).EQ.0).AND.(AT(IELDOF,4).EQ.0) .AND.
     &       (AT(IELDOF,5).EQ.0).AND.(AT(IELDOF,6).EQ.0)) THEN
              IELDOF=IELDOF-1
          ELSE IF (IELDOF.GT.1) THEN
             DO 49  II=1,IELDOF-1
               IF (LM(II).EQ.LM(IELDOF)) THEN
                 DO 48 JJ=1,6
                   AT(II,JJ)=AT(II,JJ)+AT(IELDOF,JJ)
48                 AT(IELDOF,JJ)=0
                 IELDOF=IELDOF-1
                 GO TO 50
               ENDIF
49           CONTINUE
          ENDIF
C
50    CONTINUE
C
C------ A2A1 FOR STIFFNESS ONLY...
      AC(1,1)=  1.
      AC(2,1)=  ALPHA/W
      AC(3,1)= -ALPHA/W
      AC(4,1)= -BETA /W
      AC(5,1)= -1.
      AC(6,1)=  BETA /W
      AC(1,2)=  0.
      AC(2,2)=  1.
      AC(3,2)=  0.
      AC(4,2)=  0.
      AC(5,2)=  0.
      AC(6,2)= -1.
      AC(1,3)=  0.
      AC(2,3)=  0.
      AC(3,3)=  1.
      AC(4,3)=  0.
      AC(5,3)=  0.
      AC(6,3)=  0.
C
      DO 55 I=1,IELDOF
      DO 55 J=1,3
        A(I,J)=0.
        DO 55 K=1,6
55        A(I,J)=A(I,J)+AT(I,K)*AC(K,J)
C
      IF ( BTEST(IBUG,7) ) THEN
           CALL WMTRX2( AT   ,IELDOF,6,' AT   ',24)
           CALL WMATRX( AC   , 6,3,' AC   ')
           CALL WMTRX2( A    ,IELDOF,3,' A    ',24)
      ENDIF
C
C
C------ GET SPRING STIFFNESS
       FALSE=.FALSE.
       LSTTYP=0
       CALL MATLIB
     & ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &   ESES,PSES,DUCT(1,1),EXCR(1,1),
     &   FT   ,F      ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,
     &   MATS  ,MATTYP,SE(ISES),IELNO,DAMAGE)
          CALL MATLIB                                                    ELE03590
     & ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                          ELE03600
     &   ESEA1,PSEA1,DUCT(1,2),EXCR(1,2),                               ELE03610
     &   FT   ,F      ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,      ELE03620
     &   MATA1 ,MATTYP,SE(ISEA1),IELNO,DAMAGE)                          ELE03630
          CALL MATLIB                                                    ELE03640
     & ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                          ELE03650
     &   ESEA2,PSEA2,DUCT(1,3),EXCR(1,3),                               ELE03660
     &   FT   ,F      ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,      ELE03670
     &   MATA2 ,MATTYP,SE(ISEA2),IELNO,DAMAGE)                          ELE03680
C                                                                        ELE03690
C TO CHECK KEL,CYCLE,CO1,CO2,CO5                                         ELE03700
C                                                                        ELE03710
          ESE=0.                                                         ELE03720
          PSE=0.                                                         ELE03730
C                                                                        ELE03740
C------- SET UP THE LOCAL STIFFNESS MATRIX                               ELE03750
          SS= SE(ISES)                                                   ELE03760
          SA1= SE(ISEA1)                                                 ELE03770
          SA2= SE(ISEA2)                                                 ELE03780
          IF ( BTEST(IBUG,7) ) THEN                                      ELE03790
            WRITE (6,*) 'SHEAR   STIFFNESS KS=',SS                       ELE03800
            WRITE (6,*) 'AXIAL   STIFFNESS KA=',SA1                      ELE03810
          ENDIF                                                          ELE03820
C                                                                        ELE03830
C------ CALCULATE SAT                                                   ELE03840
          SAL1=SA1/(LENGTH**2)                                           ELE03850
          SAL2=SA2/(LENGTH**2)                                           ELE03860
          SSL=SS                                                         ELE03870
          DO 60 J=1,IELDOF                                               ELE03880
            SAT(1,J)=SSL*A(J,1)                                          ELE03890
            SAT(2,J)=SAL1*A(J,2)                                         ELE03900
   60       SAT(3,J)=SAL2*A(J,3)                                         ELE03910
C                                                                        ELE03920
C------ CALCULATE ASAT AND CONVERT TO HALF STORAGE MODE  BY COLUMNS      ELE03930
C                                                                        ELE03940
C          1  3  6 10 15 21 28 36 45 55 66 78     300.  NUMBER INDICATES ELE03950
C             2  5  9 14 20 27 35 44 54 65 77      .    LOCATION OF DATA ELE03960
C                4  8 13 19 26 34 43 53 64 76      .    IN ARRAY STIFF   ELE03970
C                   7 12 18 25 33 42 52 63 75      .                     ELE03980
C                     11 17 24 32 41 51 62 74      .                     ELE03990
C                        16 23 31 40 50 61 73      .                     ELE04000
C                           22 30 39 49 60 72      .                     ELE04010
C                              29 38 48 59 71      .                     ELE04020
C                                 37 47 58 70      .                     ELE04030
C                                    46 57 69      .                     ELE04040
C                                       56 68      .                     ELE04050
C                                          67      .                     ELE04060
C                                             .    .                     ELE04070
C                                              .   .                     ELE04080
C                                               .  .                     ELE04090
          L=0                                                            ELE04100
          DO 70 J=1,IELDOF                                               ELE04110
            DO 70 I=J,1,-1                                               ELE04120
            L=L+1                                                        ELE04130
            DSTIFF(L)=0                                                  ELE04140
            DO 70 K=1,3                                                  ELE04150
   70         DSTIFF(L)=DSTIFF(L)+A(I,K)*SAT(K,J)                       ELE04160
          DO 71 J=1,L                                                    ELE04170
   71       STIFF(J)=DSTIFF(J)                                          ELE04180
C                                                                        ELE04190
          IF ( BTEST(IBUG,7) ) THEN                                      ELE04200
            WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)        ELE04210
            CALL LMATRX(STIFF,MD,IELDOF,(L+1),'WALL STIFFNESS')          ELE04220
          ENDIF                                                          ELE04230
C                                                                        ELE04240
C-----------------------------------------------------------------       ELE04250
C  ASSEMBLE GEOMETRIC STIFFNESS MATRIX FOR A UNIT LOAD...                ELE04260
C-----------------------------------------------------------------       ELE04270
          LKG=L+1                                                        ELE04280
          IF ( PKG.EQ.0 .AND. KGDATA(1).NE.2 ) GO TO 180                 ELE04290
          IF (KGDATA(1).EQ.0 .OR. KGDATA(2).EQ.0 ) GO TO 180             ELE04300
          AXIAL=.TRUE.                                                   ELE04310
C                                                                        ELE04320
C---------- DETERMINE THE AXIAL LOAD TO BE USED FOR KG...                ELE04330
            IF (KGDATA(1).EQ.1) THEN                                     ELE04340
                PGEOM =  PKG                                             ELE04350
            ELSE IF (KGDATA(1).EQ.2) THEN                                ELE04360
                PGEOM = -F(2)                                            ELE04370
            ENDIF                                                        ELE04380
C                                                                        ELE04390
C----- ASSEMBLE TRANSFORMATION MATRIX A                                  ELE04400
C  NOTE: AT IS DIMMENSIONED AS A 12X6 MATRIX, BUT ONLY THE FIRST IELDOF  ELE04410
C        ROW'S CONTAIN INFORMATION. THE BALANCE OF THE MATRIX IS         ELE04420
C        UNDEFINED.  THE GLOBAL DOF COORESPONDING TO THE ROW'S OF A      ELE04430
C        ARE STORED IN VECTOR LM.  AGAIN, ONLY THE FIRST IELDOF ROW'S    ELE04440
C        OF LM ARE DEFINED.                                              ELE04450
          DO 140 I=1,24                                                  ELE04460
          DO 140 K=1,6                                                   ELE04470
  140       AT(I,K)=0.00                                                 ELE04480
          KGDOF=0                                                        ELE04490
          DO 145 I=1,4                                                   ELE04500
          DO 145 J=1,6                                                   ELE04510
C                                                                        ELE04520
            KGDOF =KGDOF +1                                              ELE04530
            IF (I.EQ.1) THEN                                             ELE04540
                LMTI=IDOF(J1,J)                                          ELE04550
                AT(KGDOF,1)=A11(J,1)                                     ELE04560
                AT(KGDOF,2)=A11(J,2)                                     ELE04570
            ELSE IF (I.EQ.2) THEN                                        ELE04580
                LMTI=IDOF(J2,J)                                          ELE04590
                AT(KGDOF,3)=A22(J,2)                                     ELE04600
            ELSE IF (I.EQ.3) THEN                                        ELE04610
                LMTI=IDOF(J3,J)                                          ELE04620
                AT(KGDOF,4)=A33(J,2)                                     ELE04630
            ELSE IF (I.EQ.4) THEN                                        ELE04640
                LMTI=IDOF(J4,J)                                          ELE04650
                AT(KGDOF,5)=A44(J,1)                                     ELE04660
                AT(KGDOF,6)=A44(J,2)                                     ELE04670
            ENDIF                                                        ELE04680
            LMKG(KGDOF)=LMTI                                             ELE04690
            IF((AT(KGDOF,1).EQ.0).AND.(AT(KGDOF,2).EQ.0).AND.            ELE04700
     &         (AT(KGDOF,3).EQ.0).AND.(AT(KGDOF,4).EQ.0).AND.            ELE04710
     &         (AT(KGDOF,5).EQ.0).AND.(AT(KGDOF,6).EQ.0)) THEN           ELE04720
                KGDOF=KGDOF-1                                            ELE04730
            ELSE IF (KGDOF.GT.1) THEN                                    ELE04740
               DO 149  II=1,KGDOF-1                                      ELE04750
                 IF (LMKG(II).EQ.LMKG(KGDOF)) THEN                       ELE04760
                   DO 148 JJ=1,6                                         ELE04770
                     AT(II,JJ)=AT(II,JJ)+AT(KGDOF,JJ)                   ELE04780
  148                AT(KGDOF,JJ)=0                                      ELE04790
                   KGDOF=KGDOF-1                                         ELE04800
                   GO TO 145                                             ELE04810
                 ENDIF                                                   ELE04820
  149          CONTINUE                                                  ELE04830
            ENDIF                                                        ELE04840
C                                                                        ELE04850
  145     CONTINUE                                                       ELE04860
C                                                                        ELE04870
C----- LOCAL GEOMETRIC STIFFNESS MATRIX                                  ELE04880
          DO 150 I2=1,12                                                 ELE04890
          DO 150 JI=1,12                                                 ELE04900
  150       S(I2,JI)=0.0                                                 ELE04910
          S(1,1)=  1.0/LENGTH                                            ELE04920
          S(2,2)=  0.5/LENGTH                                            ELE04930
          S(3,3)=  0.5/LENGTH                                            ELE04940
          S(4,4)=  0.5/LENGTH                                            ELE04950
          S(5,5)=  1.0/LENGTH                                            ELE04960
          S(6,6)=  0.5/LENGTH                                            ELE04970
          S(1,5)= -1.0/LENGTH                                            ELE04980
          S(2,6)= -0.5/LENGTH                                            ELE04990
          S(3,4)= -0.5/LENGTH                                            ELE05000
```

```
      S(5,1)= -1.0/LENGTH                                               ELE05010
      S(6,2)= -0.5/LENGTH                                               ELE05020
      S(4,3)= -0.5/LENGTH                                               ELE05030
C                                                                       ELE05040
C----- CALCULATE SAT                                                    ELE05050
      DO 160 I=1,6                                                      ELE05060
        DO 160 J=1,KGDOF                                                ELE05070
        SAT2(I,J)=0                                                     ELE05080
        DO 160 K=1,6                                                    ELE05090
  160     SAT2(I,J)=SAT2(I,J)+S(I,K)*AT(J,K)                            ELE05100
C                                                                       ELE05110
C----- CALCULATE ASAT AND CONVERT TO HALF STORAGE MODE  BY COLUMNS      ELE05120
      DO 170 J=1,KGDOF                                                  ELE05130
        DO 170 I=J,1,-1                                                 ELE05140
        L=L+1                                                           ELE05150
        STIFF(L)=0                                                      ELE05160
        DO 170 K=1,6                                                    ELE05170
  170     STIFF(L)=STIFF(L)+AT(I,K)*SAT2(K,J)                           ELE05180
C                                                                       ELE05190
C----- PRINT GEOMETRIC STIFF DATA FOR DEB_UGGING                        ELE05200
      IF ( BTEST(IBUG,7) ) THEN                                         ELE05210
        CALL WMTRX2( AT    ,KGDOF,6,' AT    ',24)                      ELE05220
        CALL WMATRX( S     , 6,6,' KG-S ')                            ELE05230
        CALL WMATRX(SAT2  ,6,KGDOF,'SAT2  ')                           ELE05240
        WRITE (6,*) 'KGDOF',KGDOF,'LMKG =',(LMKG(I),I=1,KGDOF)          ELE05250
        CALL LMATRX(STIFF(LKG),MD,KGDOF,300,'WALL GEOMETRIC STIFF.')    ELE05260
      ENDIF                                                             ELE05270
C                                                                       ELE05280
C----- RELEASE UNUSED STORAGE                                           ELE05290
  180 IREL=600-L                                                        ELE05300
C                                                                       ELE05310
C     PRINT COLUMN DATA                                                 ELE05320
      IF (FIRST .OR. BTEST(IBUG,7) .OR. BTEST(IBUG,8) )                 ELE05330
     &  WRITE (6,192) SE(ISES+47)                                       ELE05340
  191 WRITE(6,193) NAME,IELNO,MATS,MATA1,MATA2,                         ELE05350
     &          ID(J1),ID(J2),ID(J3),ID(J4),                           ELE05360
     &     LENGTH,W,ALPIN,PKG                                          ELE05370
C                                                                       ELE05380
  192 FORMAT(/' ELEMENT 14, R/C SHEAR WALL ELEMENT( ',                  ELE05390
     &     'SHEAR SPAN LENGTH RATIO= ',G12.4, ')'// T17,               ELE05400
     &     '       ELEM',' SHEAR',' AXIAL1',' AXIAL2',' JOINT',' JOINT',ELE05410
     &     ' JOINT',' JOINT',    LENGTH   ',    WIDTH    ',           ELE05420
     &     '    ALPHA    ',     PKG     '/,T17,                        ELE05430
     &     '       #',' MATL',' MATL',' MATL',     #1',     #2',       ELE05440
     &     '    #3',     #4'/)                                         ELE05450
C                                                                       ELE05460
  193 FORMAT(1X,A18,8I6,2X,1P,4G12.4)                                   ELE05470
      IOPT=IOPT                                                         ELE05480
      RETURN                                                            ELE05490
C*************************** DETERMINE GEOMETRIC STIFF ********************ELE05500
  200 CONTINUE                                                          ELE05510
C- KG IS DETERMINED FOR A UNIT LOAD WITH IOPT=1,                        ELE05520
C- THE ACTUAL LOAD (PGEOM) IS DETERMINED HERE.                          ELE05530
C- KG IS MULTIPLIED BY PGEOM WHEN THE TOTAL STIFFNESS IS ASSEMBLED      ELE05540
C- NEGATIVE PGEOM IS COMPRESSION....                                    ELE05550
C                                                                       ELE05560
      AXIAL=.TRUE.                                                      ELE05570
C---------- DETERMINE THE AXIAL LOAD TO BE USED FOR KG...               ELE05580
      IF (KGDATA(1).EQ.1) THEN                                          ELE05590
            PGEOM =  PKG                                                ELE05600
      ELSE IF (KGDATA(1).EQ.2) THEN                                     ELE05610
            PGEOM = -F(2)                                               ELE05620
      ENDIF                                                             ELE05630
      IF (BTEST(IBUG,7)) WRITE (6,*) 'AXIAL ',AXIAL,'  PGEOM=',PGEOM    ELE05640
C                                                                       ELE05650
      RETURN                                                            ELE05660
C                                                                       ELE05670
C*************************** DETERMINE STIFFNESS **************************ELE05680
  300 CONTINUE                                                          ELE05690
      SS=SS                                                             ELE05700
      SA1=SA1                                                           ELE05710
      SA2=SA2                                                           ELE05720
      IF (BTEST(IBUG,7)) WRITE (6,402) IELNO,SS,SA1,SA2                 ELE05730
      IF (BTEST(IBUG,7)) WRITE (6,402) IELNO,SE(ISES),SE(ISEA1)         ELE05740
     &  ,SE(ISEA2)                                                      ELE05750
C                                                                       ELE05760
C------ SS,SA1,SA2 CONTAIN THE STIFFNESSES, WHICH WERE DETERMINED IN    ELE05770
C        THE LAST CALL TO MA_FLIB                                       ELE05780
C        IF SS=SE(ISES)  AND SA1=SE(ISEA1) AND SA2=SE(ISEA2)            ELE05790
C        THEN THE STIFFNESS HAS NOT CHANGED, RETURN                     ELE05800
C                                                                       ELE05810
      IF (SS.EQ.SE(ISES).AND.SA1.EQ.SE(ISEA1)                           ELE05820
     &  .AND.SA2.EQ.SE(ISEA2)) RETURN                                   ELE05830
C                                                                       ELE05840
C------- SET UP THE LOCAL STIFFNESS MATRIX                              ELE05850
      SS= SE(ISES)                                                      ELE05860
      SA1= SE(ISEA1)                                                    ELE05870
      SA2= SE(ISEA2)                                                    ELE05880
      IF (BTEST(IBUG,7)) WRITE (6,402) IELNO,SS,SA1,SA2                 ELE05890
      SAL1= SE(ISEA1)/(LENGTH*2)                                        ELE05900
      SAL2= SE(ISEA2)/(LENGTH*2)                                        ELE05910
      SSL= SE(ISES)                                                     ELE05920
C                                                                       ELE05930
C----- CALCULATE SAT                                                    ELE05940
      DO 360 J=1,IELDOF                                                 ELE05950
        SAT(1,J)=SSL*A(J,1)                                             ELE05960
        SAT(2,J)=SAL1*A(J,2)                                            ELE05970
  360   SAT(3,J)=SAL2*A(J,3)                                            ELE05980
C                                                                       ELE05990
C----- CALCULATE ASAT, STORE IN UPPER TRIANGULAR MATRIX                 ELE06000
      L=0                                                               ELE06010
      DO 370 J=1,IELDOF                                                 ELE06020
        DO 370 I=J,1,-1                                                 ELE06030
        L=L+1                                                           ELE06040
        DSTIFF(L)=0                                                     ELE06050
        DO 370 K=1,3                                                    ELE06060
  370     DSTIFF(L)=DSTIFF(L)+A(I,K)*SAT(K,J)                           ELE06070
      DO 371 J=1,L                                                      ELE06080
  371     STIFF(J)=DSTIFF(J)                                            ELE06090
C                                                                       ELE06100
      IF ( BTEST(IBUG,7) ) THEN                                         ELE06110
        WRITE (6,*) 'IELDOF',IELDOF,'LM =',(LM(I),I=1,IELDOF)           ELE06120
        CALL LMATRX(STIFF,MD,IELDOF,(L+1),'WALL STIFFNESS')             ELE06130
      ENDIF                                                             ELE06140
C                                                                       ELE06150
      RETURN                                                            ELE06160
C*************************** DETERMINE INCRIMENTAL FORCES *****************ELE06170
  400 CONTINUE                                                          ELE06180
       IF (PRINT.AND.FIRST) WRITE (6,491) STEPID                        ELE06190
       IF (BTEST(IBUG,7)) THEN                                          ELE06200
         WRITE (6,401) IELNO                                            ELE06210
  401    FORMAT (1X,30('='),'SHEAR WALL, ELEMENT#',I6,1X,30('=') )       ELE06220
       ENDIF                                                            ELE06230
C                                                                       ELE06240
C-------- SET UP THE COMPONENT STIFFNESSES                              ELE06250
      SS= SE(ISES)                                                      ELE06260
      SA1= SE(ISEA1)                                                    ELE06270
      SA2= SE(ISEA2)                                                    ELE06280
      IF (BTEST(IBUG,7)) WRITE (6,402) IELNO,SS,SA1,SA2                 ELE06290
  402 FORMAT(' IELNO=',I1,' SS=',G12.5,' SA1=',G12.5,' SA2=',G12.5)     ELE06300
C                                                                       ELE06310
C------- LOOP FOR EACH LOAD                                             ELE06320
      DO 430 LOAD=1,NLOAD                                               ELE06330
C                                                                       ELE06340
C------- TRANSFORM FRAME DISPL_INTO LOCAL DISPL                         ELE06350
      DO 410 I=1,3                                                      ELE06360
        R(I)=0                                                          ELE06370
        V(I)=0                                                          ELE06380
        DO 410 J=1,IELDOF                                               ELE06390
          K=LM(J)                                                       ELE06400
  410     R(I)=R(I)+A(J,I)*DISPL(K,LOAD)                                ELE06410
C                                                                       ELE06420
C------- INCRIMENTAL FORCES                                             ELE06430
      F(2)=SA1*R(2)/(LENGTH*2)                                          ELE06440
      F(3)=SA2*R(3)/(LENGTH*2)                                          ELE06450
      F(1)=SS*R(1)                                                      ELE06460
C                                                                       ELE06470
C------- CHECK STIFFNESS AND CALCULATE UNBALANCED FORCES                ELE06480
C        THIS IS PREFORMED ONLY FOR INELASTIC ANALYSIS                  ELE06490
      P(1) =FT(1)+F(1)                                                  ELE06500
      P(2) =FT(2)+F(2)                                                  ELE06510
      P(3) =FT(3)+F(3)                                                  ELE06520
      IF (.NOT. ELSTIC) THEN                                            ELE06530
C           -- TRANSFER PERMANENT HYSTERESIS DATA TO TEMPROARY STORAGE...ELE06540
            ISE2=ISE/2                                                  ELE06550
            DO 415 I=1,ISE2                                             ELE06560
  415         SE(I+ISE2)=SE(I)                                          ELE06570
C           -- SHEAR -----------------                                  ELE06580
C           -- SET UP TEMPORARY - TOTAL LOADS, DISPL'S AND VELOC        ELE06590
            PL=FT(1)                                                    ELE06600
            DL=RT(1)                                                    ELE06610
            D =(RT(1)+R(1))                                             ELE06620
            VC=VT(1)+V(1)                                               ELE06630
            VL=VT(1)                                                    ELE06640
C           --  SET INITIAL KKL BASED ON INITIAL DISP. INCREMENT(DSINC) ELE06650
C           --  IF INITIAL DISPL. INCREMENT IS POSITIVE, POSTFG=.TRUE.  ELE06660
            IF(SE(ISE2+44).EQ.0) THEN                                   ELE06670
               IF(SE(ISE2+4).GT.0) THEN                                 ELE06680
                  SE(ISE2+3)=1                                          ELE06690
                  SE(ISE2+44)=1                                         ELE06700
               END IF                                                   ELE06710
               IF(SE(ISE2+4).LT.0) THEN                                 ELE06720
                  SE(ISE2+3)=3                                          ELE06730
                  SE(ISE2+44)=1                                         ELE06740
                  SE(ISE2+37)=0                                         ELE06750
               END IF                                                   ELE06760
            END IF                                                      ELE06770
C           -- CALL HYST MODEL TO GET NEW STIFFNESS AND                 ELE06780
C              THE TOTAL FORCE P AT DISPL D.                            ELE06790
            MOPT=3                                                      ELE06800
            CALL MATLIB                                                 ELE06810
     &     (MOPT,LSTTYP,FALSE ,IREL,FALSE,NAME,                         ELE06820
     &     SSES,PSES,DUCT(1,1),EXCR(1,1),                               ELE06830
     &       P(1),PL,D,DL,VC,VL,LELEM ,LMAT  ,                          ELE06840
     &       MATS,MATTYP,SE(ISE2+ISES),IELNO,DAMAGE)                    ELE06850
            IF (SE(ISE2+ISES).NE.SE(ISES)) NEWK=.TRUE.                  ELE06860
C           -- AXIAL -----------------                                  ELE06870
C           -- SET UP TEMPORARY - TOTAL LOADS, DISPL'S AND VELOC        ELE06880
            PL=FT(2)*2                                                  ELE06890
            P(2)=P(2)*2                                                 ELE06900
            D =(RT(2)+R(2))/LENGTH                                      ELE06910
            DL=RT(2) /LENGTH                                            ELE06920
            VC=VT(2)+V(2)                                               ELE06930
            VL=VT(2)                                                    ELE06940
C           -- CALL HYST MODEL TO GET NEW STIFFNESS AND                 ELE06950
C              THE TOTAL FORCE P AT DISPL D.                            ELE06960
            MOPT=3                                                      ELE06970
            CALL MATLIB                                                 ELE06980
     &     (MOPT,LSTTYP,FALSE ,IREL,FALSE,NAME,                         ELE06990
     &     ESEA1,PSEA1,DUCT(1,2),EXCR(1,2),                             ELE07000
     &       P(2),PL,D,DL,VC,VL,LELEM ,LMAT  ,                          ELE07010
     &       MATA1 ,MATTYP,SE(ISE2+ISEA1),IELNO,DAMAGE)                 ELE07020
            P(2)=P(2)/2                                                 ELE07030
            IF (SE(ISE2+ISEA1).NE.SE(ISEA1)) NEWK=.TRUE.                ELE07040
C           -- AXIAL -----------------                                  ELE07050
C           -- SET UP TEMPORARY - TOTAL LOADS, DISPL'S AND VELOC        ELE07060
            PL=FT(3)*2                                                  ELE07070
            P(3)=P(3)*2                                                 ELE07080
            D =(RT(3)+R(3))/LENGTH                                      ELE07090
            DL=RT(3) /LENGTH                                            ELE07100
            VC=VT(3)+V(3)                                               ELE07110
            VL=VT(3)                                                    ELE07120
C           -- CALL HYST MODEL TO GET NEW STIFFNESS AND                 ELE07130
C              THE TOTAL FORCE P AT DISPL D.                            ELE07140
            MOPT=3                                                      ELE07150
            CALL MATLIB                                                 ELE07160
     &     (MOPT,LSTTYP,FALSE ,IREL,FALSE,NAME,                         ELE07170
     &     ESEA2,PSEA2,DUCT(1,3),EXCR(1,3),                             ELE07180
     &       P(3),PL,D,DL,VC,VL,LELEM ,LMAT  ,                          ELE07190
     &       MATA2 ,MATTYP,SE(ISE2+ISEA2),IELNO,DAMAGE)                 ELE07200
            P(3)=P(3)/2                                                 ELE07210
            IF (SE(ISE2+ISEA2).NE.SE(ISEA2)) NEWK=.TRUE.                ELE07220
         ENDIF                                                          ELE07230
C                                                                       ELE07240
         IF (ELSTIC) THEN                                               ELE07250
            ESES=0.50*(FT(1)+F(1))*(RT(1)+R(1))                         ELE07260
            ESEA1=0.50*(FT(2)+F(2))*(RT(2)+R(2))                        ELE07270
            ESEA2=0.50*(FT(3)+F(3))*(RT(3)+R(3))                        ELE07280
            ESE= ESES+ESEA1+ESEA2                                       ELE07290
            PSE=0.0                                                     ELE07300
         ELSE                                                           ELE07310
            PSE=PSES+PSEA1+PSEA2                                        ELE07320
            ESE=ESES+ESEA1+ESEA2                                        ELE07330
         ENDIF                                                          ELE07340
C                                                                       ELE07350
C------- SAVE MAXIMUN LOADS                                             ELE07360
      IF (NLOAD.GT.1) THEN                                              ELE07370
        DO 389 I=1,3                                                    ELE07380
          IF (ABS(F(I)).GT.ABS(FMAX(I)) ) THEN                          ELE07390
            FMAX(I  )= F(I)                                             ELE07400
            FMAX(I+3)= R(I)                                             ELE07410
          ENDIF                                                         ELE07420
  389   CONTINUE                                                        ELE07430
      ENDIF                                                             ELE07440
C                                                                       ELE07450
C------- PRINT DATA                                                     ELE07460
      IF (PRINT) WRITE (6,392) IELNO,LOAD,                              ELE07470
     &          ID(J1),ID(J2),ID(J3),ID(J4),                            ELE07480
     &                                (F(I),R(I),I=1,3)                 ELE07490
  392 FORMAT (I10,I5,2X,4I5,1P,6G15.6)                                  ELE07500
C                                                                       ELE07510
C------- CALCULATE THE UNBALANCED MEMBER FORCE ON A JOINT.              ELE07520
      IF (LOAD.EQ.1) THEN                                               ELE07530
            UBS = -( F(1)+FT(1)-P(1) )                                  ELE07540
            UBA1 = -( F(2)+FT(2)-P(2) )                                 ELE07550
            UBA2 = -( F(3)+FT(3)-P(3) )                                 ELE07560
  363   FORMAT(4G13.4)                                                  ELE07570
            DO 440 I=1,IELDOF                                           ELE07580
              K=LM(I)                                                   ELE07590
              EFUB(I)= A(I,1)*UBS + A(I,2)*UBA1 + A(I,3)*UBA2           ELE07600
  440         FUB(K)= FUB(K) + EFUB(I)                                  ELE07610
            IF (BTEST(IBUG,7)) THEN                                     ELE07620
              WRITE (6,493) (I,F(I),FT(I),P(I),I=1,3)                   ELE07630
              WRITE (6,494) (I,LM(I),EFUB(I),I=1,IELDOF)                ELE07640
            ENDIF                                                       ELE07650
      ENDIF                                                             ELE07660
      F(1)=P(1)-FT(1)                                                   ELE07670
      F(2)=P(2)-FT(2)                                                   ELE07680
      F(3)=P(3)-FT(3)                                                   ELE07690
  430 CONTINUE                                                          ELE07700
C                                                                       ELE07710
C------- THIS IS FOR SOL05A ONLY                                        ELE07720
      IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                         ELE07730
        DO 431 I=1,3                                                    ELE07740
  431     RSAF(I)=F(I)                                                  ELE07750
      ENDIF                                                             ELE07760
C                                                                       ELE07770
  491 FORMAT (//'    R/C SHEAR WALL FORCES...',5X,A/                    ELE07780
     &  '   ELEMENT LOAD',T17,                                          ELE07790
     &  ' JNT-1',' JNT-2',' JNT-3',' JNT-4',                            ELE07800
     &  '      SHEAR     ',' SHEAR DISPL.  ',                          ELE07810
     &  '      AXIAL1    ',' AXIAL DISPL. 1',                          ELE07820
     &  '      AXIAL2    ',' AXIAL DISPL. 2'/)                         ELE07830
  492 FORMAT (I10,I5,2X,4I5,1P,6G15.6)                                  ELE07840
```

```
  493 FORMAT ('  I=',I1,'  F=',1P,G12.4,'  FT=',G12.4,'  P=',G12.4)
  494 FORMAT ('  I=',I5,'  LM(I)=',I6,'  FUB(I)=',1P,G12.4)
C
      IOPT=IOPT
      RETURN
C------------------------ DETERMINE TOTAL FORCES, ENERGIES ECT ---------
  500 CONTINUE
      IF( ICOMN .NE. 1 .AND. ICOMN .NE. 2 ) THEN
       IF (PRINT.AND.FIRST) WRITE (6,491) STEPID
C
C------ TOTAL FORCES, DISPLACEMENTS AND VELOCITIES
      DO 510 I=1,3
      FT(I) = FT(I) + F(I)
      RT(I) = RT(I) + R(I)
      VT(I) = VT(I) + V(I)
      IF ( ABS(FT(I)).GT.ABS(FMAX(I)) ) THEN
        FMAX(I  )=FT(I)
        FMAX(I+3)=RT(I)
      ENDIF
  510 CONTINUE
C
C------ PRINT DATA
      IF (PRINT) WRITE (6,492) IELNO,LOAD,
     &              ID(J1),ID(J2),ID(J3),ID(J4),
     &                                   (FT(I),RT(I),I=1,3)
      ELSE IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN
        IF( PRINT .AND. FIRST) WRITE(6,497) STEPID
          DO 513 I=1,3
  513       FT(I)=RSAF(I)
          IF (PRINT) WRITE (6,492) IELNO,LOAD,
     &              ID(J1),ID(J2),ID(J3),ID(J4),
     &                                   (FT(I),I=1,3)
      ENDIF
  497 FORMAT (/'    R/C SHEAR WALL FORCES...',5X,A/
     & ' ELEMENT LOAD',T17,
     & ' JNT-1',' JNT-2',' JNT-3',' JNT-4',
     & '    SHEAR      ','    AXIAL1    ','    AXIAL2    ')
C
C------ TRANSFER TEMPROARY HYSTERESIS DATA TO PERMANENT STORAGE...
      ISE2=ISE/2
      IF (.NOT.ELSTIC) THEN
        DO 520 I=1,ISE2
  520     SE(I)=SE(I+ISE2)
      ENDIF
C
      RETURN
C
C------------------------ DETERMINE FIXED END FORCES, AND ADD TO FORCE-
  600 RETURN
C
C------- FIXED END FORCES ARE NOT AVAILABLE FOR R/C SHEAR WALL
C
C---------------------- WRITE MEMBER FORCES TO OUTPUT FILE ---------------
  700 CONTINUE
C
C.......... SHEAR ENERGY, DUCTILITY AND EXCURSION .....
      IF (ELSTIC) THEN
         ESES=0.50*(FT(1)+F(1))*(RT(1)+R(1))
         ESEA1=0.50*(FT(2)+F(2))*(RT(2)+R(2))
         ESEA2=0.50*(FT(3)+F(3))*(RT(3)+R(3))
C CHANGE FOR DAMAGE
         ESE= ESES/LENGTH+ESEA1+ESEA2
         PSE=0.0
      ELSE
        CALL MATLIB
     & ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &   ESES,PSES,DUCT(1,1),EXCR(1,1),
     &   FT    ,F      ,RT    ,R      ,0.     ,0.     ,LELEM ,LMAT  ,
     &   MATS  ,MATTYP,SE(ISES),IELNO,DAMAGE)
C.......... AXIAL ENERGY, DUCTILITY AND EXCURSION (1)..
        CALL MATLIB
     & ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &   ESEA1,PSEA1,DUCT(1,2),EXCR(1,2),
     &   FT    ,F      ,RT    ,R      ,0.     ,0.     ,LELEM ,LMAT  ,
     &   MATA1 ,MATTYP,SE(ISEA1),IELNO,DAMAGE)
C.......... AXIAL ENERGY, DUCTILITY AND EXCURSION (2)..
        CALL MATLIB
     & ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &   ESEA2,PSEA2,DUCT(1,3),EXCR(1,3),
     &   FT    ,F      ,RT    ,R      ,0.     ,0.     ,LELEM ,LMAT  ,
     &   MATA2 ,MATTYP,SE(ISEA2),IELNO,DAMAGE)
C
         ESE=ESES+ESEA1+ESEA2
         PSE=PSES+PSEA1+PSEA2
      ENDIF
C

      IF (PRINT) WRITE (IUNIT,710) ID(J1),ID(J2),ID(J3),ID(J4)
      WRITE (IUNIT,711) (FT(I),RT(I),I=1,3),ESE,PSE
C     &       ((DUCT(I,J),EXCR(I,J),I=1,3),J=1,3)
  710 FORMAT (4I5,1P,6E10.3/(8E10.3) )
  711 FORMAT (1P,8E10.3)
C
      RETURN
C
C--------------------- READ AND PRINT MEMBER FORCES FROM OUTPUT FILE ----
  800 CONTINUE
      BACKSPACE(IUNIT)
      READ (IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT
      READ (IUNIT,710) NODE1,NODE2,NODE3,NODE4
      ISTEP=-IWRITE
C
  810 READ (IUNIT,815,END=830) (F(I),R(I),I=1,3),ESE,PSE
  815 FORMAT (8E10.3)
         ISTEP=ISTEP+IWRITE
         T=TO+DT*ISTEP
         IF (ISTEP.EQ.0) WRITE (6,805) NODE1,NODE2,NODE3,NODE4
         IF (MOD(ISTEP,INC).EQ.0)
     &   WRITE (6,820) ISTEP,T,(F(I),R(I),I=1,3),ESE,PSE
         GO TO 810
C
  830 CONTINUE

C     WRITE (6,835)
C
C     WRITE (6,840) 'BENDING COMPONENT',(DUCT(I,1),EXCR(I,1),I=1,3)
C     WRITE (6,840) 'SHEAR COMPONENT'  ,(DUCT(I,2),EXCR(I,2),I=1,3)
C     WRITE (6,840) 'AXIAL COMPONENT'  ,(DUCT(I,3),EXCR(I,3),I=1,3)
C
  805 FORMAT (/'    R/C SHEAR WALL FORCES...',/
     & 5X,'NODE1=',I6,5X,'NODE2=',I6,5X,'NODE3=',I6,5X,'NODE4=',I6,//
     &    '    STEP     TIME    ',
     &  '      SHEAR     ',' SHEAR DISPL. ',
     &  '      AXIAL1    ',' AXIAL DISPL. 1',
     &  '      AXIAL2    ',' AXIAL DISPL. 2',
     &  '   ESE          ','          PSE          ')
  820 FORMAT (I6,1P,G12.4,6G14.5,2G12.4)
  835 FORMAT (/T10,' MAXIMUM DUCTILITIES AND EXCURSION RATIOS'/
     &        T10,'=====================================')
  840 FORMAT (/T10,A/
     & T15,'DISPLACEMENT DEFINITION: U1=',1P,G12.4,5X,'E1=',G12.4/
     & T15,'ENERGY DEFINATION #1:    U2=',1P,G12.4,5X,'E2=',G12.4/
     & T15,'ENERGY DEFINATION #2:    U3=',1P,G12.4,5X,'E3=',G12.4)
      RETURN
C
C---------------------- DETERMINE THE LENGTH OF THE MATERIAL DATA
  900 CONTINUE
      ISE=0
C------- LOOP FOR EACH MATEIAL -------
      DO 910 I=1,3
        IF (I.EQ.1) THEN
          ISES=ISE+1
        ELSE IF (I.EQ.2) THEN
          ISEA1=ISE+1
        ELSE IF (I.EQ.3) THEN
          ISEA2=ISE+1
        ENDIF
        MATI = RINPUT(I)
        CALL MATLIB
     & ( 0    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &   ESE, PSE,DUCT(1,2),EXCR(1,2),
     &   FT    ,F      ,RT    ,R      ,0.     ,0.     ,LELEM ,LMAT  ,
     &   MATI  ,MATTYP,SE,IELNO,DAMAGE)
  910   ISE=ISE + LELEM
C
C------- DOUBLE THE STORAGE, TO ALLOW FOR TEMPROARY VALUES...
        ISE=ISE*2
C
      RETURN
C
C--------------------- DETERMINE THE STRAIN ENERGY
 1000 CONTINUE
C--------------------- DUCTILITEIS AND EXCURSION RATIOS
 1100 CONTINUE
C     ... NOTE ELE03 WAS LAST CALLED TO CALC INCIRMENTAL DISPL,
C              THE ENERGIES ASSOCIATED WITH INCR DISPL ARE STORED IN
C              THE ADDRESSES: ISES+ISE/2
C                             ISEA1+ISE/2
C                             ISEA2+ISE/2
      ISE2=ISE/2
C.......... SHEAR ENERGY, DUCTILITY AND EXCURSION .....
      CALL MATLIB
     & ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &   ESES,PSES,DUCT(1,1),EXCR(1,1),
     &   FT    ,F      ,RT    ,R      ,0.     ,0.     ,LELEM ,LMAT  ,
     &   MATS  ,MATTYP,SE(ISES+ISE2),IELNO,DAMAGE)
C.......... AXIAL ENERGY, DUCTILITY AND EXCURSION (1)..
      CALL MATLIB
     & ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &   ESEA1,PSEA1,DUCT(1,2),EXCR(1,2),
     &   FT    ,F      ,RT    ,R      ,0.     ,0.     ,LELEM ,LMAT  ,
     &   MATA1 ,MATTYP,SE(ISEA1+ISE2),IELNO,DAMAGE)
C.......... AXIAL ENERGY, DUCTILITY AND EXCURSION (2)..
      CALL MATLIB
     & ( 4    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &   ESEA2,PSEA2,DUCT(1,3),EXCR(1,3),
     &   FT    ,F      ,RT    ,R      ,0.     ,0.     ,LELEM ,LMAT  ,
     &   MATA2 ,MATTYP,SE(ISEA2+ISE2),IELNO,DAMAGE)
C
         IF( DUCFAG .EQ. 0 ) THEN
         IF( ABS(DUCT(1,1)) .GE. 1 .OR. ABS(DUCT(1,2)) .GE. 1
     &   .OR. ABS(DUCT(1,3)) .GE. 1 ) THEN
              DUCFAG=ABS(DUCT(1,1))
              IF( DUCFAG .LT. ABS(DUCT(1,2)) ) DUCFAG=ABS(DUCT(1,2))
              IF( DUCFAG .LT. ABS(DUCT(1,3)) ) DUCFAG=ABS(DUCT(1,3))
         ENDIF
         ENDIF
C
      IF (ELSTIC) THEN
         ESES=0.50*(FT(1)+F(1))*(RT(1)+R(1))
         ESEA1=0.50*(FT(2)+F(2))*(RT(2)+R(2))
         ESEA2=0.50*(FT(3)+F(3))*(RT(3)+R(3))
         ESE= ESES+ESEA1+ESEA2
         PSE=0.0
      ELSE
         ESE=ESES+ESEA1+ESEA2
         PSE=PSES+PSEA1+PSEA2
      ENDIF
C
      IF (BTEST(IBUG,7)) THEN
          WRITE (6,*) 'ESES ESEA1 ESEA2 ',ESES,ESEA1,ESEA2
          WRITE (6,*) 'PSES PSEA1 PSEA2  ',PSES,PSEA1,PSEA2
          WRITE (6,*) 'ESE  PSE  LENGTH',ESE ,PSE ,LENGTH
      ENDIF
       IF (PRINT.AND.FIRST) WRITE (6,1191)
       IF (PRINT) WRITE (6,1192)IELNO,((DUCT(I,J),EXCR(I,J),I=1,3),J=1,3)
 1192 FORMAT (1X,I5,18(7,2)
 1191 FORMAT /' R/C SHEAR WALL DUCTILITY AND EXCURSION RATIOS'/6X,
     & ' |---------- SHEAR COMPONENT ----------|',
     & ' |---------- AXIAL COMPONENT 1----------|',
     & ' |---------- AXIAL COMPONENT 2----------|'/,' ELEM#',
C        12345678901234567890123456789012345678901234567890123456789012
     & ' |-- DEFN 1--- --- DEFN 2--- --- DEFN 3--|',
     & ' |-- DEFN 1--- --- DEFN 2--- --- DEFN 3--|',
     & ' |-- DEFN 1--- --- DEFN 2--- --- DEFN 3--|'/6X,
     & '  DUCT.  EXCR.  DUCT.  EXCR.  DUCT.  EXCR.',
     & '  DUCT.  EXCR.  DUCT.  EXCR.  DUCT.  EXCR.',
     & '  DUCT.  EXCR.  DUCT.  EXCR.  DUCT.  EXCR.')
C        12345678901234567890123456789012345678901234567890123456789012
      RETURN
C
C------------------------ WRITE MAXIMUM AND MINIMUM VALUES
 1200 CONTINUE
      IF (FIRST) WRITE(6,491)'MAXIMUM LOADS AND DISPL AT MAXIMUM LOADS '
     &           //' NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'
     &           WRITE (6,492) IELNO,LOAD,
     &           ID(J1),ID(J2),ID(J3),ID(J4),
     &                (FMAX(I),FMAX(3+I),I=1,3)
C
      RETURN
C
C------------------------ RESET INTERNAL FORCES
 1300 CONTINUE
C
C----- ZERO MATRICIES
      W = SE(ISES + 78)
      CO1=SE(ISES + 37)
C     CO2=SE(ISES + 44)
C     CO3=SE(ISES + 45)
C     CO4=SE(ISES + 46)
C
      DO 1310 I=1,ISE
 1310   SE(I)=0
C
      SE(ISES+77)=LENGTH
      SE(ISES+78)=W
      SE(ISES + 37)=CO1
C
      DO 1325 I=1,2
         FMAX(I)=0
         FMAX(I+2)=0
         F (I)=0
         FT(I)=0
         R (I)=0
         RT(I)=0
         V (I)=0
 1325    VT(I)=0
      WRITE (6,1330) IELNO
 1330 FORMAT (' SHEAR WALL..... ELEMENT #',I5,
     &   '  INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO')

      CALL MATLIB
     & ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &   ESES,PSES,DUCT(1,1),EXCR(1,1),
     &   FT    ,F      ,RT    ,R      ,0.     ,0.     ,LELEM ,LMAT  ,
     &   MATS  ,MATTYP,SE(ISES),IELNO,DAMAGE)
      CALL MATLIB
     & ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &   ESEA1,PSEA1,DUCT(1,2),EXCR(1,2),
     &   FT    ,F      ,RT    ,R      ,0.     ,0.     ,LELEM ,LMAT  ,
     &   MATA1 ,MATTYP,SE(ISEA1),IELNO,DAMAGE)
      CALL MATLIB
     & ( 2    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,
     &   ESEA2,PSEA2,DUCT(1,3),EXCR(1,3),
```

```
     &   FT    ,F     ,RT    ,R     ,0.    ,0.    ,LELEM ,LMAT  ,      ELE10690
     &   MATA2  ,MATTYP,SE(ISEA2),IELNO,DAMAGE)                         ELE10700
C                                                                       ELE10710
      RETURN                                                            ELE10720
C--------------------- DAMAGE INDEX                                     ELE10730
 1400 IF (ELSTIC) RETURN                                                ELE10740
      FMAX1=FMAX(1)                                                     ELE10750
      FMAX2=FMAX(2)                                                     ELE10760
      FMAX3=FMAX(3)                                                     ELE10770
      FMAX4=FMAX(4)                                                     ELE10780
      FMAX5=FMAX(5)                                                     ELE10790
      FMAX6=FMAX(6)                                                     ELE10800
C.......... SHEAR ENERGY, AND DAMAGE INDEX                             ELE10810
      CALL MATLIB                                                       ELE10820
     & ( 5    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                          ELE10830
     &   ESES,PSES,DUCT(1,1),EXCR(1,1),                                 ELE10840
     &   FMAX1,0.    ,FMAX4,0.    ,0.    ,0.    ,LELEM ,LMAT  ,       ELE10850
     &   MATS  ,MATTYP,SE(ISES),IELNO,DAMS)                             ELE10860
C.......... AXIAL ENERGY, AND DAMAGE INDEX (1)                         ELE10870
      CALL MATLIB                                                       ELE10880
     & ( 5    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                          ELE10890
     &   ESEA1,PSEA1,DUCT(1,2),EXCR(1,2),                               ELE10900
     &   FMAX2,0.    ,FMAX5,0.    ,0.    ,0.    ,LELEM ,LMAT  ,       ELE10910
     &   MATA1  ,MATTYP,SE(ISEA1),IELNO,DAMA1)                          ELE10920
C.......... AXIAL ENERGY, AND DAMAGE INDEX (2)                         ELE10930
      CALL MATLIB                                                       ELE10940
     & ( 5    ,LSTTYP ,FALSE ,IREL,FALSE,NAME,                          ELE10950
     &   ESEA2,PSEA2,DUCT(1,3),EXCR(1,3),                               ELE10960
     &   FMAX3,0.    ,FMAX6,0.    ,0.    ,0.    ,LELEM ,LMAT  ,       ELE10970
     &   MATA2  ,MATTYP,SE(ISEA2),IELNO,DAMA2)                          ELE10980
C                                                                       ELE10990
      ESE=ESES+ESEA1+ESEA2                                              ELE11000
      PSE=PSES+PSEA1+PSEA2                                              ELE11010
      DAMAGE=DAMS*(ESES+PSES)+DAMA1*(ESEA1+PSEA1)                       ELE11020
     &       +DAMA2*(ESEA2+PSEA2)                                       ELE11030
      IF (FIRST) WRITE (6,1420)                                         ELE11040
      WRITE (6,1410)IELNO,DAMS,ESES ,PSES,DAMA1,ESEA1,PSEA1,DAMA2,      ELE11050
     &ESEA2,PSEA2                                                       ELE11060
 1410 FORMAT (1X,I5,1P,3(F14.5,2G14.5))                                 ELE11070
 1420 FORMAT (/6X,                                                      ELE11080
     & ' |---------- SHEAR COMPONENT ------------|',                    ELE11090
     & ' |---------- AXIAL COMPONENT ------------|'/,' ELEM#',          ELE11100
C        12345678901234567890123456789012345678901234567890123456789012ELE11110
     & ' |-- DAMAGE -- ---- ESE ---- ---- PSE ---|',                    ELE11120
     & ' |-- DAMAGE -- ---- ESE ---- ---- PSE ---|')                    ELE11130
      RETURN                                                            ELE11140
      END                                                               ELE11150
@PROCESS SOUMP GOSTMT XREF                                              ENE00010
C=======================================================================ENE00020
C       DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT                              ENE00030
C       END DEBUG                                                       ENE00040
C=======================================================================ENE00050
C   ENERGY BALANCE                                                      ENE00060
C=======================================================================ENE00070
      SUBROUTINE ENERGY(L1,L2,L3,L4,L5,L6,NDOF,MAXNOD,NCOS,PT2,COND,    ENE00080
     &  IMD,IMDMS,IMDKG,MD,MDMS,MDKG,DYN,WRITE,ABORT,                   ENE00090
     &  STIFF,KG,MASS,                                                  ENE00100
     &  EIE,ESE,PSE,EKE,EDD,                                            ENE00110
     &  ALPHA,BETA,KGFORM,NELMT,NNODE,DT,SUMUB,                         ENE00120
     &   SUMRC,DSUMRC,  SUMFD,DSUMFD,FDAMP,DFDAMP,  FZ,                 ENE00130
     &  GA,GV,GD,GAT,GVT,GDT,   WORK,WORK2,FG,DFG,                      ENE00140
     &  DACC,DVEL,DDISP,ACC,VEL,DISP,  LOAD,DLOAD,                      ENE00150
     &  ID      , IDOF     , CNST     , JTFLG    , COORD    ,          ENE00160
     &  COSINE   , JTCOS    )                                          ENE00170
C                                                                       ENE00180
      DOUBLE PRECISION  EIE,ESE,DEIE,PSE,DEDD,EDD,EKE                   ENE00190
      COMMON /ISPEC/ ISP1,SGD(3000,2),REACF(6)                          ENE00200
      DIMENSION COORD(MAXNOD,3),ID(MAXNOD),                             ENE00210
     &          IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),CNST(MAXNOD,6),        ENE00220
     &          JTFLG(MAXNOD),JTCOS(MAXNOD)                             ENE00230
C                                                                       ENE00240
      REAL MASS,KG,LOAD                                                 ENE00250
C                                                                       ENE00260
      CHARACTER*80 NAME                                                 ENE00270
      DIMENSION STIFF(IMD),MD(L6+1)                                     ENE00280
      DIMENSION MASS(IMDMS),MDMS(L6+1)                                  ENE00290
      DIMENSION KG(IMDKG),MDKG(L6+1)                                    ENE00300
      DIMENSION RINPUT(100),FDAMP(L6),DFDAMP(L6),FG(6),DFG(6)           ENE00310
      DIMENSION  SUMRC(6),DSUMRC(6),SUMFD(6),DSUMFD(6),SUMUB(6)         ENE00320
      LOGICAL BTEST,PRINT,HEAD,PT2,AXIAL,COND,F,T,DYN,WRITE,ABORT       ENE00330
C                                                                       ENE00340
      DIMENSION  GD(3), GV(3), GA(3)                                    ENE00350
      DIMENSION GDT(3),GVT(3),GAT(3)                                    ENE00360
      DIMENSION DDISP(L6),DVEL(L6),DACC(L6)                             ENE00370
      DIMENSION  DISP(L6), VEL(L6), ACC(L6)                             ENE00380
      DIMENSION WORK(L6),WORK2(L6),LOAD(L6),DLOAD(L6)                   ENE00390
      DATA F/.FALSE./, T/.TRUE./                                        ENE00400
C                                                                       ENE00410
C------------------ NOMENCLATURE ---------------------------------------CENE00420
C  DACC  INCRIMENTAL RELATIVE ACCELERATION                              CENE00430
C  DVEL  INCRIMENTAL RELATIVE VELOCITY                                  CENE00440
C  DDISP INCRIMENTAL RELATIVE DISPLACEMENT                              CENE00450
C  ACC   PREVIOUS STEP'S RELATIVE ACCELERATION                          CENE00460
C  VEL   PREVIOUS STEP'S RELATIVE VELOCITY                              CENE00470
C  DISP  PREVIOUS STEP'S RELATIVE DISPLACEMENT                          CENE00480
C  DLOAD INCRIMENTAL APPLIED LOAD, NOT M*AG                             CENE00490
C  LOAD  PREVIOUS STEP'S APPLIED LOAD, NOT M*AG                         CENE00500
C  DFDAMP INCRIMENTAL DAMPING FORCE                                     CENE00510
C  FDAMP PREVIOUS STEP'S DAMPING FORCE                                  CENE00520
C                                                                       CENE00530
C  GD    INCRIMENTAL GROUND DISPLACEMENT                                CENE00540
C  GV    INCRIMENTAL GROUND VELOCITY                                    CENE00550
C  GA    INCRIMENTAL GROUND ACCELERATION                                CENE00560
C  GDT   TOTAL GROUND DISPLACEMENT                                      CENE00570
C  GVT   TOTAL GROUND VELOCITY                                          CENE00580
C  GAT   TOTAL GROUND ACCELERATION                                      CENE00590
C  WORK  TEMP FORCE VECTOR                                              CENE00600
C  WORK2 TEMP DISPL OR VELOCITY VECTOR                                  CENE00610
C-----------------------------------------------------------------------CENE00620
C                                                                       ENE00630
      L41=L4+1                                                          ENE00640
C                                                                       ENE00650
C=======================================                                ENE00660
C===  INTEGRATE GROUND ACCELERATION  ===                                ENE00670
C=======================================                                ENE00680
      IF (DYN) THEN                                                     ENE00690
      DO 10 I=1,3                                                       ENE00700
         GA(I)=GA(I)                                                    ENE00710
         GV(I)=(2*GAT(I)+GA(I)) *DT /2                                  ENE00720
         GD(I)=GVT(I)*DT + (3*GAT(I)+GA(I)) *DT*DT /6                   ENE00730
         GAT(I)=GA(I)+GAT(I)                                            ENE00740
         GVT(I)=GV(I)+GVT(I)                                            ENE00750
   10    GDT(I)=GD(I)+GDT(I)                                            ENE00760
      ENDIF                                                             ENE00770
       IF (PT2) THEN                                                    ENE00780
        IF (DYN) THEN                                                   ENE00790
             WRITE (6,22) (I,ACC(I),VEL(I),DISP(I),LOAD(I),I=1,L4)      ENE00800
             WRITE (6,23) (I,DACC(I),DVEL(I),DDISP(I),DLOAD(I),I=1,L4)  ENE00810
             WRITE (6,24) (I,FG(I),DFG(I),FDAMP(I),DFDAMP(I),I=1,L6)    ENE00820
        ELSE                                                            ENE00830
             WRITE (6,22) (I,ACC(I),VEL(I),DISP(I),LOAD(I),I=1,L6)      ENE00840
             WRITE (6,23) (I,DACC(I),DVEL(I),DDISP(I),DLOAD(I),I=1,L6)  ENE00850
             WRITE (6,24) (I,FG(I),DFG(I),FDAMP(I),DFDAMP(I),I=1,L6)    ENE00860
        ENDIF                                                           ENE00870
   22   FORMAT(/T10,'TOTAL RELATIVE RESPONSE FOR PREVIOUS TIME STEP'/    ENE00880
     &  (T10,'DOF:',I6,' ACC:',1P,G13.5,' VEL:',G13.5,' DSP:',G13.5,    ENE00890
     &       'LOAD:',G13.5))                                            ENE00900
   23   FORMAT(/T10,'INCRIMENTAL RELATIVE RESPONSE'/                     ENE00910
     &  (T10,'DOF:',I6,' ACC:',1P,G13.5,' VEL:',G13.5,' DSP:',G13.5,    ENE00920
     &       'LOAD:',G13.5))                                            ENE00930
   24   FORMAT(/T10,'SPECIAL FORCES '/                                   ENE00940
     &  (T10,'DOF:',I6,' FD:',1P,G13.5,' DFD:',G13.5,' FDAMP:',G13.5,   ENE00950
     &       ' DFDAMP:',G13.5))                                         ENE00960
        ENDIF                                                           ENE00970
C                                                                       ENE00980
C=======================================                                ENE00990
C===  CALCULATE INPUT EARTHQUAKE ENERGY  ===                            ENE01000
C=======================================                                ENE01010
C                                                                       ENE01020
      DEIE=0                                                            ENE01030
C                                                                       ENE01040
C..... CALCULATE THE INCR. INPUT ENERGY DUE TO GRND MOTION              ENE01050
      IF (DYN) THEN                                                     ENE01060
          IF (PT2) WRITE(6,*) ' SUMRC', SUMRC                           ENE01070
          IF (PT2) WRITE(6,*) 'DSUMRC',DSUMRC                           ENE01080
          IF (PT2) WRITE(6,*) ' SUMFD', SUMFD                           ENE01090
          IF (PT2) WRITE(6,*) 'DSUMFD',DSUMFD                           ENE01100
          IF (PT2) WRITE(6,*) ' SUMUB', SUMUB                           ENE01110
          DO 48 I=1,3                                                   ENE01120
C           FORCEI= SUMRC(I)+ 0.5*DSUMRC(I) + SUMFD(I) + 0.5*DSUMFD(I)  ENE01130
C    &            + SUMUB(I)                                           ENE01140
            FORCEI= SUMRC(I)+ 0.5*DSUMRC(I) + SUMFD(I) + 0.5*DSUMFD(I)  ENE01150
     &            + SUMUB(I)+ REACF(I)                                  ENE01160
            ENER=GD(I)*FORCEI                                           ENE01170
            IF (PT2) WRITE (6,*) I,                                     ENE01180
     &          'FORCEI',FORCEI,' GD',GD(I),'  ENER',ENER               ENE01190
   48       DEIE=DEIE + ENER                                            ENE01200
      ENDIF                                                             ENE01210
C                                                                       ENE01220
C..... CALCULATE THE INCR. INPUT ENERGY DUE TO APPLIED LOADS...         ENE01230
          N2=L6                                                         ENE01240
          IF (DYN) N2=L4                                                ENE01250
          DO 50 I=L3,N2                                                 ENE01260
            FORCEI=LOAD(I)+DLOAD(I)/2                                   ENE01270
            ENER=FORCEI*DDISP(I)                                        ENE01280
C           WRITE(70,*) I, ENER                                         ENE01290
            IF (PT2) WRITE (6,*) I,                                     ENE01300
     &        'FORCEI',FORCEI,'  DDISP',DDISP(I),'  ENER',ENER          ENE01310
   50       DEIE=DEIE + ENER                                            ENE01320
C                                                                       ENE01330
C..... CALCULATE THE ENERGY DUE TO GEOMETRIC STIFFNESS                  ENE01340
       IF (KGFORM.EQ.2 ) THEN                                           ENE01350
C....    GET THE INCRIMENTAL DISPL AND STORE IN VECTOR WORK2.....       ENE01360
C...     SAVE INCRIMENTAL IN VECTOR WORK2                               ENE01370
         DO 51 I=L3,L4                                                  ENE01380
   51      WORK(I)=DDISP(I)                                             ENE01390
C...     ADD GROUND RESPONSE IN JOINT CORRDINATES TO RELATIVE RESP      ENE01400
C          MULTIPLY GROUND DISPL * DIRECTION COSINE                     ENE01410
C          OMIT ALL DOF THAT ARE CONSTRAINED TO A MASTER DOF...         ENE01420
         IF (GD(1).NE.0 .OR. GD(2).NE.0 .OR. GD(3).NE.0) THEN           ENE01430
           DO 52 NODE=1,NNODE                                           ENE01440
             ICOS=JTCOS(NODE)                                           ENE01450
             DO 52 K=1,3                                                ENE01460
             IF (.NOT. BTEST(JTFLG(NODE),K+11)) THEN                    ENE01470
               II=IDOF(NODE,K)                                          ENE01480
               IF (II.LE.L4 .AND. II.GE.L3)                             ENE01490
     &         WORK(II)=WORK(II)        + GD(1)*COSINE(1,K,ICOS)        ENE01500
     &         + GD(2)*COSINE(2,K,ICOS) + GD(3)*COSINE(3,K,ICOS)        ENE01510
             ENDIF                                                      ENE01520
   52        CONTINUE                                                   ENE01530
       ENDIF                                                            ENE01540
C                                                                       ENE01550
C..... CALCULATE THE INCRIMENTAL INPUT ENERGY DUE TO KG...              ENE01560
       DIKG=0                                                           ENE01570
         DO 53 I=L3,L4                                                  ENE01580
            ENER= (FG(I)+DFG(I)/2)*WORK(I)                              ENE01590
            IF (PT2) WRITE (6,*) I,                                     ENE01600
     &          'FG,DFG',FG(I),DFG(I),'  DDISP',WORK(I),'  ENER',ENER   ENE01610
   53       DIKG=DIKG+ENER                                              ENE01620
       DEIE=DEIE+DIKG                                                   ENE01630
      ENDIF                                                             ENE01640
C*                                                                      ENE01650
      IF ( ISP1 .EQ. 1 ) THEN                                           ENE01660
         DO 61 I=L3,L4                                                  ENE01670
   61      WORK(I)=DDISP(I)                                             ENE01680
C...     ADD GROUND RESPONSE IN JOINT CORRDINATES TO RELATIVE RESP      ENE01690
C          MULTIPLY GROUND DISPL * DIRECTION COSINE                     ENE01700
C          OMIT ALL DOF THAT ARE CONSTRAINED TO A MASTER DOF...         ENE01710
         IF (GD(1).NE.0 .OR. GD(2).NE.0 .OR. GD(3).NE.0) THEN           ENE01720
           DO 62 NODE=1,NNODE                                           ENE01730
             ICOS=JTCOS(NODE)                                           ENE01740
             DO 62 K=1,3                                                ENE01750
             IF (.NOT. BTEST(JTFLG(NODE),K+11)) THEN                    ENE01760
               II=IDOF(NODE,K)                                          ENE01770
               IF (II.LE.L4 .AND. II.GE.L3)                             ENE01780
     &         WORK(II)=WORK(II)        + GD(1)*COSINE(1,K,ICOS)        ENE01790
     &         + GD(2)*COSINE(2,K,ICOS) + GD(3)*COSINE(3,K,ICOS)        ENE01800
             ENDIF                                                      ENE01810
   62        CONTINUE                                                   ENE01820
         ENDIF                                                          ENE01830
         DEINT=0                                                        ENE01840
          DO 57 I=L3,L4                                                 ENE01850
            ENER= SGD(I,1)*WORK(I)                                      ENE01860
C           WRITE(6,*)'SGD:',SGD(I,1)                                   ENE01870
   57       DEINT=DEINT+ENER                                            ENE01880
            DEIE=DEIE+DEINT                                             ENE01890
      ENDIF                                                             ENE01900
C*                                                                      ENE01910
C..... CALCULATE THE TOTAL INPUT ENERGY                                 ENE01920
      EIE=EIE + DEIE                                                    ENE01930
C                                                                       ENE01940
C     IF( ISP1 .EQ. 1 ) THEN                                            ENE01950
C       EIE=EIE+ESE                                                     ENE01960
C       ISP1=0                                                          ENE01970
C     ENDIF                                                             ENE01980
C                                                                       ENE01990
C                                                                       ENE02000
C=======================================                                ENE02010
C===  CALCULATE TOTAL STRAIN ENERGY   ===                               ENE02020
C=======================================                                ENE02030
C..... CALCULATE THE TOTAL STRAIN ENERGY                                ENE02040
      DESE=ESE                                                          ENE02050
      DPSE=PSE                                                          ENE02060
      ESE=0                                                             ENE02070
      PSE=0                                                             ENE02080
C                                                                       ENE02090
C     ------ GET ELASTIC AND PLASTIC STRAIN ENERGY FOR EACH ELEMENT     ENE02100
      LSTTYP=0                                                          ENE02110
      PRINT=.FALSE.                                                     ENE02120
      HEAD =.FALSE.                                                     ENE02130
      DO 60 IELNO=1,NELMT                                               ENE02140
        CALL ELELIB                                                     ENE02150
     &  (10,LSTTYP ,.FALSE.,IREL,.FALSE.,  NAME ,EESE,EPSE,DAMAGE,      ENE02160
     &   DUCFAG,                                                        ENE02170
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,I2DISP,IZLOAD,MSTOR)    ENE02180
        IF (PT2) WRITE (6,55) IELNO,EESE,EPSE                           ENE02190
   55   FORMAT('   ELEMENT#',I5,'   ELASTIC SE=',1P,G13.4,              ENE02200
     &                          '   PLASTIC SE=',1P,G13.4)             ENE02210
        ESE = ESE + EESE                                                ENE02220
   60   PSE = PSE + EPSE                                                ENE02230
C                                                                       ENE02240
C..... CALCULATE THE INCRIMENTAL STRAIN ENERGY                          ENE02250
       DESE=ESE-DESE                                                    ENE02260
       DPSE=PSE-DPSE                                                    ENE02270
C                                                                       ENE02280
C=================================                                      ENE02290
C===  CALCULATE KINETIC ENERGY  ===                                     ENE02300
C=================================                                      ENE02310
      IF (DYN) THEN                                                     ENE02320
C                                                                       ENE02330
C..... GET THE ABSOLUTE VELOCITY AND STORE IN VECTOR WORK2.....         ENE02340
C...  SAVE RELATIVE RESPONSE IN VECTOR WORK2                            ENE02350
      DO 15 I=1,L4                                                      ENE02360
   15   WORK2(I)=VEL(I)+DVEL(I)                                         ENE02370
```

```
C...     ADD GROUND RESPONSE IN JOINT CORRDINATES TO RELATIVE RESP
C          MULTIPLY GROUND VELOCITY * DIRECTION COSINE
C          OMIT ALL DOF THAT ARE CONSTRAINED TO A MASTER DOF...
         IF (GVT(1).NE.0 .OR. GVT(2).NE.0 .OR. GVT(3).NE.0) THEN
           DO 20 NODE=1,NNODE
             ICOS=JTCOS(NODE)
             DO 20 K=1,3
             IF (.NOT. BTEST(JTFLG(NODE),K+11)) THEN
               II=IDOF(NODE,K)
               IF (II.LE.L4)
     &         WORK2(II)=WORK2(II)        + GVT(1)*COSINE(1,K,ICOS)
     &         + GVT(2)*COSINE(2,K,ICOS) + GVT(3)*COSINE(3,K,ICOS)
             ENDIF
  20         CONTINUE
          ENDIF
C
C..... CALCULATE THE TOTAL KE
         DEKE=EKE
         CALL ETMPLY(PT2,L1,L2,L3,L4,L41,L4,
     &     IMDMS,MDMS,MASS,WORK,WORK2,WORK2,EKE)
         EKE=EKE/2.0
C
C..... CALCULATE THE INCRIMENTAL KE
         DEKE=EKE-DEKE
       ELSE
C
C..... ZERO KE FOR STATIC CASE.
         DEKE=0
         EKE =0
       ENDIF
C
C------------------------------------
C---  CALCULATE DAMPINED ENERGY   ---
C------------------------------------
       IF (DYN) THEN
C
C..... CALCULATE THE INCRIMENTAL DAMPING ENERGY
         DEDD =0
         IF (ALPHA.NE.0 .OR. BETA.NE.0) THEN
           DO 85 I=L3,L4
              FORCEI= FDAMP(I)+FFDAMP(I)/2
              VMI = VEL(I)+ DVEL(I)/2
              ENER=FORCEI*VMI
              IF (PT2) WRITE (6,*) I,
     &          'FORCEI',FORCEI,'  VM',VMI,'  ENER',ENER
  85       DEDD=DEDD + ENER
           DEDD=DEDD*DT
         ENDIF
C
C..... CALCULATE THE TOTAL DAMPINED ENERGY
         EDD=EDD+DEDD
       ELSE
C
C..... ZERO DE FOR STATIC CASE.
         EDD=0
         DEDD=0
       ENDIF
C
C-------------------------------
C---  CHECK ENERGY BALANCE   ---
C-------------------------------
       DIFF=EIE-ESE-EKE-PSE-EDD
       IF (EIE.NE.0) THEN
          RE=DIFF/EIE * 100
       ELSE
          RE=100.00
       ENDIF
C
C--------------------
C---  PRINT DATA ---
C--------------------
       IF (WRITE)
     &WRITE (6,100) DEIE,EIE,DESE,ESE,DEKE,EKE,DPSE,PSE,DEDD,EDD,RE
  100 FORMAT (/,T10,' ENERGY BALANCE'//
     &        T10,'================',
     & T40,'INCRIMENTAL',T58,'TOTAL'/
     & T15,'INPUT ENERGY'              ,T40,1P,G12.5,T55,G12.5/
     & T15,'ELASTIC STRAIN ENERGY'     ,T40,1P,G12.5,T55,G12.5/
     & T15,'KINETIC ENERGY '           ,T40,1P,G12.5,T55,G12.5/
     & T15,'PLASTIC STRAIN ENERGY'     ,T40,1P,G12.5,T55,G12.5/
     & T15,'DAMPING ENERGY'            ,T40,1P,G12.5,T55,G12.5/
     & T15,'RELATIVE ER','ROR'         ,              T55,G12.5,'%')
       IF (DYN.AND.WRITE) WRITE (6,110) ( GD(I), GV(I), GA(I),I=1,3),
     &                                  (GDT(I),GVT(I),GAT(I),I=1,3)
  110 FORMAT (/,T10,' GROUND RESPONSE' /
     &        T10,'================='/
     & T10,'INCRIMENTAL VALUES',T46,'DISPLACEMENT',
     &   T63,'VELOCITY',T76,'ACCELERATION'/
     & T15,'GLOBAL X AXIS TRANSLATION...',T45,1P,3G15.5,/
     & T15,'GLOBAL Y AXIS TRANSLATION...',T45,1P,3G15.5,/
     & T15,'GLOBAL Z AXIS TRANSLATION...',T45,1P,3G15.5,/
     & T10,'TOTAL VALUES'/
     & T15,'GLOBAL X AXIS TRANSLATION...',T45,1P,3G15.5,/
     & T15,'GLOBAL Y AXIS TRANSLATION...',T45,1P,3G15.5,/
     & T15,'GLOBAL Z AXIS TRANSLATION...',T45,1P,3G15.5)
C
C----------------------------------------
C---  DUCTILITIES AND EXCURSION RATIOS ---
C----------------------------------------
       LSTTTP=0
       PRINT=WRITE
       HEAD =WRITE
       DO 200 IELNO=1,NELMT
  200    CALL ELELIB
     &   (11,LSTTTP ,.FALSE.,IREL,.FALSE., NAME ,EESE,EPSE,DAMAGE,
     &    DUCFAG,
     &    IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,I$DISP,I$LOAD,MSTOR)
C
       RETURN
       END
C-----------------------------------------------------------------------
C          DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE
C          END DEBUG
C-----------------------------------------------------------------------
C   CALC THE ENERGY (E) FOR U(TRANSP) * P = U(TRANSP) * A * X
C
C - MATRIX A IS AN UPPER TRIANGULAR MATRIX, WITH MAIN DIAGONAL ADDRESS
C     MATRIX MD.
C     DOF L1 TO L2 HAVE BEEN CONDENSED OUT
C     DOF L3 TO L4 HAVE BEEN MODIFIED FOR THE CONDENSATION OF L1 TO L2
C     DOF L5 TO L6 HAVE NOT BEEN MODIFIED
C - X IS A VECTOR.
C - P IS A DUMMY INPUT VECTOR.
C-----------------------------------------------------------------------
       SUBROUTINE ETMPLY(PRINT,L1,L2,L3,L4,L5,L6,IMD,MD,A,P,X,U,E)
       DOUBLE PRECISION E,E1
       DIMENSION A(IMD),X(L6),P(L6),MD(L6+1),U(L6)
       LOGICAL BTEST,PRINT,T,F
       IJI(I,J)=MD(J)+J-I
C
       T=.TRUE.
       F=.FALSE.
C
       IF (PRINT) THEN
          WRITE (6,500) 'CALC ENERGY FOR PARTIALLY REDUCED A MATRIX',
     &          L1,L2,L3,L4,L5,L6,ZERO,X3
          CALL LMATRX(A,MD,L6,IMD,'MATRIX A ')
          DO 515 I=L1,L6
  515       WRITE(6,520) I,X(I),U(I)
       ENDIF
C
C------------ CALC PF' = KFF'*X1
       CALL TMPLY(F,T,IMD,L6,L3,L4,L3,L4,MD,A,P,X)
C
C------------ CALC E1 = XF' * PF'
       E=0
       DO 10 L=L3,L4
    10   E = E + U(L)*P(L)
       E1=E
C
C------------ CALC PF  = KFR *XR
       CALL TMPLY(F,T,IMD,L6,L1,L4,L5,L6,MD,A,P,X)
C
C------------ CALC PF  = KFR *XR
       CALL TMPLY(F,T,IMD,L6,L5,L6,L1,L6,MD,A,P,X)
C
C------------ CALC E2 + E3 + E4 = XF * PF + XR * PR
       DO 20 L=L1,L6
   20    E = E + U(L)*P(L)
C
C------------ PRINT ANSWER ...
       IF (PRINT) WRITE (6,510) E,E1
C
  500 FORMAT (/2X,A,/2X,'L1:',I5,'   L2:',I5,
     &  &'   L3:',I5,'   L4:',I5,
     &  &'   L5:',I5,'   L6:',I5,'  ZERO: ',L1,'    X3: ',L1)
  510 FORMAT (/2X,' TOTAL ENERGY:',1P,G13.5,'   E1:',G13.5)
  520 FORMAT (2X,'DOF:',I6,'  X(DOF)',1P,G14.5,'  U(DOF):',G14.5)
       RETURN
       END
C-----------------------------------------------------------------------
C          DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE
C          END DEBUG
       SUBROUTINE FEMREL(MA,MB,RELA,RELB,ITORQ,IELNO)
       REAL    MA,MB
       INTEGER REL
       LOGICAL RELA,RELB,BTEST
C      WRITE (6,*) RELA,RELB,MA,MB,ITORQ
C
       IF (ITORQ.EQ.0) THEN
            IF (RELA .AND. RELB) THEN
               MA =0
               MB=0
            ELSE IF (RELB) THEN
               MA = MA-MB/2
               MB=0
            ELSE IF (RELA) THEN
               MB= MB-MA/2
               MA =0
            ENDIF
       ELSE
            IF (RELA .AND. RELB) THEN
               WRITE (6,900) IELNO,'TORSION WITH BOTH ENDS RELEASED'
               MA=0
               MB=0
            ELSE IF (RELB) THEN
               MA=MA+MB
               MB=0
            ELSE IF (RELA) THEN
               MB=MB+MA
               MA =0
            ENDIF
       ENDIF
C
  900 FORMAT('  ER','ROR CALCULATING FIXED END FORCES FOR ELEMENT #',
     &  I6,' ',A,' IS NOT AVAILABLE')
       RETURN
       END
C            STRIPS PRECEEDING BLANK SPACES OFF A CHARACTER STRING
C          DEBUG UNIT(6),SUBCHK,SUBTRACE
C          END DEBUG
C
       SUBROUTINE FIRST(WORD)
       LOGICAL BTEST
       CHARACTER*(*) WORD
            IF (INDEX(WORD,' ').EQ.0) RETURN
       DO 10 I=1,(LEN(WORD))
            J=INDEX(WORD(I:),' ')
            IF (J.EQ.1) GO TO 10
                 WORD=WORD(I:)
                 RETURN
    10  CONTINUE
    20  RETURN
       END
C-----------------------------------------------------------------------
C          DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE
C          END DEBUG
C-----------------------------------------------------------------------
       SUBROUTINE FORM(IOPT)
       CHARACTER*80  NAME
       CHARACTER*130 ELNO
       LOGICAL PRINT,HEAD,FALSE,AXIAL,BTEST,BUG5,BUG57
       INCLUDE (ZCOMN)
C
C-----------------------------------------------------------------------
C   MATRIX STORAGE SCHEME:
C  -THE STIFFNESS MATRIX (K) IS BANDED AND SYMMETRIC.
C   THUS ONLY THE SKYLINE ABOVE THE MAIN DIAGONAL NEEDS TO BE STORED.
C   THE MATRIX K IS STORED IN A LINEAR ARRAY BY COLUMNS.
C   THE VALUE ON THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST.
C   RELATIVE ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE
C   MAIN DIAGONAL TERMS ARE STORED IN MD.
C   THE ABSOLUTE ADDRESS OF THE FIRST ELEMENT IN THE K MATRIX IS I%STIF.
C   THE STIFFNESS MATRIX K IS STORED IN THE Z ARRAY, Z(I%STIF)=K(1,1)
C   THE ABSOLUTE ADDRESS OF THE FIRST ELEMENT IS THE MD MATRIX IS I%MD.
C   THE STIFFNESS MATRIX K IS STORED IN THE NZ ARRAY, NZ(I%MD)=MD(1)
C   THUS: IF I LE J (UPPER TRIANGULAR MATRIX)
C            IF I%MD-1+J < I%MD-1+J+1
C                  IJ=NZ(I%MD-1+J)+J-I  : RELATIVE ADDRESS IF ELEMENT (I,J)
C                  B(I,J)=Z(I%STIF-1 + IJ)
C         ELSE
C                  B(I,J)=0             : (I,J) IS ABOVE THE SKYLINE
C
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW...
C        |  1  |  3  |  |  12 |
C        |     |  2  |  5  |  11 |   NOTE: B(1,3),B(1,4) AND B(2,4) ARE ZERO.
C   B=   |     |     |  4  |  7  |  10 |
C        |     | SYMM.     |  6  |  9  |   MD= 1, 2, 4, 6, 8, 13
C        |     |           |     |  8  |
C   SAY I%MD=121,I%STIF=200
C   NZ(121)=1          Z(200)=1
C   NZ(122)=2          Z(201)=2
C   NZ(123)=4          Z(202)=3
C   NZ(124)=6          Z(203)=4
C   NZ(125)=8          Z(204)=5
C   NZ(126)=13         Z(205)=6
C                      Z(206)=7
C                      Z(206)=8
C                      Z(206)=9
C                      Z(206)=10
C                      Z(206)=11
C                      Z(206)=12
C   THUS, FOR B(3,4)  I=3, J=4
C              IJ=NZ(I%MD-1+J)+J-I = NZ(121-1+4) +4-3 = 6+4-3 =7
C              B(I,J)=Z(I%STIF-1 + IJ) = Z(200-1+7) = Z(206) = 7  OK
C   THUS, FOR B(2,4)  I=2, J=4
C              IJ=NZ(I%MD-1+J)+J-I = NZ(121-1+4) +4-2 = 6+4-2 =8
C              NZ(I%MD -1+J-1)=NZ(121-1+4-1)=8
C              SINCE IJ GE NZ(I%MD-1+J+1)
C              B(I,J)=0  (I,J) IS ABOVE THE SKYLINE....
C-----------------------------------------------------------------------
C
       BUG5 = BTEST(IBUG,5)
```

```
      BUG57= BTEST(IBUG,7) .AND. BUG5                                   FOR00600
                                                                        FOR00610
                                                                        FOR00620
      LSTTYP=0                                                          FOR00630
      PRINT=.FALSE.                                                     FOR00640
      HEAD =.FALSE.                                                     FOR00650
C                                                                       FOR00660
C-----------------------------------------------------------------------FOR00670
C IOPT=1 FORM STIFFNESS MATRIX                                          FOR00680
C IOPT=2 FORM GEOMETRIC STIFFNESS MATRIX                                FOR00690
C-----------------------------------------------------------------------FOR00700
C                                                                       FOR00710
       IF (BUG5)  WRITE (6,*) ' IN ROUTINE FORM IOPT=',IOPT             FOR00720
C                                                                       FOR00730
   49 FORMAT (1X,A/('  I=',I3,'  MDMASS(I)=',I11))                     FOR00740
C                                                                       FOR00750
      GO TO (100,200),IOPT                                              FOR00760
C                                                                       FOR00770
C=========================                                              FOR00780
C= ASSEMBLE STIFFNESS  ====                                            FOR00790
C=========================                                              FOR00800
  100 CONTINUE                                                          FOR00810
          IST =I%STIF                                                   FOR00820
          IEND=I%STIF +N%(I%MD+NDOF) -1                                 FOR00830
C                                                                       FOR00840
C----------- ZERO STIFFNESS                                             FOR00850
      DO 110 I=IST,IEND                                                 FOR00860
  110   Z(I)=0                                                          FOR00870
C                                                                       FOR00880
C----------- LOOP FOR EACH ELEMENT                                      FOR00890
      DO 120 IELNO=1,NELMT                                              FOR00900
C                                                                       FOR00910
C----------- GET STIFFNESS OF EACH ELEMENT                              FOR00920
       FALSE=.FALSE.                                                    FOR00930
       CALL ELELIB                                                      FOR00940
     &   (3,LSTTYP ,FALSE ,IREL,FALSE ,NAME ,EESE,EPSE,DAMAGE,         FOR00950
     &   DUCFAG,                                                        FOR00960
     &   IELNO ,IELDOF,KGDOF,PGEOM,Z   ,AXIAL,I%DISP,I%LOAD,MSTOR)     FOR00970
C                                                                       FOR00980
       IF (BUG57) THEN                                                  FOR00990
         WRITE (ELNO,15)  IELNO,(N%(I+I%LM-1),I=1,MIN(IELDOF,18))       FOR01000
   15    FORMAT ('ELEMENT #',I6,' LM=',18I6)                           FOR01010
         CALL LMATRX(Z(I%KE),MD,IELDOF,MD(IELDOF+1),ELNO)               FOR01020
       ENDIF                                                            FOR01030
C                                                                       FOR01040
      CALL FORML(Z(I%STIF),N%(I%MD),NDOF,N%(I%MD+NDOF),                 FOR01050
     &     Z(I%KE),MD,IELDOF,MD(IELDOF+1),N%(I%LM),1.)                 FOR01060
C                                                                       FOR01070
C----------- SUBTRACT GEOMETRIC STIFFNESS                               FOR01080
      N%(I%KGDT+3)=N%(I%KGDT+3)                                         FOR01090
C                                                                       FOR01100
      IF (N%(I%KGDT+3).EQ.1) THEN                                       FOR01110
        FALSE=.FALSE.                                                   FOR01120
        CALL ELELIB                                                     FOR01130
     &   (2,LSTTYP ,FALSE ,IREL,FALSE, NAME ,EESE,EPSE,DAMAGE,         FOR01140
     &   DUCFAG,                                                        FOR01150
     &   IELNO ,IELDOF,KGDOF,PGEOM,Z   ,AXIAL,I%DISP,I%LOAD,MSTOR)     FOR01160
C                                                                       FOR01170
        KGDOF=KGDOF                                                     FOR01180
C                                                                       FOR01190
        IF (BUG57) THEN                                                 FOR01200
          WRITE (ELNO,15)  IELNO,(N%(I+I%LMKG-1),I=1,MIN(KGDOF,18))     FOR01210
          CALL LMATRX(Z(I%KEKG),MD,KGDOF,MD(KGDOF+1),ELNO)              FOR01220
        ENDIF                                                           FOR01230
C                                                                       FOR01240
        IF (.NOT. AXIAL) GO TO 120                                      FOR01250
C            DETERMINE THE FACTOR FOR KG                                FOR01260
               IF (N%(I%KGDT+1).EQ.1) THEN                              FOR01270
                         ACCELZ= Z(I%KGDT)                              FOR01280
                         PGEOM=PGEOM*(1+ACCELZ)                         FOR01290
               ENDIF                                                    FOR01300
C     SWAP THE SIGN ON PGEOM FOR SUBTRACTION.....                       FOR01310
      PGEOM=-PGEOM                                                      FOR01320
C                                                                       FOR01330
      CALL FORML(Z(I%STIF),N%(I%MD),NDOF,N%(I%MD+NDOF),                 FOR01340
     &     Z(I%KEKG),MD,KGDOF,MD(KGDOF+1),N%(I%LMKG),PGEOM)            FOR01350
      ENDIF                                                             FOR01360
C                                                                       FOR01370
  120   CONTINUE                                                        FOR01380
C                                                                       FOR01390
      RETURN                                                            FOR01400
C                                                                       FOR01410
C----------- ASSEMBLE GEOMETRIC STIFNESS                                FOR01420
  200 CONTINUE                                                          FOR01430
      IF (N%(I%KGDT+3).NE.2 ) RETURN                                    FOR01440
          IST =I%KG                                                     FOR01450
          IEND=I%KG + N%(I%MDKG+NDOF) - 1                               FOR01460
C                                                                       FOR01470
C----------- ZERO STIFFNESS                                             FOR01480
      DO 210 I=IST,IEND                                                 FOR01490
  210   Z(I)=0                                                          FOR01500
C                                                                       FOR01510
C------------ LOOP FOR EACH ELEMENT                                     FOR01520
      DO 220 IELNO=1,NELMT                                              FOR01530
C                                                                       FOR01540
C----------- GET STIFFNESS OF EACH ELEMENT                              FOR01550
       FALSE=.FALSE.                                                    FOR01560
       CALL ELELIB                                                      FOR01570
     &   (2,LSTTYP ,FALSE ,IREL,FALSE, NAME ,EESE,EPSE,DAMAGE,         FOR01580
     &   DUCFAG,                                                        FOR01590
     &   IELNO ,IELDOF,KGDOF,PGEOM,Z   ,AXIAL,I%DISP,I%LOAD,MSTOR)     FOR01600
C                                                                       FOR01610
       IF (.NOT. AXIAL) GO TO 220                                       FOR01620
C                                                                       FOR01630
       IF (BUG57) THEN                                                  FOR01640
         WRITE (6,*)  'PGEOM=',PGEOM                                    FOR01650
         WRITE (ELNO,15)  IELNO,(N%(I+I%LMKG-1),I=1,MIN(KGDOF,18))      FOR01660
         CALL LMATRX(Z(I%KEKG),MD,KGDOF,MD(KGDOF+1),ELNO)               FOR01670
       ENDIF                                                            FOR01680
C                                                                       FOR01690
      CALL FORML(Z(I%KG),N%(I%MDKG),NDOF,N%(I%MDKG+NDOF),               FOR01700
     &     Z(I%KEKG),MD,KGDOF,MD(KGDOF+1),N%(I%LMKG),PGEOM)            FOR01710
  220   CONTINUE                                                        FOR01720
C                                                                       FOR01730
      RETURN                                                            FOR01740
      END                                                               FOR01750
C=======================================================================FOR00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE                                 FOR00020
C         END DEBUG                                                     FOR00030
      SUBROUTINE FORML(K,MD,NDOF,IMD,KE,MDKE,IELDOF,IMDKE,LM,FACTOR)    FOR00040
C**   DOUBLE PRECESION K                                                FOR00050
      LOGICAL BTEST                                                     FOR00060
      DIMENSION MD(NDOF+1),MDKE(IELDOF+1),LM(IELDOF)                    FOR00070
      REAL K(IMD),KE(IMDKE)                                             FOR00080
C                                                                       FOR00090
C-----------------------------------------------------------------------FOR00100
C   MATRIX STORAGE SCHEME:                                              FOR00110
C  -THE STIFFNESS MATRIX (K) IS BANDED AND SYMMETRIC.                   FOR00120
C   THUS ONLY THE SKYLINE ABOVE THE MAIN DIAGONAL NEEDS TO BE STORED.   FOR00130
C   THE MATRIX K IS STORED IN A LINEAR ARRAY BY COLUMNS.                FOR00140
C   THE VALUE ON THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST. FOR00150
C   RELATIVE ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE      FOR00160
C   MAIN DIAGONAL TERMS ARE STORED IN MD.                               FOR00170
C   THUS: FOR K(I,J)                                                    FOR00180
C         IF I LE J (UPPER TRIANGULAR MATRIX)                           FOR00190
C         IF MD(J)+J-I < MD(J+1)                                        FOR00200
C               IJ=MD(J)+J-I      ! RELATIVE ADDRESS IF ELEMENT (I,J)   FOR00210
C               K(I,J)=K(IJ)                                            FOR00220
C          ELSE                                                         FOR00230
C               K(I,J)=0             ! (I,J) IS ABOVE THE SKYLINE       FOR00240
C                                                                       FOR00250
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW...             FOR00260
C       |  1  |  3  |     |     |  12 |                                 FOR00270
C       |     |  2  |  5  |     |  11 |  NOTE: K(1,3),K(1,4) AND K(2,4) ARE ZERO. FOR00280
C   K=  |     |     |  4  |  7  |  10 |                                 FOR00290
C       |  SYMM.    |     |  6  |  9  |  MD= 1, 2, 4, 6, 8, 13          FOR00300
C       |           |     |     |  8  |                                 FOR00310
C   SAY IZMD=121,I%STIF=200                                             FOR00320
C   MD(1)=1           K( 1)=1                                           FOR00330
C   MD(2)=2           K( 2)=2                                           FOR00340
C   MD(3)=4           K( 3)=3                                           FOR00350
C   MD(4)=6           K( 4)=4                                           FOR00360
C   MD(5)=8           K( 5)=5                                           FOR00370
C   MD(6)=13          K( 6)=6                                           FOR00380
C                     K( 7)=7                                           FOR00390
C                     K( 8)=8                                           FOR00400
C                     K( 9)=9                                           FOR00410
C                     K(10)=10                                          FOR00420
C                     K(11)=11                                          FOR00430
C                     K(12)=12                                          FOR00440
C   THUS, FOR K(3,4)  I=3, J=4                                          FOR00450
C              IJ=MD(J)+J-I = MD(4)+4-3 = 6+4-3 =7                      FOR00460
C              K(I,J)=K(IJ) = K(7) =  7  OK                             FOR00470
C   THUS, FOR B(2,4)  I=2, J=4                                          FOR00480
C              IJ=MD(J)+J-I = MD(4) +4-2 = 6+4-2 =8                     FOR00490
C              MD(J+1)=MD(4+1)=8                                        FOR00500
C              SINCE IJ GE MD(J+1)                                      FOR00510
C              K(I,J)=0  (I,J) IS ABOVE THE SKYLINE....                 FOR00520
C-----------------------------------------------------------------------FOR00530
C   THE ELEMENT STIFFNESS IS STORED IN A SIMILAR MANNER.                FOR00540
C   KE IS THE LINEAR ARRAY CONTAINING THE ELEMENT STIFFNESS             FOR00550
C   MDKE IS THE ARRAY CONTAINING THE MAIN DIAGONAL ADDRESSES FOR KE     FOR00560
C-----------------------------------------------------------------------FOR00570
C                                                                       FOR00580
C----------- MAP ELEMENT STIFFNESS INTO GLOBAL STIFFNESS                FOR00590
C LI    = GLOBAL DOF ASSOCIATED WITH THE I'TH ROW OF THE MEMBER STIFF.  FOR00600
C LJ    = GLOBAL DOF ASSOCIATED WITH THE I'TH ROW OF THE MEMBER STIFF.  FOR00610
C IJELM = ABSOLUTE ADDRESS OF THE MEMBER STIFFNESS TERM (I,J)           FOR00620
C IJGLOB= ABSOLUTE ADDRESS OF THE GLOBAL STIFFNESS TERM (LI,LJ)         FOR00630
C                                                                       FOR00640
C*    MAXIJ = MD(NDOF+1)                                                FOR00650
C                                                                       FOR00660
      DO 30 I=1,IELDOF                                                  FOR00670
        LI=LM(I)                                                        FOR00680
C*      IF (LI.LE.0 .OR. LI.GT.NDOF) GO TO 30                           FOR00690
        DO 20 J=I,IELDOF                                                FOR00700
          LJ=LM(J)                                                      FOR00710
C*        IF (LJ.LE.0 .OR. LJ.GT.NDOF) GO TO 20                         FOR00720
          IJELM =MDKE(J)+J-I                                            FOR00730
          IF (KE(IJELM).EQ.0) GO TO 20                                  FOR00740
          IF (LI.LE.LJ) THEN                                            FOR00750
              IJGLOB=MD(LJ)+LJ-LI                                       FOR00760
          ELSE                                                          FOR00770
              IJGLOB=MD(LI)+LI-LJ                                       FOR00780
          ENDIF                                                         FOR00790
C*        IF ( IJGLOB.GE.MAXIJ ) THEN                                   FOR00800
C*          WRITE (6,*) 'ER','ROR IN MAPPING ELEMENT STIFFNESS FOR',    FOR00810
C*   &          ' ELEMENT #',IELNO,LI,LJ,IJGLOB,MAXIJ                  FOR00820
C*        ELSE                                                          FOR00830
            K(IJGLOB)=K(IJGLOB)+KE(IJELM) * FACTOR                      FOR00840
C*        ENDIF                                                         FOR00850
   20   CONTINUE                                                        FOR00860
   30 CONTINUE                                                          FOR00870
C                                                                       FOR00880
C----- NOTE: LINES COMMENTED OUT C* ARE FOR ER_ROR CHECKING, THE VALUES FOR00890
C           IN LM ARE ASSUMED TO BE CORRECT...                          FOR00900
C                                                                       FOR00910
      RETURN                                                            FOR00920
      END                                                               FOR00930
C=======================================================================GET00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE                                 GET00020
C         END DEBUG                                                     GET00030
C                                                                       GET00040
C SUBROUTINE GETCHR DETERMINS THE VALUE OF THE FIRST CHARACTER          GET00050
C                   AFTER AN OPTIONAL (LWORD=.TRUE.) KEYWORD (WORD).    GET00060
C                   AND REMOVES THE STRING (WORD)//CHAR    IF           GET00070
C                   MODIFY=.TRUE.                                       GET00080
C                                                                       GET00090
      SUBROUTINE GETCHR(VERB,WORD,CHAR,LWORD,MODIFY)                    GET00100
C                                                                       GET00110
      CHARACTER*(*) VERB,WORD,CHAR                                      GET00120
      LOGICAL MODIFY, LWORD ,BTEST                                      GET00130
C                                                                       GET00140
 999  CHAR=' '                                                          GET00150
      IF (LWORD) THEN                                                   GET00160
           ISTART=INDEX(VERB,WORD)                                      GET00170
           IF (ISTART.LE.0) RETURN                                      GET00180
           I=ISTART+LEN(WORD)                                           GET00190
      ELSE                                                              GET00200
           I=1                                                          GET00210
           ISTART=1                                                     GET00220
           CALL FIRST(VERB)                                             GET00230
      ENDIF                                                             GET00240
C                                                                       GET00250
      IF (VERB(I:I).EQ.'''' ) THEN                                      GET00260
         I=I+1                                                          GET00270
         IJ=I                                                           GET00280
  10     J=INDEX(VERB(IJ:),'''')-2  + IJ                               GET00290
           IF (VERB((J+2):(J+2)).EQ.'''') THEN                          GET00300
              IJ=J+3                                                    GET00310
              GO TO 10                                                  GET00320
           ENDIF                                                        GET00330
         IF (J.EQ.0) THEN                                               GET00340
           CHAR=VERB(I: )                                               GET00350
           GO TO 10                                                     GET00360
         ELSE                                                           GET00370
           CHAR=VERB(I:J)                                               GET00380
         ENDIF                                                          GET00390
  20     IJ=INDEX(CHAR,'''''')                                          GET00400
         IF (IJ.NE.0) THEN                                              GET00410
           CHAR=CHAR(:IJ)//CHAR((IJ+2):)                                GET00420
           GO TO 20                                                     GET00430
           ENDIF                                                        GET00440
      ELSE                                                              GET00450
         J=INDEX(VERB(I:),' ') + I -2                                   GET00460
         IF (J.GE.I) CHAR=VERB(I:J)                                     GET00470
      ENDIF                                                             GET00480
C                                                                       GET00490
      IF (MODIFY) THEN                                                  GET00500
         IF (VERB(J+2:J+2).EQ.',') J=J+1                                GET00510
         VERB(ISTART: )=VERB(J+2: )                                     GET00520
         CALL FIRST(VERB)                                               GET00530
      ENDIF                                                             GET00540
C                                                                       GET00550
      RETURN                                                            GET00560
      END                                                               GET00570
C=======================================================================GET00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE                                 GET00020
C         END DEBUG                                                     GET00030
C                                                                       GET00040
C SUBROUTINE GETINT DETERMINS THE VALUE OF THE FIRST INTEGER (INTEG)    GET00050
C                   AFTER AN OPTIONAL (LWORD=.TRUE.) KEYWORD (WORD).    GET00060
C                   AND REMOVES THE STRING (WORD)//INTEGER IF           GET00070
C                   MODIFY=.TRUE.  INTEG CAN BE ANY LENGTH REPRESENTABLEGET00080
C                   BY SINGLE PRECESION.                                GET00090
C                                                                       GET00100
      SUBROUTINE GETINT(VERB,WORD,INTEG,DEFULT,LWORD,MODIFY)            GET00110
C                                                                       GET00120
      CHARACTER*(*) VERB,WORD                                           GET00130
      CHARACTER*10  NUMB                                                GET00140
      LOGICAL MODIFY, LWORD, HIT, BTEST                                 GET00150
      INTEGER DEFULT                                                    GET00160
      DATA NUMB/'0123456789'/                                           GET00170
C                                                                       GET00180
```

```
      INTEG=DEFULT                                                      GET00190
      HIT=.TRUE.                                                        GET00200
      IF (LWORD) THEN                                                   GET00210
           ISTART=INDEX(VERB,WORD)                                      GET00220
           IF (ISTART.LE.0) RETURN                                      GET00230
           I=ISTART+LEN(WORD)                                           GET00240
      ELSE                                                              GET00250
           I=1                                                          GET00260
           ISTART=1                                                     GET00270
      ENDIF                                                             GET00280
C                                                                       GET00290
      ISIGN = 1                                                         GET00300
      L=LEN(VERB)                                                       GET00310
    5 IF (I.GT.L) RETURN                                                GET00320
      IF (VERB(I:I).EQ.'-') THEN                                        GET00330
         ISIGN =-1                                                      GET00340
         I=I+1                                                          GET00350
         GO TO 5                                                        GET00360
      ELSE IF (VERB(I:I).EQ.'+' .OR.VERB(I:I).EQ.' ') THEN              GET00370
         I=I+1                                                          GET00380
         GO TO 5                                                        GET00390
      ENDIF                                                             GET00400
C                                                                       GET00410
   10 IF (LGE(VERB(I:I),'0') .AND. LLE(VERB(I:I),'9') ) THEN            GET00420
         IF ( HIT) THEN                                                 GET00430
              HIT=.FALSE.                                               GET00440
              INTEG=0                                                   GET00450
         ENDIF                                                          GET00460
         DO 20 J=1,10                                                   GET00470
          IF(VERB(I:I).EQ. NUMB(J:J) ) THEN                             GET00480
               INTEG=10*INTEG+ (J-1)                                    GET00490
               I=I+1                                                    GET00500
               GO TO 10                                                 GET00510
          ENDIF                                                         GET00520
   20    CONTINUE                                                       GET00530
      ENDIF                                                             GET00540
C                                                                       GET00550
      INTEG=INTEG*ISIGN                                                 GET00560
C                                                                       GET00570
      IF (MODIFY) THEN                                                  GET00580
         IF (VERB(I:I).EQ.',') I=I+1                                    GET00590
         VERB(ISTART: )=VERB(I: )                                       GET00600
      ENDIF                                                             GET00610
C                                                                       GET00620
      RETURN                                                            GET00630
      END                                                               GET00640
C********** HYST02 ***************************************************HYS00010
C       DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                        HYS00020
C       END DEBUG                                                       HYS00030
C*************************************************************************HYS00040
C     SU_BROUTINE AXLMOD - AXIAL HYSTERESIS MODEL                       HYS00050
C                          REFERANCE: ANALYSIS OF THE FULL SCALE SEVEN  HYS00060
C                                     STORY R/C TEST STRUCTURE.         HYS00070
C                                                                       HYS00080
C     P     = CURRENT LOAD, POSITIVE IS TENSION                         HYS00090
C     D     = CURRENT DISPLACMENT, POSITIVE IS ELONGATION               HYS00100
C     S     = STIFFNESS                                                 HYS00110
C     KC    = VIRGIN COMPRESSION STIFFNESS. EC                          HYS00120
C     KT1   = PRE-YEILD VIRGIN TENSILE STIFFNESS. 0.9*KC                HYS00130
C     KT2   = POST-YEILD VIRGIN TENSILE STIFFNESS, 0.001*KC             HYS00140
C     PY    = TENSILE YEILD POINT, PHO*PY                               HYS00150
C     FMAX  = MAXIMUN TENSILE FORCE BEFORE UNLOADING                    HYS00160
C     DMAX  = MAXIMUN TENSILE DISPLACMENT BEFORE UNLOADING              HYS00170
C     FX    = FORCE AT STIFFNESS CHANGE POINT ON THE UNLOADING CURVE    HYS00180
C     DX    = DISPL. AT STIFFNESS CHANGE POINT ON THE UNLOADING CURVE   HYS00190
C     PI,DI = INTERSECTION OF TWO LINES                                 HYS00200
C     DYT   = TENSILE YEILD DISPLACMENT = PY/KT1                        HYS00210
C     DYC   = COMPRESSION YEILD DISPLACMENT = PY/KC                     HYS00220
C     FP    = FORCE AT COMPRESSION STIFFNESS CHANGE POINT               HYS00230
C     DP    = COMPRESSION  AT COMPRESSION STIFFNESS CHANGE POINT        HYS00240
C     KR    = UNLOADING STIFFNESS = KC/(DMAX/DYT)**(.20)                HYS00250
C     KS    = LOADING STIFFNESS, LINE BETWEEN (FX,DX) AND (FP,DP)       HYS00260
C     KT    = LOADING STIFFNESS, LINE BETWEEN (FP,DP) AND (-2PY,-2DY)   HYS00270
C                                                                       HYS00280
C                                                                       HYS00290
      SUBROUTINE HYST02(P,D,PL,DL,V,VL,S,HRLN,KC,KT1,KT2,PY,ALPHA,BETA, HYS00300
     &    DYT,DYC,CSE,FMAX,DMAX,FX,DX,FP,DP,KR,KS,KT,NRL,A,PRINT,        HYS00310
     &    SEQ,ESE,PSE,PSEOLD,UPOS,UNEG,EXCR,IDR)                         HYS00320
C                                                                       HYS00330
      REAL*4 KC,KT1,KT2,KR,KS,KT,NRL                                    HYS00340
      DIMENSION UPOS(3),UNEG(3),EXCR(6),ESE(3)                          HYS00350
      INTEGER RULEI                                                     HYS00360
      LOGICAL LT,PRINT,PGTPL,PLTPL                                      HYS00370
C                                                                       HYS00380
 8000 IF (PRINT) WRITE (6,8010) P,D,PL,DL,V,VL,S,HRLN,KC,KT1,KT2,PY,    HYS00390
     &    ALPHA,BETA,DYT,DYC,FMAX,DMAX,FX,DX,FP,DP,KR,KS,KT,NRL,A,      HYS00400
     &    SEQ,ESE(3),PSE,PSEOLD,CSE,UPOS,UNEG,EXCR                      HYS00410
 8010 FORMAT(                                                           HYS00420
     &' AXLMOD-P=',G13.5,'    D=',G13.5,'   PL=',G13.5,'   DL=',G13.5,/ HYS00430
     &' AX-    V=',G13.5,'   VL=',G13.5,'    S=',G13.5,' HRLN=',G13.5,/ HYS00440
     &' AX-   KC=',G13.5,'  KT1=',G13.5,'  KT2=',G13.5,'   PY=',G13.5,/ HYS00450
     &' AX-ALPHA=',G13.5,' BETA=',G13.5,'  DYT=',G13.5,'  DYC=',G13.5,/ HYS00460
     &' AX- FMAX=',G13.5,' DMAX=',G13.5,'   FX=',G13.5,'   DX=',G13.5,/ HYS00470
     &' AX-   FP=',G13.5,'   DP=',G13.5,'   KR=',G13.5,'   KS=',G13.5,/ HYS00480
     &' AX-   KT=',G13.5,'  NRL=',G13.5,'    A=',G13.5,'  SEQ=',G13.5,/ HYS00490
     &' AX-  ESE=',G13.5,'  PSE=',G13.5,' PSEO=',G13.5,'  CSE=',G13.5,/ HYS00500
     &' AX-  UP1=',G13.5,'  UP2=',G13.5,'  UP3=',G13.5/                 HYS00510
     &' AX-  UN1=',G13.5,'  UN2=',G13.5,'  UN3=',G13.5/                 HYS00520
     &' AX-  EX1=',G13.5,'  EX2=',G13.5,'  EX3=',G13.5/                 HYS00530
     &' AX-  EX4=',G13.5,'  EX5=',G13.5,'  EX6=',G13.5)                 HYS00540
C                                                                       HYS00550
C..... SET INITIAL STIFFNESS AND RETURN                                 HYS00560
      IF (NRL.EQ.0) THEN                                                HYS00570
                 IF (P.LE.0) THEN                                       HYS00580
                      NRL=1                                             HYS00590
                      HRLN=1.1                                          HYS00600
                      S=KC                                              HYS00610
                      SEQ=KC                                            HYS00620
                      A=P                                               HYS00630
                 ELSE                                                   HYS00640
                      NRL=2                                             HYS00650
                      HRLN=2.1                                          HYS00660
                      S=KT1                                             HYS00670
                      SEQ=KC                                            HYS00680
                      A=P                                               HYS00690
                 ENDIF                                                  HYS00700
                 RETURN                                                 HYS00710
      ENDIF                                                             HYS00720
C                                                                       HYS00730
C..... RETURN IF NO STEP HAS BEEN TAKEN                                 HYS00740
      IF (P.EQ.PL .OR. D.EQ.DL ) RETURN                                 HYS00750
C                                                                       HYS00760
C..... SET LOADING FLAGS                                                HYS00770
      PGTPL=P.GT.PL                                                     HYS00780
      PLTPL=P.LT.PL                                                     HYS00790
C                                                                       HYS00800
C..... CALC EXCURSION RATIO AT EVERY ZERO CROSSING...                   HYS00810
      IF (P*PL.LE.0 .AND. PL.GT.0)                                      HYS00820
     &    CALL DNE(0,UPOS,EXCR,DYT,D,ESE,PSE,PSEOLD,CSE)                HYS00830
      IF (P*PL.LE.0 .AND. PL.LE.0) THEN                                 HYS00840
          EXCR(4)=0                                                     HYS00850
          EXCR(5)=0                                                     HYS00860
          EXCR(6)=0                                                     HYS00870
      ENDIF                                                             HYS00880
C                                                                       HYS00890
C..... CALC DUCTILITY IF LOAD GT PAST LOAD                              HYS00900
C                                                                       HYS00910
C     DUCTILITY FOR POSITIVE LOADS ONLY                                 HYS00920
      IF (P.GT.0 .AND. PGTPL)                                           HYS00930
     &    CALL DNE(1,UPOS,EXCR,DYT,D ,ESE(2),PSE,PSEOLD,CSE)            HYS00940
C                                                                       HYS00950
  999 CALL IDRECT(IDR,IDRV,IDRVO,P,PL,V,VL)                             HYS00960
      IDR=IDR                                                           HYS00970
                                                                        HYS00980
    1 IF (NRL.EQ.0) NRL=1                                               HYS00990
C                                                                       HYS01000
      RULEI=NRL                                                         HYS01010
C                                                                       HYS01020
      GO TO (100,200,300,400,500,600,700,800,900,                       HYS01030
     &                                1000,1100,1200,1300),RULEI        HYS01040
C                                                                       HYS01050
C RULE 1 - LOADING AND UNLOADING ON THE COMPRESSION BRANCH BEFORE YEILD HYS01060
C                                                                       HYS01070
  100 SEQ=KC                                                            HYS01080
      IF (P.GT.0) GO TO 200                                             HYS01090
      S=KC                                                              HYS01100
      NRL=1                                                             HYS01110
      HRLN=1.1                                                          HYS01120
      IF (PLTPL) A=MIN(10.*P,-2.*PY)                                    HYS01130
      IF (PGTPL) A=0.                                                   HYS01140
      RETURN                                                            HYS01150
C                                                                       HYS01160
C RULE 2 - TENSION LOADING BEFORE YEILD                                 HYS01170
C                                                                       HYS01180
  200 IF (IDR) 230,230,210                                              HYS01190
  210 IF (P.GE.PY) GO TO 600                                            HYS01200
      S=KT1                                                             HYS01210
      NRL=2                                                             HYS01220
      HRLN=2.1                                                          HYS01230
      A=PY                                                              HYS01240
      RETURN                                                            HYS01250
  230 IF (PL.LE.0) GO TO 210                                            HYS01260
      FMAX=AMAX1(P,FMAX)                                                HYS01270
      DMAX=AMAX1(D,DMAX)                                                HYS01280
      FX  =FMAX-PY                                                      HYS01290
      DX  =DMAX-PY/KC                                                   HYS01300
      S=KC                                                              HYS01310
      NRL=3                                                             HYS01320
      HRLN=2.2                                                          HYS01330
      A=FX                                                              HYS01340
      RETURN                                                            HYS01350
C                                                                       HYS01360
C RULE 3 - UNLOADING ON TENSION BEFORE YEILD                            HYS01370
C                                                                       HYS01380
  300 IF (PGTPL) GO TO 500                                              HYS01390
      IF (P.GT.0)  GO TO 330                                            HYS01400
  310 IF (KC.NE.KT1) THEN                                               HYS01410
         CALL INTSCT(DI,PI,D,P,KC,DX,FX,KT1)                            HYS01420
      ELSE                                                              HYS01430
         PI=P                                                           HYS01440
         DI=D                                                           HYS01450
      ENDIF                                                             HYS01460
      IF (P.LE.PI) GO TO 320                                            HYS01470
      S=KC                                                              HYS01480
      NRL=3                                                             HYS01490
      HRLN=3.11                                                         HYS01500
      A=PI                                                              HYS01510
      RETURN                                                            HYS01520
  320 S=KT1                                                             HYS01530
      NRL=4                                                             HYS01540
      HRLN=3.12                                                         HYS01550
      A=PY                                                              HYS01560
      RETURN                                                            HYS01570
  330 IF (P.GE.FMAX) GO TO 200                                          HYS01580
      S=KC                                                              HYS01590
      NRL=3                                                             HYS01600
      HRLN=3.2                                                          HYS01610
      A=FX                                                              HYS01620
      RETURN                                                            HYS01630
C                                                                       HYS01640
C RULE 4 - LOADING TOWARDS Y' FROM FX.                                  HYS01650
C                                                                       HYS01660
  400 IF (IDR) 500,500,410                                              HYS01670
  410 IF (ABS(P).GE.PY) GO TO 100                                       HYS01680
      S=KT1                                                             HYS01690
      NRL=4                                                             HYS01700
      HRLN=4.1                                                          HYS01710
      A=-PY                                                             HYS01720
      RETURN                                                            HYS01730
C                                                                       HYS01740
C RULE 5 - LOADING AND UNLOADING BETWEEN BRANCHES O-Y AND FX-Y'         HYS01750
C                                                                       HYS01760
  500 IF (KC.NE.KT1) THEN                                               HYS01770
         CALL INTSCT(DITENS,PITENS,D,P,KC,DYT,PY,KT1)                   HYS01780
         CALL INTSCT(DICOMP,PICOMP,D,P,KC,DX ,FX,KT1)                   HYS01790
      ELSE                                                              HYS01800
         PITENS=P                                                       HYS01810
         DITENS=D                                                       HYS01820
         PICOMP=P                                                       HYS01830
         DICOMP=D                                                       HYS01840
      ENDIF                                                             HYS01850
      IF (P.LE.PICOMP) GO TO 330                                        HYS01860
      IF (P.GE.PITENS) GO TO 210                                        HYS01870
      S=KC                                                              HYS01880
      NRL=5                                                             HYS01890
      HRLN=5.1                                                          HYS01900
      IF (PGTPL ) A=PITENS                                              HYS01910
      IF (PLTPL ) A=PICOMP                                              HYS01920
      RETURN                                                            HYS01930
C                                                                       HYS01940
C RULE 6 - LOADING AFTER TENSION YEILD                                  HYS01950
C                                                                       HYS01960
  600 IF (IDR) 630,630,610                                              HYS01970
  610 S=KT2                                                             HYS01980
      NRL=6                                                             HYS01990
      HRLN=6.1                                                          HYS02000
      A=5*P                                                             HYS02010
C.......... CALC UNLOADING CURVE FOR ENERGY INFO                        HYS02020
          FMAX=AMAX1(P,FMAX)                                            HYS02030
          DMAX=AMAX1(D,DMAX)                                            HYS02040
          KR=KC*(DMAX/DYT)**(-1. * ALPHA)                               HYS02050
          FX  =FMAX-PY                                                  HYS02060
          DX  =DMAX-PY/KR                                               HYS02070
          KS  =(PY+FX)/(DX-DYC)                                         HYS02080
C-------- FORCE KR GE KS, TO PREVENT UNSTABLE LOOPS...                  HYS02090
          IF (KS.GT.KR)  KR=(FMAX+PY)/(DMAX-DYC)                        HYS02100
          SEQ=KR                                                        HYS02110
      RETURN                                                            HYS02120
  630 FMAX=AMAX1(P,FMAX)                                                HYS02130
      DMAX=AMAX1(D,DMAX)                                                HYS02140
      KR=KC*(DMAX/DYT)**(-1. * ALPHA)                                   HYS02150
      FX  =FMAX-PY                                                      HYS02160
      DX  =DMAX-PY/KR                                                   HYS02170
      KS  =(PY+FX)/(DX-DYC)                                             HYS02180
C-------- FORCE KR GE KS, TO PREVENT UNSTABLE LOOPS...                  HYS02190
      IF (KS.GT.KR) THEN                                                HYS02200
        KR=(FMAX+PY)/(DMAX-DYC)                                         HYS02210
        KS=KR                                                           HYS02220
        DX=DMAX-PY/KR                                                   HYS02230
      ENDIF                                                             HYS02240
      DP  =DYC + BETA*(DX-DYC)                                          HYS02250
      FP  =FX - (DX-DP)*KS                                              HYS02260
      KT  =(2*PY+FP)/(DP-2*DYC)                                         HYS02270
      DQ  =DYC + PY/KR                                                  HYS02280
      S=KR                                                              HYS02290
      NRL=7                                                             HYS02300
      HRLN=6.2                                                          HYS02310
      A=FX                                                              HYS02320
      RETURN                                                            HYS02330
C                                                                       HYS02340
C RULE 7 - UNLOADING AFTER YEILD                                        HYS02350
C                                                                       HYS02360
  700 SEQ=KR                                                            HYS02370
      IF (IDR) 730,730,710                                              HYS02380
```

```
 710  IF (P.GE.FMAX) GO TO 600                                          HYS02390
      S=KR                                                              HYS02400
      NRL=7                                                             HYS02410
      HRLN=7.1                                                          HYS02420
      A=FMAX                                                            HYS02430
      RETURN                                                            HYS02440
 730  IF (P.LE.FX) GO TO 800                                            HYS02450
      S=KR                                                              HYS02460
      NRL=7                                                             HYS02470
      HRLN=7.2                                                          HYS02480
      A=FX                                                              HYS02490
      RETURN                                                            HYS02500
C                                                                       HYS02510
C RULE 8 - ON BRANCH BETWEEN DX & DP                                    HYS02520
C                                                                       HYS02530
 800  IF (PGTPL) THEN                                                   HYS02540
        IF (KR.EQ.KS) GO TO 1200                                        HYS02550
        GO TO 1300                                                      HYS02560
      ENDIF                                                             HYS02570
      IF (P.LE.FP) GO TO 900                                            HYS02580
      S=KS                                                              HYS02590
      NRL=8                                                             HYS02600
      HRLN=8.1                                                          HYS02610
      A=FP                                                              HYS02620
      RETURN                                                            HYS02630
C                                                                       HYS02640
C RULE 9 - ON BRANCH FP-Y'                                              HYS02650
C                                                                       HYS02660
 900  IF (PGTPL) GO TO 1300                                             HYS02670
      IF (P.LE.(-2*PY) ) GO TO 1000                                     HYS02680
      S=KT                                                              HYS02690
      NRL=9                                                             HYS02700
      HRLN=9.1                                                          HYS02710
      A=-2*PY                                                           HYS02720
      RETURN                                                            HYS02730
C                                                                       HYS02740
C RULE 10 - ON THE COMPRESSION CURVE AFTER YELDING                      HYS02750
C                                                                       HYS02760
 1000 SEQ=  1. / ( 1./KC + (4./KR - 4./KC)*(PY/P)**2 )                 HYS02770
C                                                                       HYS02780
      IF (IDR) 1030,1030,1010                                           HYS02790
 1010 S=KC                                                              HYS02800
      NRL=10                                                            HYS02810
      HRLN=10.1                                                         HYS02820
      A=10*P                                                            HYS02830
      RETURN                                                            HYS02840
 1030 IF (ABS(P).LE.PY) GO TO 1100                                      HYS02850
      S=KC                                                              HYS02860
      NRL=10                                                            HYS02870
      HRLN=10.2                                                         HYS02880
      A=-PY                                                             HYS02890
      RETURN                                                            HYS02900
C                                                                       HYS02910
C RULE 11 - UNLOADING BETWEEN Y' & Q                                    HYS02920
C                                                                       HYS02930
 1100 SEQ=KR                                                            HYS02940
      IF (ABS(P).GT.PY ) GO TO 1000                                     HYS02950
      IF (     P .GE. 0.) GO TO 1200                                    HYS02960
      S=KR                                                              HYS02970
      NRL=11                                                            HYS02980
      HRLN=11.1                                                         HYS02990
      IF (PGTPL) A=0                                                    HYS03000
      IF (PLTPL) A=-PY                                                  HYS03010
      RETURN                                                            HYS03020
C                                                                       HYS03030
C RULE 12 - LOADING FORM Q TO M                                         HYS03040
C                                                                       HYS03050
 1200 IF (PLTPL) THEN                                                   HYS03060
        IF (KS.EQ.KR .AND. P.GT.FP) GO TO 800                           HYS03070
        GO TO 1300                                                      HYS03080
      ENDIF                                                             HYS03090
      IF ( P .GE.FMAX) GO TO 600                                        HYS03100
      S=KS                                                              HYS03110
      NRL=12                                                            HYS03120
      HRLN=12.1                                                         HYS03130
      A=FMAX                                                            HYS03140
      RETURN                                                            HYS03150
C                                                                       HYS03160
C RULE 13 - LOADING AND UNLOADING INSIDE Q-M-D-P-Y'-Y'                  HYS03170
C                                                                       HYS03180
 1300  IF (KR.NE.KS) THEN                                               HYS03190
         CALL INTSCT(DITENS,PITENS,D,P,KR,DMAX,FMAX,KS)                 HYS03200
         CALL INTSCT(DICOMP,PICOMP,D,P,KR,DX ,FX,KS )                   HYS03210
       ELSE                                                             HYS03220
          IF (PGTPL) GO TO 1200                                         HYS03230
          PITENS=P                                                      HYS03240
          DITENS=D                                                      HYS03250
          PICOMP=P                                                      HYS03260
          DICOMP=D                                                      HYS03270
       ENDIF                                                            HYS03280
C                                                                       HYS03290
C--------IS POINT WITHIN PY'-FP-PY' ?                                   HYS03300
       IF (PICOMP.LE.FP)  THEN                                          HYS03310
         IF (KC.NE.KT) THEN                                             HYS03320
            CALL INTSCT(DICOMP,PICOMP,D,P,KC,DP,FP,KT )                 HYS03330
         ELSE                                                           HYS03340
              IF (PLTPL) GO TO  900                                     HYS03350
              PICOMP=P                                                  HYS03360
              DICOMP=D                                                  HYS03370
         ENDIF                                                          HYS03380
         IF (KR.EQ.KS .AND. PLTPL) GO TO  800                           HYS03390
         S2=KT                                                          HYS03400
       ELSE                                                             HYS03410
         S2=KR                                                          HYS03420
       ENDIF                                                            HYS03430
        IF (PGTPL) THEN                                                 HYS03440
           IF (P.GT.PITENS) GO TO 1200                                  HYS03450
           A=PITENS                                                     HYS03460
           HRLN=13.1                                                    HYS03470
        ENDIF                                                           HYS03480
        IF (PLTPL) THEN                                                 HYS03490
           IF (P.LT.PICOMP) GO TO 800                                   HYS03500
           A=PICOMP                                                     HYS03510
           HRLN=13.2                                                    HYS03520
        ENDIF                                                           HYS03530
       S=MAX(KR,S2)                                                     HYS03540
       NRL=13                                                           HYS03550
       RETURN                                                           HYS03560
C                                                                       HYS03570
       END                                                              HYS03580
C*****************************************************************************HYS00010
C          DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                     HYS00020
C          END DEBUG                                                    HYS00030
C*****************************************************************************HYS00040
C**  BENDING HYSTERESIS MODEL..... HYST03 ...............................HYS00050
C*****************************************************************************HYS00060
C                                                                       HYS00070
C     INPUT VARIABLES                                                   HYS00080
C     P       = CURRENT LOAD                                            HYS00090
C     D       = CURRENT DISPLACEMENT                                    HYS00100
C     PL      = LAST LOAD                                               HYS00110
C     VEL     = PRODUCT OF LAST AND CURRENT VELOCITY                    HYS00120
C     NB      = NUMBER OF BACKBONE CURVE POINTS                         HYS00130
C     NI      = MAXIMUN NUMBER OF SECONDARY LOOPS                       HYS00140
C     PC      = CRACKING LOAD                                           HYS00150
C     DC      = CRACKING DISPLACEMENT                                   HYS00160
C     PB(j)   = BACKBONE CURVE LOADING POINT j, j=1,NB                  HYS00170
C     DB(j)   = BACKBONE CURVE DISPLACEMENT POINT j,1,NB                HYS00180
C     SI      = INITIAL STIFFNESS, PC/DC                                HYS00190
C                                                                       HYS00200
C     OUTPUT VARIABLES                                                  HYS00210
C     ....AN INITIAL VALUE OF ZERO , OR FALSE                           HYS00220
C                 IS REQUIRED FOR ALL OUTPUT VARIABLES...               HYS00230
C     K       = STIFFNESS                                               HYS00240
C     RULE    = RULE NUMBER, STORED IN THE FORMAT RN.DIR UPON EXIT OF   HYS00250
C               THIS SU_BROU_TINE                                       HYS00260
C     A       = LOAD LIMIT BEFORE NEXT RULE CHANGE, IN THE CURRENT DIR  HYS00270
C     PMG     = MAXIMUN PAST LOAD IN THE POSITIVE DIRECTION             HYS00280
C     PMH     = MAXIMUN PAST LOAD IN THE NEGATIVE DIRECTION             HYS00290
C     DMG     = MAXIMUN PAST DISPLACEMENT IN THE POSITIVE DIRECTION     HYS00300
C     DMH     = MAXIMUN PAST DISPLACEMENT IN THE NEGATIVE DIRECTION     HYS00310
C     PR(I)   = SECONDARY LOOP LOADING POINT, I                         HYS00320
C     DR(I)   = SECONDARY LOOP DISPLACEMENT POINT, I                    HYS00330
C     FR(I)   = SECONDARY LOOP FLAG                                     HYS00340
C     IR      = NUMBER OF SECONDARY LOOPS ACTIVE                        HYS00350
C     ALPHA1  = POSITIVE LOAD DEGRADING STIFFNESS FACTOR                HYS00360
C     ALPHA2  = NEGATIVE LOAD DEGRADING STIFFNESS FACTOR                HYS00370
C     BACKB   = BACKBONE LOADING FLAG                                   HYS00380
C                                                                       HYS00390
C     INTERNAL VARIABLES                                                HYS00400
C     J       = BACKBONE CURVE POINT # IN THE CURRENT DIRECTION          HYS00410
C     JO      = TEMPORARY VARIABLE                                      HYS00420
C     I,IT    = SECONDARY LOOP NUMBER                                   HYS00430
C     DIR     = CURRENT DIRECTION                                       HYS00440
C               DIR=1, POSITIVE LOADING                                 HYS00450
C               DIR=2, POSITIVE UNLOADING                               HYS00460
C               DIR=3, NEGATIVE LOADING                                 HYS00470
C               DIR=4, NEGATIVE UNLOADING                               HYS00480
C     DIRL    = LAST DIRECTION                                          HYS00490
C     IRULE   = INTEGER VALUE OF RULE #                                 HYS00500
C     LRULE   = INTEGER VALUE OF LAST RULE #                            HYS00510
C     REVRSL  = LOAD REVERSAL FLAG                                      HYS00520
C     PRINT   = PRINT FLAG, USED FOR DEBU_GGING                         HYS00530
C     AP      = ABSOLUTE VALUE OF THE LOAD, P.                          HYS00540
C     PM      = MAXIMUN PAST LOAD IN THE CURRENT DIRECTION              HYS00550
C     DM      = MAXIMUN PAST DISPLACEMENT IN THE CURRENT DIRECTION      HYS00560
C     DMAX    = MAXIMUN PAST DISPLACEMENT IN BOTH DIRECTIONS            HYS00570
C     Q1      = ONE QUARTER OF PM, USED BY UNLOADING CURVES             HYS00580
C     Q3      = THREE QUARTERS OF PM, USED BY UNLOADING CURVES          HYS00590
C     DQ1     = DISPLACEMENT AT Q1 ON UNLOADING BRANCH                  HYS00600
C     DQ3     = DISPLACEMENT AT Q3 ON UNLOADING BRANCH                  HYS00610
C     DQ1     = INTERMEDIATE DISPLACEMENT INTERCEPT OF UNLOADING BRANCH HYS00620
C     DQ2     = INTERMEDIATE DISPLACEMENT INTERCEPT OF UNLOADING BRANCH HYS00630
C     D0      = DISPLACEMENT INTERCEPT OF UNLOADING BRANCH               HYS00640
C     DP      = DIFFERANCE IN LOAD, USED FOR CALCULATING K              HYS00650
C     DD      = DIFFERANCE IN DISPLACEMENT, USED FOR CALCULATING K      HYS00660
C     PX      = LOAD INTERCEPT OF LOADING AND BACKBONE CURVE            HYS00670
C     DX      = DISPLACEMENT INTERCEPT OF LOADING AND BACKBONE CURVE    HYS00680
C     PC3     = PC/3                                                    HYS00690
C     DC3     = DISPLACEMENT AT PC3                                     HYS00700
C     P2,D2   = INTERSECTION OF LOADING AND UNLOADING CURVES            HYS00710
C     PI,DI   = SMALL LOOP UNLOADING POINTS                             HYS00720
C     X       = MAXIMUN DISPLACEMENT                                    HYS00730
C     K8      = STIFFNESS FROM RULE 8                                   HYS00740
C     FLAG1   = UPPER SECONDARY LOOP FLAG                               HYS00750
C     FLAG2   = LOWER SECONDARY LOOP FLAG                               HYS00760
C     ALPHAA  = DEGRADING FACTOR FOR FIRST CYCLE                        HYS00770
C     ALPHAB  = DEGRADING FACTOR FOR SUBSEQUENT CYCLES                  HYS00780
C     S       = TEMPORARY STIFFNESS                                     HYS00790
C     S1      = UNLOADING STIFFNESS FROM FS1                            HYS00800
C     S2      = UNLOADING STIFFNESS FROM FS2                            HYS00810
C     S3      = UNLOADING STIFFNESS FROM FS3                            HYS00820
C     SU      = STIFFNESS BETWEEN CURRENT AND RELOADING POINT           HYS00830
C     SO2     = STIFFNESS BETWEEN PA AND D0                             HYS00840
C     SO3     = STIFFNESS BETWEEN PB AND D0                             HYS00850
C                                                                       HYS00860
C     INTERNAL FUNCTIONS                                                HYS00870
C     FD2     = DISPLACEMENT INTERCEPT OF LOADING AND UNLOADING CURVE   HYS00880
C     FS1     = UNLOADING STIFFNESS WHEN P>.75 PMAX                     HYS00890
C     FS2     = UNLOADING STIFFNESS WHEN .75 PMAX > P > .25 PMAX        HYS00900
C     FS3     = UNLOADING STIFFNESS WHEN .25 PMAX > P                   HYS00910
C     FSL     = LOADING STIFFNESS WHEN .P < PC/3 WITHOUT REVERSAL...    HYS00920
C                                                                       HYS00930
C     SU_BROU_TINES CALLED                                              HYS00940
C     DIRECT  = DETERMINES THE VALUES OF DIR AND DIRL                   HYS00950
C     INTSCT  = DETERMINES THE INTERSECTION OF TWO LINES IN POINT-SLOPE  HYS00960
C               FORM                                                    HYS00970
C     KMIN    = DETERMINES THE STIFFNESS BASED ON DP,DD AND K_DEFULT    HYS00980
C                                                                       HYS00990
      SUBROUTINE HYST03(P,D,PL,DL,VEL,NB,NI,PC,DC,PB,DB,SI,CSE,         HYS01000
     & K,RULE,RDIR,A,PMG,PMH,DMG,DMH,BACKB,ALPHA1,ALPHA2,               HYS01010
     & IR,SR1,SR2,SR3,SRM,PR,DR,FR,PRINT,                              HYS01020
     & SE,DT,ESE,PSE,PSEOLD,UPOS,UNEG,EXCR)                            HYS01030
C                                                                       HYS01040
      IMPLICIT REAL(A-H,K,O-Z)                                          HYS01050
      IMPLICIT INTEGER(I,J,L-N)                                         HYS01060
      INTEGER DIR,DIRL                                                  HYS01070
      CHARACTER*4  FR(1)                                                HYS01080
      LOGICAL  BACKB,REVRSL,PRINT,BTEST                                 HYS01090
      DIMENSION PB(1),DB(1),PR(1),DR(1),UPOS(3),UNEG(3),EXCR(6),ESE(3)  HYS01100
      DATA ALPHAA/1.129/,ALPHAB/1.029/                                  HYS01110
C                                                                       HYS01120
      FS1(X)= SI *(DC/X)**.294                                          HYS01130
      FS2(X)= SI *(0.8344*(DC/X-1) +1)                                  HYS01140
      FS3(X)= SI *(0.9092*(DC/X-1) +1)                                  HYS01150
      FSL(X)= SI *(DC/X)**.285                                          HYS01160
      FD2(X)= DM-.05*PM/(FS1(X))                                        HYS01170
C                                                                       HYS01180
C                                                                       HYS01190
 1000 IF (PRINT) WRITE (6,1010) P,D,PL,DL,NB,NI,PC,DC,SI,K,             HYS01200
     &      RULE,A,PMG,PMH,DMG,DMH,BACKB,IR,DIR,                       HYS01210
     & SE,DT,ESE(1),PSE,PSEOLD,UPOS,CSE,UNEG,EXCR,VEL                  HYS01220
 1010 FORMAT(                                                           HYS01230
     &' B1-    P=',G13.5,'    D=',G13.5,'    PL=',G13.5,'   DL=',G13.5/ HYS01240
     &' B1-   NB=',G13.5,'   NI=',G13.5,'   PC=',G13.5,'   DC=',G13.5/ HYS01250
     &' B1-   SI=',G13.5,'    K=',G13.5,'  RULE=',G13.5,'    A=',G13.5/ HYS01260
     &' B1-  PMG=',G13.5,'  PMH=',G13.5,'  DMG=',G13.5,'  DMH=',G13.5/ HYS01270
     &' B1-BACKB=',G13.5,'   IR=',G13.5,'  DIR=',G13.5,'   SE=',G13.5/ HYS01280
     &' B1-   DT=',G13.5,'  ESE=',G13.5,'  PSE=',G13.5,' PSEO=',G13.5/ HYS01290
     &' B1-  UP1=',G13.5,'  UP2=',G13.5,'  UP3=',G13.5,'  CSE=',G13.5/ HYS01300
     &' B1-  UN1=',G13.5,'  UN2=',G13.5,'  UN3=',G13.5/                 HYS01310
     &' B1-  EX1=',G13.5,'  EX2=',G13.5,'  EX3=',G13.5/                 HYS01320
     &' B1-  EX4=',G13.5,'  EX5=',G13.5,'  EX6=',G13.5/'  VEL=',G13.5)  HYS01330
       IF (IR.GT.0.AND.PRINT) WRITE (6,1020) (I,PR(I),DR(I),FR(I),I=1,IR)HYS01340
 1020 FORMAT                                                            HYS01350
     & (' B1-    I=',G13.5,'    PR=',G13.5,'    DR=',G13.5,'    FR=',A )HYS01360
C                                                                       HYS01370
      AP=ABS(P)                                                         HYS01380
C..... DETERMINE IF ELASTIC                                             HYS01390
      IF (RULE.EQ.0 .AND. AP.LT.PC ) THEN                               HYS01400
            K=SI                                                        HYS01410
            SE=K                                                        HYS01420
            RETURN                                                      HYS01430
      ENDIF                                                             HYS01440
C                                                                       HYS01450
C..... DETERMINE IF WALL IS INACTIVE, IF SO RETURN                      HYS01460
      IF (RULE.LT.0.0) RETURN                                           HYS01470
C                                                                       HYS01480
C..... DETERMINE CURRENT DIRECTION                                      HYS01490
      LRULE=INT(RULE)                                                   HYS01500
      DIR =RDIR                                                         HYS01510
      CALL DIRECT(DIR,DIRL,P,PL,VEL)                                    HYS01520
      RDIR =DIR                                                         HYS01530
      IF (PRINT) WRITE (6,1030) DIR,DIRL                                HYS01540
 1030 FORMAT                                                            HYS01550
     & (' B1-  DIR=',I8  ,'  DIRL=',I8)                                 HYS01560
C                                                                       HYS01570
C..... DETERMINE MAXIMUN AND MINIMUN VALUES                             HYS01580
            PMG=MAX(P,PMG, PC)                                          HYS01590
            DMG=MAX(D,DMG, DC)                                          HYS01600
            PMH=MIN(P,PMH,-PC)                                          HYS01610
            DMH=MIN(D,DMH,-DC)                                          HYS01620
            DMAX=MAX(DMG,-DMH)                                          HYS01630
            PMAX=MAX(PMG,-PMH)                                          HYS01640
```

```
C                                                                       HYS01650
C..... SET MAXIMUN AND MINIMUN VALUES...                                HYS01660
      IF (DIR.EQ.1 .OR. DIR.EQ.2 ) THEN                                 HYS01670
            PM=PMG                                                      HYS01680
            DM=DMG                                                      HYS01690
      ELSE IF (DIR.EQ.3 .OR. DIR.EQ.4) THEN                             HYS01700
            PM=PMH                                                      HYS01710
            DM=DMH                                                      HYS01720
      ENDIF                                                             HYS01730
C                                                                       HYS01740
C..... CALCULATE EQUIVALENT UNLOADING STIFFNESS FOR ENERGY...           HYS01750
        IF (BACKB .OR. RULE.EQ.1.0) THEN                                HYS01760
            S1=FS1(DMAX)                                                HYS01770
            S2=FS2(DMAX)                                                HYS01780
            S3=FS3(DMAX)                                                HYS01790
            DQ =DM - PM*(DMH-DMG)/(PMH-PMG)                             HYS01800
            Q1=PM/4                                                     HYS01810
            Q3=3*Q1                                                     HYS01820
            DQ3=DM -0.25*PM/S1                                          HYS01830
            S02=KMIN((DQ3-D0),Q3,S1)                                    HYS01840
            DQ1=DQ3-0.50*PM/MAX(S2,S02)                                 HYS01850
            S0J=KMIN((DQ1-D0),Q1,S1)                                    HYS01860
            SE= 0.5 * PM**2 / (                                         HYS01870
     &       (PM+Q3)*0.5*(DM-DQ3)                                       HYS01880
     &      +(Q3+Q1)*0.5*(DQ3-DQ1)                                      HYS01890
     &      +(   Q1)*0.5*(   Q1)/MAX(S3,S03) )                          HYS01900
        ENDIF                                                           HYS01910
C                                                                       HYS01920
C..... CHECK TO SEE IF LAST RULE IS STILL IN EFFECT                     HYS01930
      IF (DIR.EQ.DIRL .AND.                                             HYS01940
     &  ((DIR.EQ.1 .AND. P.LT.A) .OR. (DIR.EQ.2 .AND. P.GT.A)  .OR.     HYS01950
     &   (DIR.EQ.3 .AND. P.GT.A) .OR. (DIR.EQ.4 .AND. P.LT.A))) THEN    HYS01960
        RETURN                                                          HYS01970
      ENDIF                                                             HYS01980
C                                                                       HYS01990
C..... CALC EXCURSION RATIO AT EVERY ZERO CROSSING...                   HYS02000
      IF (P*PL.LE.0) CALL DNE(0,UPOS,EXCR, DY,DLI,ESE,PSE,PSEOLD,CSE)   HYS02010
C                                                                       HYS02020
C..... ERASE SMALL LOOP FLAGS                                           HYS02030
      IF (IR.GT.0) THEN                                                 HYS02040
        IT=IR                                                           HYS02050
        DO 10 I=1,IT                                                    HYS02060
          IF ((FR(I).EQ.'L' .AND. ( P.LE.PR(I) .OR. D.LE.DR(I) ))  .OR. HYS02070
     &       (FR(I).EQ.'7' .AND. ( P.GE.PR(I) .OR. D.GE.DR(I) ))) THEN HYS02080
            IR=I-1                                                      HYS02090
            GO TO 15                                                    HYS02100
          ENDIF                                                         HYS02110
   10   CONTINUE                                                        HYS02120
      ENDIF                                                             HYS02130
   15 I=IR+1                                                            HYS02140
C                                                                       HYS02150
      IF (DIR.EQ.1 .OR. DIR.EQ.3 )  THEN                                HYS02160
C..... LOADING RULES ..................................... RULES 1,6-11HYS02170
C                                                                       HYS02180
C..... LOADING ON BACKBONE                                      RULE 1HYS02190
    1     IF (BACKB .OR. RULE.EQ.0.0) THEN                              HYS02200
            K=0.                                                        HYS02210
            DO 18 J=2,NB                                                HYS02220
              DD=DB(J)-ABS(D)                                           HYS02230
              DP=PB(J)-AP                                               HYS02240
              IF (AP.LT.PB(J) .AND. (DD*DP).GT.0.) THEN                 HYS02250
                S=DP/DD                                                 HYS02260
                IF (S.GT.K) THEN                                        HYS02270
                  K=S                                                   HYS02280
                  A=SIGN(PB(J),P)                                       HYS02290
                ENDIF                                                   HYS02300
              ENDIF                                                     HYS02310
   18       CONTINUE                                                    HYS02320
            RULE=1                                                      HYS02330
            BACKB=.TRUE.                                                HYS02340
            IR=0                                                        HYS02350
            IF (DIR.EQ.1) THEN                                          HYS02360
              ALPHA1=ALPHAA                                             HYS02370
            ELSE                                                        HYS02380
              ALPHA2=ALPHAA                                             HYS02390
            ENDIF                                                       HYS02400
C                                                                       HYS02410
C..... LOADING AFTER UNLOADING                                          HYS02420
          ELSE                                                          HYS02430
C..............  RELOADING INSIDE SMALL LOOPS                    RULE 11HYS02440
   20       I=I-1                                                       HYS02450
            IF (I.GT.0 ) THEN                                           HYS02460
              IF (.NOT.(( FR(I).EQ.'7' .AND. DIR.EQ.1 )  .OR.           HYS02470
     &                 ( FR(I).EQ.'L' .AND. DIR.EQ.3 )))  GO TO 20HYS02480
                K=KMIN((D-DR(I)),(P-PR(I)),SI)                          HYS02490
                A=PR(I)                                                 HYS02500
                RULE=11+I/1000.                                         HYS02510
C                                                                       HYS02520
C.............. INITIAL RELOADING BELOW PC/3                           HYS02530
C..............................  WITH REVERSAL                  RULE 8HYS02540
            ELSE IF (AP.LT.(PC/3) )THEN                                 HYS02550
              BACKB=.FALSE.                                             HYS02560
              IF ((DIR.EQ.1 .AND. DIRL.EQ.4) .OR.                       HYS02570
     &            (DIR.EQ.3 .AND. DIRL.EQ.2)) THEN                      HYS02580
                P2=.95*PM                                               HYS02590
                D2=FD2(DMAX)                                            HYS02600
                K=KMIN((D-D2),(P-P2),SI)                                HYS02610
                KB=K                                                    HYS02620
                A= P2                                                   HYS02630
                RULE=8                                                  HYS02640
C                                                                       HYS02650
                PC3=SIGN(PC/3,PM)                                       HYS02660
                D01=DM-PM*(1./FS1(DMAX)+2./FS2(DMAX)+1./FS3(DMAX))/4.HYS02670
                D02=DM - PM*(DMH-DMG)/(PMH-PMG)                         HYS02680
                IF (DIR.EQ.1) D0=MAX(D01,D02)                           HYS02690
                IF (DIR.EQ.3) D0=MIN(D01,D02)                           HYS02700
                DC3=D0+PC3/FSL(DMAX)                                    HYS02710
                S=KMIN((D-DC3),(P-PC3),SI)                              HYS02720
                IF (S.GT.K) THEN                                        HYS02730
                    K=S                                                 HYS02740
                    A= PC3                                              HYS02750
                    RULE=8.1                                            HYS02760
                ENDIF                                                   HYS02770
C                                                                       HYS02780
C.............................. WITH REVERSAL, FAR SIDE LOADING    RULE 9HYS02790
              IF ((DIR.EQ.1 .AND. D.LT.0) .OR.                          HYS02800
     &            (DIR.EQ.3 .AND. D.GT.0)) THEN                         HYS02810
                K=K0                                                    HYS02820
                DO 40 J=2,NB                                            HYS02830
                  IF ( PB(J) .LT.ABS(PM) ) GO TO 40                     HYS02840
                  DD= -SIGN(DB(J),D)-D                                  HYS02850
                  DP= -SIGN(PB(J),D)-P                                  HYS02860
                  IF ( (DD*DP).LE.0.) GO TO 40                          HYS02870
                    S=DP/DD                                             HYS02880
                    IF (S.GT.K .AND. S.LE.SI) THEN                      HYS02890
                      K=S                                               HYS02900
                      A=SIGN(PB(J),PM)                                  HYS02910
                      RULE=9                                            HYS02920
                      BACKB=.TRUE.                                      HYS02930
                    ENDIF                                               HYS02940
   40             CONTINUE                                             HYS02950
              ENDIF                                                     HYS02960
C                                                                       HYS02970
C.............................. WITHOUT REVERSAL               RULE 6HYS02980
              ELSE                                                      HYS02990
                K=FSL(DMAX)                                             HYS03000
                A= SIGN((PC/3),PM)                                      HYS03010
                RULE=6                                                  HYS03020
              ENDIF                                                     HYS03030
C                                                                       HYS03040
C.............. MIDRANGE RELOADING CURVE                       RULE 7HYS03050
              ELSE IF (AP.LT.ABS(.95*PM) ) THEN                         HYS03060
                BACKB=.FALSE.                                           HYS03070
                P2=.95*PM                                               HYS03080
                D2=FD2(DMAX)                                            HYS03090
                K=KMIN((D-D2),(P-P2),SI)                                HYS03100
                A=P2                                                    HYS03110
                RULE=7                                                  HYS03120
C                                                                       HYS03130
C.............. LOADING TOWARDS THE BACKBONE CURVE              RULE 10HYS03140
              ELSE                                                      HYS03150
                BACKB=.TRUE.                                            HYS03160
                IF (AP.GE.ABS(PM) .OR. (ABS(DM)).LE.DC) GO TO 1         HYS03170
                IR=0                                                    HYS03180
                IF (DIR.EQ.1) THEN                                      HYS03190
                    IF (ALPHA1.NE.ALPHAA .AND. ALPHA1.NE.ALPHAB)        HYS03200
     &                          ALPHA1=ALPHAA                           HYS03210
                    DD=ALPHA1*DM-D                                      HYS03220
                ELSE                                                    HYS03230
                    IF (ALPHA2.NE.ALPHAA .AND. ALPHA2.NE.ALPHAB)        HYS03240
     &                          ALPHA2=ALPHAA                           HYS03250
                    DD=ALPHA2*DM-D                                      HYS03260
                ENDIF                                                   HYS03270
                DP=PM-P                                                 HYS03280
                IF ((DP*DD).GT.0) THEN                                  HYS03290
                    K=MIN( (DP/DD),SI)                                  HYS03300
                ELSE                                                    HYS03310
                    K=SI                                                HYS03320
                ENDIF                                                   HYS03330
                DO 50 J=2,NB                                            HYS03340
                  IF ( PB(J).LE.ABS(PM) ) GO TO 50                      HYS03350
                    J0=J-1                                              HYS03360
                    DD=DB(J)-DB(J0)                                     HYS03370
                    DP=PB(J)-PB(J0)                                     HYS03380
                    IF ((DP*DD).LE.0.) GO TO 50                         HYS03390
                     CALL INTSCT(DX,PX,D,P,K,SIGN(DB(J),P),             HYS03400
     &                           SIGN(PB(J),P),(DP/DD) )                HYS03410
                     PX=PX                                              HYS03420
                     DX=DX                                              HYS03430
                     IF (ABS(DX).GE.DB(J0) .AND. ABS(DX).LE.DB(J).AND.HYS03440
     &                  ABS(PX).GT.ABS(PM)  .AND.  PM*PX.GT.0) THEN     HYS03450
                        IF (AP.GT.ABS(PX)) GO TO 1                      HYS03460
                        A= PX                                           HYS03470
                        RULE=10                                         HYS03480
                        IF (DIR.EQ.1)  THEN                             HYS03490
                             ALPHA1=ALPHAB                              HYS03500
                        ELSE                                            HYS03510
                             ALPHA2=ALPHAB                              HYS03520
                        ENDIF                                           HYS03530
                        GO TO 100                                       HYS03540
                      ENDIF                                             HYS03550
   50               CONTINUE                                            HYS03560
                    GO TO 1                                             HYS03570
                ENDIF                                                   HYS03580
            ENDIF                                                       HYS03590
C                                                                       HYS03600
C                                                                       HYS03610
      ELSE                                                              HYS03620
C..... UNLOADING RULES ..................................... RULES 2-5HYS03630
C                                                                       HYS03640
C..... INITILIZE VALUES                                                 HYS03650
            BACKB=.FALSE.                                               HYS03660
            S1=FS1(DMAX)                                                HYS03670
            S2=FS2(DMAX)                                                HYS03680
            S3=FS3(DMAX)                                                HYS03690
            DQ =DM - PM*(DMH-DMG)/(PMH-PMG)                             HYS03700
            Q1=PM/4                                                     HYS03710
            Q3=3*Q1                                                     HYS03720
            DQ3=DM -0.25*PM/S1                                          HYS03730
            S02=KMIN((DQ3-D0),Q3,SI)                                    HYS03740
            DQ1=DQ3-0.50*PM/MAX(S2,S02)                                 HYS03750
            S03=KMIN((DQ1-D0),Q1,SI)                                    HYS03760
            SU=0.                                                       HYS03770
C                                                                       HYS03780
C                                                                       HYS03790
C..............  UNLOADING INSIDE SMALL POSITIVE LOOPS           RULE 5 HYS03800
            I=IR+1                                                      HYS03810
   70       I=I-1                                                       HYS03820
            IF (I.GT.0) THEN                                            HYS03830
              IF (.NOT.((DIR.EQ.2 .AND. FR(I).EQ.'L')    .OR.           HYS03840
     &                  (DIR.EQ.4 .AND. FR(I).EQ.'7')) ) GO TO 70     HYS03850
              IF ((DIR.EQ.2 .AND.( (P.GT.Q3 .AND. PR(I).GT.Q3) .OR.     HYS03860
     &             (Q3.GE.P .AND.   P.GT.Q1 .AND. PR(I).GT.Q1) .OR.     HYS03870
     &             (Q1.GE.P .AND.   PR(I).GT.0 ))) .OR.                 HYS03880
     &            (DIR.EQ.4 .AND.( (P.LT.Q3 .AND. PR(I).LT.Q3) .OR.     HYS03890
     &             (Q3.LE.P .AND.   P.LT.Q1 .AND. PR(I).LT.Q1) .OR.     HYS03900
     &             (Q1.LE.P .AND.   PR(I).LT.0 ))) ) THEN               HYS03910
                K=(PR(I)-P)/(DR(I)-D)                                   HYS03920
                IF (K.LE.0 .OR. K.GT.SI) GO TO 70                       HYS03930
                A=PR(I)                                                 HYS03940
                RULE=5+I/1000.                                          HYS03950
                GO TO 100                                               HYS03960
              ELSE                                                      HYS03970
                  SU=KMIN((DR(I)-D),(PR(I)-P),SI)                       HYS03980
              ENDIF                                                     HYS03990
            ENDIF                                                       HYS04000
C                                                                       HYS04010
C..............  CALC DUCTILITY AND EXCURSION RATIO...                 HYS04020
            IF  (RULE.EQ.1 .OR. RULE.GT.5)  THEN                        HYS04030
              IF (DIR.EQ.2) THEN                                        HYS04040
                DLI= MAX( D , DL )                                      HYS04050
                CALL DNE(1,UPOS,EXCR, DY,DLI,ESE(2),PSE,PSEOLD,CSE)     HYS04060
              ELSE IF (DIR.EQ.4) THEN                                   HYS04070
                DLI= MIN( D , DL )                                      HYS04080
                CALL DNE(1,UNEG,EXCR,-DY,DLI,ESE(3),PSE,PSEOLD,CSE)     HYS04090
              ENDIF                                                     HYS04100
            ENDIF                                                       HYS04110
C                                                                       HYS04120
C..............  UNLOADING ON THE TOP SEGMENT                   RULE 2HYS04130
            IF(AP.GT.ABS(Q3)) THEN                                      HYS04140
              K=MAX(S1,SU)                                              HYS04150
              A=Q3                                                      HYS04160
              RULE=2                                                    HYS04170
C..............  UNLOADING ON THE MIDDLE SEGMENT                RULE 3HYS04180
            ELSE IF(AP.GT.ABS(Q1)) THEN                                 HYS04190
              K=MAX(S2,S02,SU)                                          HYS04200
              A=Q1                                                      HYS04210
              RULE=3                                                    HYS04220
C..............  UNLOADING ON THE BOTTOM SEGMENT                RULE 4HYS04230
            ELSE                                                        HYS04240
              K=MAX(S3,S03,SU)                                          HYS04250
              A=0.                                                      HYS04260
              RULE=4                                                    HYS04270
            ENDIF                                                       HYS04280
      ENDIF                                                             HYS04290
C..... SAVE REVERSAL POINTS FOR SMALL LOOPS.....                        HYS04300
  100 IRULE=INT(RULE)                                                   HYS04310
      REVRSL=IRULE.EQ.1.OR.IRULE.EQ.10.OR.LRULE.EQ.1.OR.LRULE.EQ.10     HYS04320
      IF      ((DIRL.EQ.1 .AND. DIR.EQ.2 .AND. .NOT. REVRSL)            HYS04330
     &          .OR. (DIRL.EQ.4 .AND. DIR.EQ.3) ) THEN                  HYS04340
        IF (IR.LT.NI)  THEN                                             HYS04350
            IR=IR+1                                                     HYS04360
            PR(IR)=P                                                    HYS04370
            DR(IR)=D                                                    HYS04380
            FR(IR)='7'                                                  HYS04390
        ENDIF                                                           HYS04400
      ELSE IF ((DIRL.EQ.2 .AND. DIR.EQ.4 .AND. .NOT. REVRSL)            HYS04410
     &          .OR. (DIRL.EQ.2 .AND. DIR.EQ.1) ) THEN                  HYS04420
        IF (IR.LT.NI)  THEN                                             HYS04430
            IR=IR+1                                                     HYS04440
            PR(IR)=P                                                    HYS04450
            DR(IR)=D                                                    HYS04460
            FR(IR)='L'                                                  HYS04470
        ENDIF                                                           HYS04480
```

```
      ENDIF                                                             HYS04490
      IF (PRINT) WRITE (6,1040) S1,S2,S3                                HYS04500
 1040 FORMAT(                                                           HYS04510
     &' S1=   S1=',G13.5,'   S2=',G13.5,'    S3=',G13.5,'     =',G13.5) HYS04520
                                                                        HYS04530
      IF ((1.05*SI).GT.K .AND. K.GT.0 ) RETURN                          HYS04540
C                                                                       HYS04550
      K=0                                                               HYS04560
      RULE=-RULE                                                        HYS04570
      RETURN                                                            HYS04580
C                                                                       HYS04590
      END                                                               HYS04600
C=======================================================================HYS00010
C=== SHEAR HYSTERESIS MODEL..... HYST04 .............................. HYS00020
C=======================================================================HYS00030
C     DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                          HYS00040
C     END DEBUG                                                         HYS00050
C=======================================================================HYS00060
C                                                                       HYS00070
C     INPUT VARIABLES                                                   HYS00080
C     P       = CURRENT LOAD                                            HYS00090
C     D       = CURRENT DISPLACEMENT                                    HYS00100
C     PL      = LAST LOAD                                               HYS00110
C     VEL     = PRODUCT OF LAST AND CURRENT VELOCITY                    HYS00120
C     NS      = NUMBER OF BACKBONE CURVE POINTS                         HYS00130
C     NI      = MAXIMUN NUMBER OF SECONDARY LOOPS                       HYS00140
C     PC      = CRACKING LOAD                                           HYS00150
C     DC      = CRACKING DISPLACEMENT                                   HYS00160
C     SI      = INITIAL STIFFNESS PC/DC                                 HYS00170
C     PS      = BACKBONE CURVE LOADING POINTS                           HYS00180
C     DS      = BACKBONE CURVE DISPLACEMENT POINTS                      HYS00190
C                                                                       HYS00200
C     OUTPUT VARIABLES                                                  HYS00210
C      AN INITIAL VALUE OF ZERO IS REQUIRED FOR ALL OUTPUT VARIABLES... HYS00220
C     K       = STIFFNESS                                               HYS00230
C     RULE    = RULE NUMBER, STORED IN THE FOR_MAT RN.DIR UPON EXIT OF  HYS00240
C               THIS SU_BROU_TINE                                       HYS00250
C     A       = LOAD LIMIT BEFORE NEXT RULE CHANGE                      HYS00260
C     PMG     = MAXIMUN PAST LOAD IN THE POSITIVE DIRECTION             HYS00270
C     PMH     = MAXIMUN PAST LOAD IN THE NEGATIVE DIRECTION             HYS00280
C     DMG     = MAXIMUN PAST DISPLACEMENT IN THE POSITIVE DIRECTION     HYS00290
C     DMH     = MAXIMUN PAST DISPLACEMENT IN THE NEGATIVE DIRECTION     HYS00300
C     PR      = SECONDARY LOOP LOADING POINT                            HYS00310
C     DR      = SECONDARY LOOP DISPLACEMENT POINT                       HYS00320
C     FR      = SECONDARY LOOP FLAG                                     HYS00330
C     IR      = SECONDARY LOOP NUMBER                                   HYS00340
C     BACKB   = BACKBONE LOADING FLAG                                   HYS00350
C     SR1     = RELOADING STIFFNESS FROM RULE 8 - USED BY RULE 11       HYS00360
C     SR2     = RELOADING STIFFNESS FROM RULE 9 - USED BY RULE 11       HYS00370
C     SR3     = RELOADING STIFFNESS FROM RULE 7 - USED BY RULE 11       HYS00380
C     SRM     = RELOADING STIFFNESS FROM RULES 8 & 9 - USED BY RULE 11  HYS00390
C                                                                       HYS00400
C     INTERNAL VARIABLES                                                HYS00410
C     J       = BACKBONE CURVE POINT # IN THE CURRENT DIRECTION          HYS00420
C     DIR     = CURRENT DIRECTION                                       HYS00430
C               DIR=1, POSITIVE LOADING                                 HYS00440
C               DIR=2, POSITIVE UNLOADING                               HYS00450
C               DIR=3, NEGATIVE LOADING                                 HYS00460
C               DIR=4, NEGATIVE UNLOADING                               HYS00470
C     DIRL    = LAST DIRECTION                                          HYS00480
C     IRULE   = INTERGER VALUE OF RULE                                  HYS00490
C     LRULE   = INTEGER VALUE OF LAST RULE #                            HYS00500
C     CRACKD  = CRACKED WALL FLAG                                       HYS00510
C     REVRSL  = LOAD REVERSAL FLAG                                      HYS00520
C     PRINT   = PRINT FLAG, USED FOR DEBU_GGING                         HYS00530
C     PC2     = PC/2, USED BY LOADING CURVES                            HYS00540
C     DC2     = DISPLACEMENT CORRESPONDING TO PC2 ON THE LOADING BRANCH  HYS00550
C     S       = INTERMEDIATE STIFFNESS                                  HYS00560
C     SR      = RELOADING STIFFNESS                                     HYS00570
C     SRP     = MINIMUN RELOADING STIFFNESS                             HYS00580
C     DO      = DISPLACEMENT INTERSEPT OF PEAK DISPLACEMENTS            HYS00590
C     DOP     = DISPLACEMENT INTERSEPT OF THE UNLOADING BRANCH          HYS00600
C     AP      = ABSOLUTE VALUE OF CURRENT LOAD P.                       HYS00610
C     PM      = MAXIMUN PAST LOAD IN THE CURRENT DIRECTION              HYS00620
C     DM      = MAXIMUN PAST DISPLACEMENT IN THE CURRENT DIRECTION      HYS00630
C     DMAX    = MAXIMUN PAST DISPLACEMENT IN EITHER DIRECTION           HYS00640
C     PMAX    = MAXIMUN PAST LOAD IN EITHER DIRECTION                   HYS00650
C     PA      = PM-PC, USED BY UNLOADING CURVES                         HYS00660
C     PB      = PC/2, USED BY UNLOADING CURVES                          HYS00670
C     DA      = DISPLACEMENT AT PA, USED BY UNLOADING CURVES            HYS00680
C     DB      = DISPLACEMENT AT PB, USED BY UNLOADING CURVES            HYS00690
C     DP      = DIFFERANCE IN LOAD, USED FOR CALCULATING K              HYS00700
C     DD      = DIFFERANCE IN DISPLACEMENT, USED FOR CALCULATING K      HYS00710
C     PX      = LOAD INTERCEPT OF LOADING AND BACKBONE CURVE            HYS00720
C     DX      = DISPLACEMENT INTERCEPT OF LOADING AND BACKBONE CURVE    HYS00730
C     P2,D2   = POINT ON THE LOADING CURVE BETWEEN RULES 7 AND 10       HYS00740
C     X       = MAXIMUN DISPLACEMENT                                    HYS00750
C               = DISPLACEMENT INTERCEPT FOR RULE 11                    HYS00760
C     X1      = 75% PC DISPLACEMNT  FOR RULE 11                         HYS00770
C     X2      = 25% PC DISPLACEMNT  FOR RULE 11                         HYS00780
C     I,IT    = TEMPORARY LABEL FOR SMALL LOOP LOADING AND UNLOADING PTSHYS00790
C     JO      = J-1                                                     HYS00800
C     ALPHA   = DEGRADING STIFFNESS FACTOR                              HYS00810
C     ALPHAP  = DEGRADING STIFFNESS FACTOR USED BY RULE 10              HYS00820
C     S1      = STIFFNESS FROM FUNCTION FS1                             HYS00830
C     S2      = STIFFNESS FROM FUNCTION FS2                             HYS00840
C     S3      = STIFFNESS FROM FUNCTION FS3                             HYS00850
C     SU      = STIFFNESS BETWEEN CURRENT AND RELOADING POINT           HYS00860
C     SO2     = STIFFNESS BETWEEN PA AND DO                             HYS00870
C     SO3     = STIFFNESS BETWEEN PB AND DO                             HYS00880
C                                                                       HYS00890
C     INTERNAL FUNCTIONS                                                HYS00900
C     FD2     = DISPLACEMENT INTERCEPT OF LOADING AND UNLOADING CURVE    HYS00910
C     FS1     = UNLOADING STIFFNESS WHEN P>(PM-PC)=PA                   HYS00920
C     FS2     = UNLOADING STIFFNESS WHEN PA>P>(PC/2)=PB                 HYS00930
C     FS3     = UNLOADING STIFFNESS WHEN P>PB                           HYS00940
C     FSR     = LOADING BELOW PC/2, WITH PINCHING                       HYS00950
C                                                                       HYS00960
C     SU_BROU_TINES CALLED                                              HYS00970
C     DIRECT  = DETERMINES THE VALUES OF DIR AND DIRL                   HYS00980
C     INTSCT  = DETERMINES THE INTERSECTION OF TWO LINES IN POINT-SLOPE  HYS00990
C               FORM                                                    HYS01000
C     KMIN    = DETERMINES THE STIFFNESS BASED ON DP,DD AND K_DEFULT     HYS01010
C                                                                       HYS01020
C                                                                       HYS01030
      SUBROUTINE HYST04(P,D,PL,DL,VEL,NS,NI,PRINT,                      HYS01040
     &    PC    ,DC    ,PS    ,DS    ,SI    ,CSE   ,DY    ,            HYS01050
     &    K     ,RULE  ,RDIR  ,A     ,PMG   ,PMH   ,                   HYS01060
     &    DMG   ,DMH   ,BACKB ,SR1   ,SR2   ,SR3   ,                   HYS01070
     &    SRM   ,IR    ,SE    ,UPOS  ,UNEG  ,EXCR  ,                   HYS01080
     &    ESE  ,PSE  ,PSEOLD,PR    ,DR    ,FR    )                    HYS01090
C                                                                       HYS01100
      IMPLICIT REAL(A-H,K,O-Z)                                          HYS01110
      IMPLICIT INTEGER(I,J,L-N)                                         HYS01120
      INTEGER DIR,DIRL                                                  HYS01130
      LOGICAL  BACKB,CRACKD,REVRSL,PRINT,BTEST                          HYS01140
      CHARACTER*4  FR(NI)                                               HYS01150
      DIMENSION PS(NS),DS(NS),PR(NI),DR(NI)                             HYS01160
      DIMENSION UPOS(3),UNEG(3),EXCR(6),ESE(3)                          HYS01170
      DATA ALPHA/1.04/                                                  HYS01180
C                                                                       HYS01190
C SYMMETRIC COIEFFICIENTS                                               HYS01200
CC    FS1(X)= SI *MIN((1.03  *(DC/X)**.388 ),1.)                        HYS01210
CC    FS2(X)= SI *MIN((1.036 *(DC/X)**.458 ),1.)                        HYS01220
CC    FS3(X)= SI *MIN((0.789 *(DC/X)**.692 ),1.)                        HYS01230
C UNSYMMETRIC COIEFFICIENTS                                             HYS01240
            FS1(X)= SI *MIN((1.4675*(DC/X)**.343 ),1.)                  HYS01250
            FS2(X)= SI *MIN((0.7761*(DC/X)**.3195),1.)                  HYS01260
            FS3(X)= SI *MIN((0.0707+(DC/X)*1.369 ),1.)                  HYS01270
      FSR(X)= SI *MIN((   (ABS(DC/X))**1.02),1.)                        HYS01280
      FD2(X)= SIGN(DMAX,PM)-.05*SIGN(PMAX,PM)/(FS1(X))                  HYS01290
C                                                                       HYS01300
 1000 IF (PRINT) WRITE (6,1010) P,D,PL,DL,NS,NI,PC,DC,SI,K,RULE,A,      HYS01310
     &     PMG,PMH,DMG,DMH,SR1,SR2,SR3,SRM,BACKB,IR,PSEOLD,VEL,        HYS01320
     &     SE,DY,ESE(1),PSE,UPOS,CSE,UNEG,EXCR                         HYS01330
 1010 FORMAT (                                                          HYS01340
     &' S1-    P=',G13.5,'    D=',G13.5,'    PL=',G13.5,'    DL=',G13.5/ HYS01350
     &' S1-   NS=',G13.5,'   NI=',G13.5,'   PC=',G13.5,'   DC=',G13.5/ HYS01360
     &' S1-   SI=',G13.5,'    K=',G13.5,' RULE=',G13.5,'    A=',G13.5/ HYS01370
     &' S1-  PMG=',G13.5,' PMH=',G13.5,'  DMG=',G13.5,'  DMH=',G13.5/  HYS01380
     &' S1-  SR1=',G13.5,' SR2=',G13.5,'  SR3=',G13.5,'  SRM=',G13.5/  HYS01390
     &' S1-BACKB=',G13.5,'  IR=',G13.5,' PSEO=',G13.5,'  VEL=',G13.5/  HYS01400
     &' S1-   SE=',G13.5,'  DY=',G13.5,'  ESE=',G13.5,'  PSE=',G13.5/  HYS01410
     &' S1-  UP1=',G13.5,' UP2=',G13.5,'  UP3=',G13.5,'  CSE=',G13.5/  HYS01420
     &' S1-  UN1=',G13.5,' UN2=',G13.5,'  UN3=',G13.5/                 HYS01430
     &' S1-  EX1=',G13.5,' EX2=',G13.5,'  EX3=',G13.5/                 HYS01440
     &' S1-  EX4=',G13.5,' EX5=',G13.5,'  EX6=',G13.5)                 HYS01450
      IF (IR.GT.0.AND.PRINT) WRITE (6,1020) (I,PR(I),DR(I),FR(I),I=1,IR)HYS01460
 1020 FORMAT                                                            HYS01470
     & (' I=',G13.5,'    PR=',G13.5,'    DR=',G13.5,'    FR=',A )      HYS01480
C                                                                       HYS01490
      AP=ABS(P)                                                         HYS01500
C                                                                       HYS01510
C..... DETERMINE IF ELASTIC                                             HYS01520
      IF (RULE.EQ.0 .AND. AP .LT.PC ) THEN                              HYS01530
            K=SI                                                        HYS01540
            SE=K                                                        HYS01550
            GO TO 9999                                                  HYS01560
      ENDIF                                                             HYS01570
C                                                                       HYS01580
C..... DETERMINE IF WALL IS INACTIVE, IF SO RE_TURN                     HYS01590
      IF (RULE.LT.0.0) GO TO 9999                                       HYS01600
C                                                                       HYS01610
C..... DETERMINE CURRENT DIRECTION                                      HYS01620
      LRULE=INT(RULE)                                                   HYS01630
      DIR =RDIR                                                         HYS01640
      CALL DIRECT(DIR,DIRL,P,PL,VEL)                                    HYS01650
      RDIR =DIR                                                         HYS01660
      IF (PRINT) WRITE (6,1030) DIR,DIRL                                HYS01670
 1030 FORMAT('   DIR=',I8  ,' DIRL=',I8  )                              HYS01680
C                                                                       HYS01690
C..... DETERMINE MAXIMUN AND MINIMUN VALUES                             HYS01700
      PMG=MAX(PMG, PC,P)                                                HYS01710
      DMG=MAX(DMG, DC,D)                                                HYS01720
      PMH=MIN(PMH,-PC,P)                                                HYS01730
      DMH=MIN(DMH,-DC,D)                                                HYS01740
      DMAX=MAX(DMG,-DMH)                                                HYS01750
      PMAX=MAX(PMG,-PMH)                                                HYS01760
C                                                                       HYS01770
C..... SET MAXIMUN AND MINIMUN VALUES...                                HYS01780
      IF (DIR.EQ.1 .OR. DIR.EQ.2 ) THEN                                 HYS01790
            PM=PMG                                                      HYS01800
            DM=DMG                                                      HYS01810
      ELSE IF (DIR.EQ.3 .OR. DIR.EQ.4) THEN                             HYS01820
            PM=PMH                                                      HYS01830
            DM=DMH                                                      HYS01840
      ENDIF                                                             HYS01850
C                                                                       HYS01860
C..... CALCULATE EQUIVALENT UNLOADING STIFFNESS FOR ENERGY...           HYS01870
        IF (BACKB .OR. RULE.EQ.0.0) THEN                                HYS01880
              S1=FS1(DMAX)                                              HYS01890
              S2=FS2(DMAX)                                              HYS01900
              S3=FS3(DMAX)                                              HYS01910
              DO=DM-PM*(DMG-DMH)/(PMG-PMH)                              HYS01920
              PA=PM-SIGN(PC,PM)                                         HYS01930
              DA=DM-SIGN(PC,PM)/S1                                      HYS01940
              PB=SIGN( (MIN(ABS(PA),(PC/2)) ),PM)                       HYS01950
              SO2=KMIN((DA-DO),PA,SI)                                   HYS01960
              DB=DA-(PA-PB)/MAX(S2,SO2)                                 HYS01970
              SO3=KMIN((DB-DO),PB,SI)                                   HYS01980
              SE= 0.5 * PM**2 / (                                       HYS01990
     &              (PM+PA)*0.5*(DM-DA)                                 HYS02000
     &            +(PA+PB)*0.5*(DA-DB)                                  HYS02010
     &            +(   PB)*0.5*(   PB)/MAX(S3,SO3) )                   HYS02020
        ENDIF                                                           HYS02030
C                                                                       HYS02040
C..... CHECK TO SEE IF LAST RULE IS STILL IN EFFECT                     HYS02050
      IF (DIR.EQ.DIRL .AND.                                             HYS02060
     &  ((DIR.EQ.1 .AND. P.LT.A) .OR. (DIR.EQ.2 .AND. P.GT.A)  .OR.     HYS02070
     &  (DIR.EQ.3 .AND. P.GT.A) .OR. (DIR.EQ.4 .AND. P.LT.A))) THEN     HYS02080
        RETURN                                                          HYS02090
      ENDIF                                                             HYS02100
C                                                                       HYS02110
C..... CALC EXCURSION RATIO AT EVERY ZERO CROSSING...                   HYS02120
      IF (P*PL.LE.0) CALL DNE(0,UPOS,EXCR, DY,DLI,ESE,PSE,PSEOLD,CSE)   HYS02130
C                                                                       HYS02140
C..... ERASE SMALL LOOP FLAGS                                           HYS02150
      IF (IR.GT.0) THEN                                                 HYS02160
        IT=IR                                                           HYS02170
        DO 10 I=1,IT                                                    HYS02180
          IF ((FR(I).EQ.'L' .AND. ( P.LE.PR(I) .OR. D.LE.DR(I) ))  .OR. HYS02190
     &        (FR(I).EQ.'7' .AND. ( P.GE.PR(I) .OR. D.GE.DR(I) ))) THEN HYS02200
            IR=I-1                                                      HYS02210
            GO TO 15                                                    HYS02220
          ENDIF                                                         HYS02230
   10   CONTINUE                                                        HYS02240
      ENDIF                                                             HYS02250
   15 I=IR+1                                                            HYS02260
C                                                                       HYS02270
      IF (DIR.EQ.1 .OR. DIR.EQ.3 )  THEN                                HYS02280
C..... LOADING RULES ..................................... RULES 1,5-11HYS02290
        PC2=SIGN((PC/2),PM)                                             HYS02300
C                                                                       HYS02310
C..... LOADING ON BACKBONE                                    RULE 1    HYS02320
   30   IF (BACKB .OR. RULE.EQ.0.0) THEN                                HYS02330
          K=0.                                                          HYS02340
          DO 40 J=2,NS                                                  HYS02350
            DD=DS(J)-ABS(D)                                             HYS02360
            DP=PS(J)-AP                                                 HYS02370
            IF (AP.LT.PS(J) .AND. (DD*DP).GT.0.) THEN                   HYS02380
              S=DP/DD                                                   HYS02390
              IF (S.GT.K .AND. S.LE.SI) THEN                            HYS02400
                K=S                                                     HYS02410
                A=SIGN(PS(J),P)                                         HYS02420
              ENDIF                                                     HYS02430
            ENDIF                                                       HYS02440
   40     CONTINUE                                                      HYS02450
          RULE=1                                                        HYS02460
          BACKB=.TRUE.                                                  HYS02470
          IR=0                                                          HYS02480
C                                                                       HYS02490
C..... LOADING AFTER UNLOADING                          RULES 6-11HYS02500
        ELSE                                                            HYS02510
            CRACKD=(ABS(DM).GT.DC)                                      HYS02520
            REVRSL=((DIR.EQ.1 .AND. DIRL.EQ.4) .OR. LRULE.EQ.8          HYS02530
     &         .OR. (DIR.EQ.3 .AND. DIRL.EQ.2) .OR. LRULE.EQ.9         HYS02540
     &         .OR. LRULE.EQ.5  .OR. LRULE.EQ.11)                       HYS02550
            IF (CRACKD) THEN                                            HYS02560
                P2=.95*SIGN(PMAX,PM)                                    HYS02570
                D2=FD2(DMAX)                                            HYS02580
            ELSE                                                        HYS02590
                P2=PM                                                   HYS02600
                D2=DM                                                   HYS02610
            ENDIF                                                       HYS02620
            S=KMIN((D2-D),(P2-P),SI)                                    HYS02630
   20       I=I-1                                                       HYS02640
C                                                                       HYS02650
C..... RELOADING INSIDE SMALL LOOPS                     RULE 11HYS02660
            IF (I.GT.0 ) THEN                                           HYS02670
              IF (.NOT.(( FR(I).EQ.'7' .AND. DIR.EQ.1 )  .OR.           HYS02680
     &                  ( FR(I).EQ.'L' .AND. DIR.EQ.3 ))) GO TO 20HYS02690
              IF (SR1.LE.0 .OR. SR2.LE.0 .OR. SR3.LE.0 .OR.             HYS02700
     &            SRM.LE.0 ) THEN                                       HYS02710
                  I=1                                                   HYS02720
```

```
                GO TO 20                                                HYS02730
              ENDIF                                                     HYS02740
              IF (AP.LT.ABS(PC2) .AND. S.GT.SRM) THEN                   HYS02750
                 REVRSL=.FALSE.                                         HYS02760
                 I=1                                                    HYS02770
                 GO TO 20                                               HYS02780
              ENDIF                                                     HYS02790
              IF (ABS(PR(I)).LE. PC/4) THEN                             HYS02800
                 K=KMIN((D-DR(I)),(P-PR(I)),SI)                         HYS02810
                 A=PR(I)                                                HYS02820
                 RULE=11.1+I/1000.                                      HYS02830
              ELSE IF (ABS(PR(I)).LE.3*PC/4) THEN                       HYS02840
                 X2=D2-(P2-1.5*PC2)/SR3                                 HYS02850
                 X1=X2-PC2/SR2                                          HYS02860
                 X=D+(PC2/2-P) /SR2                                     HYS02870
                 IF (AP.LT.PC/4 .AND. ((DIR.EQ.1 .AND. X.LT.X1).OR.    HYS02880
     &               (DIR.EQ.3 .AND. X.GT.X1))) THEN                    HYS02890
                      K=KMIN((X1-D),(PC2/2-P),SI)                       HYS02900
                      A=SIGN((PC/4),PM)                                 HYS02910
                      RULE=11.2+I/1000.                                 HYS02920
                 ELSE                                                   HYS02930
                      K=KMIN((DR(I)-D),(PR(I)-P),SI)                    HYS02940
                      A=PR(I)                                           HYS02950
                      RULE=11.3+I/1000.                                 HYS02960
                 ENDIF                                                  HYS02970
              ELSE                                                      HYS02980
                 X2=DR(I)-(PR(I)-1.5*PC2)/SRM                           HYS02990
                 X1=X2-PC2/SR2                                          HYS03000
                 X=D+(PC2/2-P) /SR2                                     HYS03010
                 IF (AP.LT.PC/4 .AND. ((DIR.EQ.1 .AND. X.LT.X1).OR.    HYS03020
     &               (DIR.EQ.3 .AND. X.GT.X1))) THEN                    HYS03030
                      K=KMIN((X1-D),(PC2/2-P),SI)                       HYS03040
                      A=SIGN((PC/4),PM)                                 HYS03050
                      RULE=11.4+I/1000.                                 HYS03060
                 ELSE                                                   HYS03070
                      X=D+(1.5*PC2-P) /SR3                              HYS03080
                      IF (AP.LT. 3*PC/4  .AND.                          HYS03090
     &                          ((DIR.EQ.1 .AND. X.LT.X2)               HYS03100
     &                     .OR. (DIR.EQ.3 .AND. X.GT.X2))) THEN         HYS03110
                           K=KMIN((X2-D),(1.5*PC2-P),SI)                HYS03120
                           A=SIGN((3*PC/4),PM)                          HYS03130
                           RULE=11.5+I/1000.                            HYS03140
                      ELSE                                              HYS03150
                           SR3=SRM                                      HYS03160
                           K=KMIN((DR(I)-D),(PR(I)-P),SI)               HYS03170
                           A=PR(I)                                      HYS03180
                           RULE=11.6+I/1000.                            HYS03190
                      ENDIF                                             HYS03200
                 ENDIF                                                  HYS03210
              ENDIF                                                     HYS03220
C                                                                       HYS03230
C.............. INITIAL RELOADING BELOW PC/2  W/O REVERSAL       RULE 6 HYS03240
              ELSE IF (AP.LT.ABS(PC2) .AND. .NOT.REVRSL ) THEN          HYS03250
                 K=FS1(DMAX)                                            HYS03260
                 A=PC2                                                  HYS03270
                 RULE=6                                                 HYS03280
C                                                                       HYS03290
C.............. RELOADING TO P2                                  RULE 7 HYS03300
              ELSE IF (AP.LT.ABS(P2) .AND.                              HYS03310
     &             (.NOT.REVRSL .OR. AP.GE.PC*0.75)) THEN               HYS03320
                 K=S                                                    HYS03330
                 A=P2                                                   HYS03340
                 RULE=7                                                 HYS03350
C                                                                       HYS03360
C.............. RELOADING BELOW PC WITH REVERSAL            RULES 8 & 9 HYS03370
              ELSE IF (AP.LT.PC*0.75 ) THEN                             HYS03380
                 SR=FSR(DMAX)                                           HYS03390
                    S1=FS1(DMAX)                                        HYS03400
                    S2=FS2(DMAX)                                        HYS03410
                    S3=FS3(DMAX)                                        HYS03420
                    PA=SIGN(PMAX,PM)-SIGN(PC,PM)                        HYS03430
                    DA=SIGN(DMAX,PM)-SIGN(PC,PM)/S1                     HYS03440
                    PB=SIGN( (MIN(ABS(PA),(PC/2)) ),PM)                 HYS03450
                    DB=DA-(PA-PB)/S2                                    HYS03460
                    DOP=DB-(PB)/S3                                      HYS03470
                    DO=DM-PM*(DMG-DMH)/(PMG-PMH)                        HYS03480
                    IF (DIR.EQ.1) DOP=MAX(DOP,DO)                       HYS03490
                    IF (DIR.EQ.3) DOP=MIN(DOP,DO)                       HYS03500
                 DC2=DOP+PC2/S1                                         HYS03510
                 SRM=KMIN((DC2-D2),(PC2-P2),S1)                         HYS03520
                 IF (S.GT.SRM) THEN                                     HYS03530
                      REVRSL=.FALSE.                                    HYS03540
                      GO TO 20                                          HYS03550
                 ENDIF                                                  HYS03560
                 SRP=KMIN((DC2-D),(PC2-P),S1)                           HYS03570
                 SR1=MAX(SRP,MIN(SR,S))                                 HYS03580
                 DC2P=D+(PC2-P)/SR1                                     HYS03590
                 SR3=KMIN((D2-DC2P),(P2-PC2),SI)                        HYS03600
                 SR2=2.00/( 1./SR1 + 1./SR3 )                           HYS03610
                 IF (AP.LT.(PC/4) ) THEN                                HYS03620
                    K=SR1                                               HYS03630
                    A=SIGN((PC/4),PM)                                   HYS03640
                    RULE=8                                              HYS03650
                 ELSE                                                   HYS03660
                    K=SR2                                               HYS03670
                    A=SIGN(PC,PM)*0.75                                  HYS03680
                    RULE=9                                              HYS03690
                 ENDIF                                                  HYS03700
C                                                                       HYS03710
C.............. LOADING TOWARDS THE BACKBONE CURVE             RULE 10HYS03720
              ELSE                                                      HYS03730
                 BACKB=.TRUE.                                           HYS03740
                 IF (.NOT. CRACKD) GO TO 30                             HYS03750
                 IR=0                                                   HYS03760
                 ALPHAP=1.0                                             HYS03770
                 IF (ABS(DM).EQ.ABS(DMAX))  ALPHAP=ALPHA                HYS03780
                 K=KMIN((ALPHAP*SIGN(DMAX,D2)-D2),(SIGN(PMAX,P2)-P2)    HYS03790
     &                ,SI)                                              HYS03800
                 DO 50 J=2,NS                                           HYS03810
                   IF ( PS(J).LE.ABS(PMAX) ) GO TO 50                   HYS03820
                   JO=J-1                                               HYS03830
                   DD=DS(J)-DS(JO)                                      HYS03840
                   DP=PS(J)-PS(JO)                                      HYS03850
                   IF ((DP*DD).LE.0.) GO TO 50                          HYS03860
                    CALL INTSCT(DX,PX,D,P,K,SIGN(DS(J),P),              HYS03870
     &                          SIGN(PS(J),P),(DP/DD) )                 HYS03880
                    IF (ABS(DX).GE.DS(JO) .AND. ABS(DX).LE.DS(J).AND.   HYS03890
     &                  ABS(PX).GT.ABS(PMAX).AND.  PM*PX.GT.0) THEN     HYS03900
                      IF (AP.GE.ABS(PX)) GO TO 30                       HYS03910
                      K=KMIN((DX-D),(PX-P),SI)                          HYS03920
                      A= PX                                             HYS03930
                      RULE=10                                           HYS03940
                      GO TO 100                                         HYS03950
                    ENDIF                                               HYS03960
   50            CONTINUE                                               HYS03970
                 GO TO 30                                               HYS03980
              ENDIF                                                     HYS03990
           ENDIF                                                        HYS04000
C                                                                       HYS04010
C                                                                       HYS04020
        ELSE                                                            HYS04030
C..... UNLOADING RULES ..........................................       HYS04040
C                                                                       HYS04050
C..... INITILIZE VALUES                                  RULES 2-5HYS04060
              BACKB=.FALSE.                                             HYS04070
              S1=FS1(DMAX)                                              HYS04080
              S2=FS2(DMAX)                                              HYS04090
              S3=FS3(DMAX)                                              HYS04100
              DO=DM-PM*(DMG-DMH)/(PMG-PMH)                              HYS04110
              PA=PM-SIGN(PC,PM)                                         HYS04120
              DA=DM-SIGN(PC,PM)/S1                                      HYS04130
              PB=SIGN( (MIN(ABS(PA),(PC/2)) ),PM)                       HYS04140
                 S02=KMIN((DA-DO),PA,SI)                                HYS04150
                 DB=DA-(PA-PB)/MAX(S2,S02)                              HYS04160
                 S03=KMIN((DB-DO),PB,SI)                                HYS04170
                 SU=0.                                                  HYS04180
C                                                                       HYS04190
C.............. UNLOADING INSIDE SMALL POSITIVE LOOPS            RULE 5 HYS04200
              I=IR+1                                                    HYS04210
   70         I=I-1                                                     HYS04220
              IF (I.GT.0) THEN                                          HYS04230
                IF (.NOT.((DIR.EQ.2 .AND. FR(I).EQ.'L')    .OR.         HYS04240
     &                   (DIR.EQ.4 .AND. FR(I).EQ.'7')) ) GO TO 70     HYS04250
C                                                                       HYS04260
                IF ((DIR.EQ.2 .AND.( (P.GT.PA .AND. PR(I).GT.PA) .OR.   HYS04270
     &               (PA.GE.P .AND.  P.GT.PB .AND. PR(I).GT.PB) .OR.    HYS04280
     &               (PB.GE.P .AND.  PR(I).GT.0 )) .OR.                 HYS04290
     &              (DIR.EQ.4 .AND.( (P.LT.PA .AND. PR(I).LT.PA) .OR.   HYS04300
     &               (PA.LE.P .AND.  P.LT.PB .AND. PR(I).LT.PB) .OR.    HYS04310
     &               (PB.LE.P .AND.  PR(I).LT.0 ))) ) THEN              HYS04320
                   K=(PR(I)-P)/(DR(I)-D)                                HYS04330
                   IF (K.LE.0 .OR. K.GT.SI) GO TO 70                    HYS04340
                   A=PR(I)                                              HYS04350
                   RULE=5+I/1000.                                       HYS04360
                   GO TO 100                                            HYS04370
                ELSE                                                    HYS04380
                   SU=KMIN((DR(I)-D),(PR(I)-P),SI)                      HYS04390
                ENDIF                                                   HYS04400
C                                                                       HYS04410
              ENDIF                                                     HYS04420
C                                                                       HYS04430
C.............. CALC DUCTILITY AND EXCURSION RATIO...                  HYS04440
              IF  (RULE.EQ.1 .OR. RULE.GT.5)  THEN                      HYS04450
C             IF ((RULE.EQ.1 .OR. RULE.GT.5) .AND. AP.GE.0.95*PMAX) THENHYS04460
                IF (DIR.EQ.2) THEN                                      HYS04470
                  DLI= MAX( D , DL )                                    HYS04480
                  CALL DNE(1,UPOS,EXCR, DY,DLI,ESE(2),PSE,PSEOLD,CSE)   HYS04490
                ELSE IF (DIR.EQ.4) THEN                                 HYS04500
                  DLI= MIN( D , DL )                                    HYS04510
                  CALL DNE(1,UNEG,EXCR,-DY,DLI,ESE(3),PSE,PSEOLD,CSE)   HYS04520
                ENDIF                                                   HYS04530
              ENDIF                                                     HYS04540
C                                                                       HYS04550
C.............. UNLOADING ON THE TOP SEGMENT                    RULE 2 HYS04560
              IF(AP.GT.ABS(PA)) THEN                                    HYS04570
                 K=MAX(S1,SU)                                           HYS04580
                 A=PA                                                   HYS04590
                 RULE=2                                                 HYS04600
C.............. UNLOADING ON THE MIDDLE SEGMENT                 RULE 3 HYS04610
              ELSE IF(AP.GT.ABS(PB)) THEN                               HYS04620
                 K=MAX(S2,S02,SU)                                       HYS04630
                 A=PB                                                   HYS04640
                 RULE=3                                                 HYS04650
C.............. UNLOADING ON THE BOTTOM SEGMENT                 RULE 4 HYS04660
              ELSE                                                      HYS04670
                 K=MAX(S3,S03,SU)                                       HYS04680
                 A=0.                                                   HYS04690
                 RULE=4                                                 HYS04700
              ENDIF                                                     HYS04710
C                                                                       HYS04720
      ENDIF                                                             HYS04730
C                                                                       HYS04740
C..... SAVE REVERSAL POINTS FOR SMALL LOOPS.....................        HYS04750
 100  IRULE=INT(RULE)                                                  HYS04760
      REVRSL=IRULE.EQ.1.OR.IRULE.EQ.10.OR.LRULE.EQ.1.OR.LRULE.EQ.10     HYS04770
      IF      ((DIRL.EQ.1 .AND. DIR.EQ.2 .AND. .NOT. REVRSL)            HYS04780
     &          .OR. (DIRL.EQ.4 .AND. DIR.EQ.3) ) THEN                  HYS04790
            IF (IR.LT.NI)  THEN                                         HYS04800
               IR=IR+1                                                  HYS04810
               PR(IR)=P                                                 HYS04820
               DR(IR)=D                                                 HYS04830
               FR(IR)='7'                                               HYS04840
            ENDIF                                                       HYS04850
      ELSE IF ((DIRL.EQ.3 .AND. DIR.EQ.4 .AND. .NOT. REVRSL)            HYS04860
     &          .OR. (DIRL.EQ.2 .AND. DIR.EQ.1) ) THEN                  HYS04870
            IF (IR.LT.NI)  THEN                                         HYS04880
               IR=IR+1                                                  HYS04890
               PR(IR)=P                                                 HYS04900
               DR(IR)=D                                                 HYS04910
               FR(IR)='L'                                               HYS04920
            ENDIF                                                       HYS04930
      ENDIF                                                             HYS04940
      IF ((1.05*SI).GT.K .AND. K.GT.0 ) GO TO 9999                      HYS04950
C                                                                       HYS04960
      K=0                                                               HYS04970
      RULE=-RULE                                                        HYS04980
 9999 CONTINUE                                                          HYS04990
C     WRITE (6,1111) P,D,K,RULE,DIR,A                                   HYS05000
C1111 FORMAT (1P,                                                       HYS05010
C    &' SI= P=',G13.5,'  D=',G13.5,'  K=',G18.9,'  RULE=',G13.5,        HYS05020
C    &'  DIR=',I5,'  A=',G13.5)                                         HYS05030
      RETURN                                                            HYS05040
C                                                                       HYS05050
      END                                                               HYS05060
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP                                  HYS00010
C*********************************************************************  HYS00020
C=== TAKEDA HYSTERESIS MODEL ..... HYST06 ..............................HYS00030
C*********************************************************************  HYS00040
C          DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                     HYS00050
C          AT 8000                                                      HYS00060
C          TRACE ON                                                     HYS00070
C          END DEBUG                                                    HYS00080
C*********************************************************************  HYS00090
C                                                                       HYS00100
C   TAKEDA HYSTERISIS MODEL - REFERANCE, S. OTANI, UILU-ENG-74-2029     HYS00110
C                     TAKEDA & SOZEN, J. ASCE ST. DIV. - DEC. 1970      HYS00120
C                                                                       HYS00130
C  PI       CURRENT LOAD                                                HYS00140
C  DI       CURRENT DISPL                                               HYS00150
C  PS       PREVIOUS LOAD                                               HYS00160
C  DS       PREVIOUS DISPL                                              HYS00170
C  VI       CURRENT VELOCITY                                            HYS00180
C  VL       PREVIOUS VELOCITY                                           HYS00190
C  S        CURRENT STIFFNESS                                           HYS00200
C  NRLN     CURRENT RULE NUMBER                                         HYS00210
C  A        UPPER LOAD LIMIT BEFORE RULE CHANGE                         HYS00220
C  B        LOWER LOAD LIMIT BEFORE RULE CHANGE                         HYS00230
C  IDRC  77PREVIOUS DIRECTION                                           HYS00240
C  RF                                                                   HYS00250
C  PC,DC    CRACKING LOAD AND DISPLACEMENT                              HYS00260
C  PY,DY    YEILD LOAD AND DISPLACEMENT                                 HYS00270
C  PU,DU    ULTIMATE LOAD AND DISPLACEMENT                              HYS00280
C  OC       ELASTIC STIFFNESS PD/DC                                     HYS00290
C  CY       POST CRACKING STIFFNESS (PY-PC)/(DY-DC)                     HYS00300
C  YU       POST YEILD STIFFNESS (PU-PY)/(DU-DY)                        HYS00310
C  CPY      CRACKING', YEILD STIFFNESS (PY+PC)/(DY+DC)                  HYS00320
C  CSE                                                                  HYS00330
C  RNRL     NEXT RULE NUMBER                                            HYS00340
C  UM1      PEAK DISPLACEMENT IN THE POSITIVE DIRECTION                 HYS00350
C  UM2      PEAK LOAD IN THE POSITIVE DIRECTION                         HYS00360
C  UM3      PEAK DISPLACEMENT IN THE NEGATIVE DIRECTION                 HYS00370
C  UM4      PEAK LOAD IN THE NEGATIVE DIRECTION                         HYS00380
C  S1       UNLOADING STIFFNESS                                         HYS00390
C  X0                                                                   HYS00400
C  X0UM                                                                 HYS00410
C  X0I                                                                  HYS00420
C  U0D                                                                  HYS00430
C  U0                                                                   HYS00440
C  X1UM                                                                 HYS00450
C  U1                                                                   HYS00460
C  U1D                                                                  HYS00470
C  X2U0                                                                 HYS00480
C  U2                                                                   HYS00490
C  X3U1                                                                 HYS00500
```

```
C  U3
C  QT
C  MAXFLG  FLAG, IF MAXFLG=0 STIFFNESS=0   BEYOND FAILURE POINT (DU,PU)
C  ACHNG   LOAD WHERE RULE CHNGS IF LOADING CONTINUES IN CURR DIRECTION
C  PRINT   PRINT DATA FOR TRACING
C  SE      ELASTIC UNLOADING CURVE
C  ESE     ELASTIC STRAIN ENERGY
C  PSE     PLASTIC STRAIN ENERGY
C  PSEOLD  PLASTIC STRAIN ENERGY AT LAST ZERO CROSSING
C  UPOS    POSITIVE DUCTILITY INFO.
C  UNEG    NEGATIVE DUCTILITY INFO.
C  EXCR    EXCURSION RATIO INFO.
C
C  JI      # FOR CURRENT DIRECTION FOR LOADING AND UNLOADING
C          # FOR NEW DIRECTION FOR REVERSAL
C          # = 2 FOR POSITIVE LOAD ,  # = 1 FOR NEGATIVE LOAD
C  IREVSL  SIGN OF LOAD IN NEW DIRECTION
C  ISGN    SIGN OF THE CURRENT LOAD FOR LOADING AND UNLOADING
C          SIGN OF THE NEW LOAD FOR REVERSAL
C
C
      SUBROUTINE HYST06(PI,DI,PS,DS,VI,VS    ,S,HRLN,A,B,IDRO,RF,
     & PC,DC,PY,DY,PU,DU,QC,QY,TU,CPY,CSE,RNRL,UM1,UM2,UM3,UM4,
     & S1,XO  ,XOUM,XOY,UOD,UO,X1UM,U1,U1D ,X2U0,U2,X3U1,U3,QY,MAXFLG,
     & ACHNG,PRINT,SE,ESE,PSE,PSEOLD,UPOS,UNEG,EXCR)
C
      REAL   UM(2,2),IDRO,MAXFLG,UPOS(3),UNEG(3),EXCR(6),ESE(3)
      LOGICAL PRINT,BTEST
 8000 IF (PRINT) WRITE (6,8010) PI,DI,PS,DS,VI,VS,S,HRLN,A,B,RF,
     & PC,DC,PY,DY,PU,DU,QC,QY,TU,CPY,RNRL,UM1,UM2,UM3,UM4,
     & S1,XO  ,XOUM,XOY,UOD,UO,X1UM,U1,U1D ,X2U0,U2,X3U1,U3,QY,
     & ACHNG,IDRO,MAXFLG,CSE,
     & SE,ESE(1),PSE,PSEOLD,UPOS,UNEG,EXCR
 8010 FORMAT(
     &' TAKEDA-PI=',G12.5,'    DI=',G12.5,'   PS=',G12.5,'   DS=',G12.5/
     &' TK-     VI=',G12.5,'    VS=',G12.5,'    S=',G12.5,' HRLN=',G12.5/
     &' TK-      A=',G12.5,'     B=',G12.5,'   RF=',G12.5,'   PC=',G12.5/
     &' TK-     DC=',G12.5,'    PY=',G12.5,'   DY=',G12.5,'   PU=',G12.5/
     &' TK-     DU=',G12.5,'    QC=',G12.5,'   CY=',G12.5,'   YU=',G12.5/
     &' TK-    CPY=',G12.5,'  RNRL=',G12.5,'  UM1=',G12.5,'  UM2=',G12.5/
     &' TK-    UM3=',G12.5,'   UM4=',G12.5,'   S1=',G12.5,'   XO=',G12.5/
     &' TK-  XOUM=',G12.5,'   XOY=',G12.5,'  UOD=',G12.5,'   UO=',G12.5/
     &' TK-  X1UM=',G12.5,'    U1=',G12.5,'  U1D=',G12.5,' X2UO=',G12.5/
     &' TK-    U2=',G12.5,'  X3U1=',G12.5,'   U3=',G12.5,'   QY=',G12.5/
     &' TK- ACHNG=',G12.5,'  IDRO=',G12.5,' MAXF=',G12.5,'  CSE=',G12.5/
     &' TK-    SE=',G12.3,'  ESE=',G12.5,'PSE=',G12.5,' PSEO=',G12.5/
     &' TK-   UP1=',G12.3,'   UP2=',G12.5,'  UP3=',G12.5/
     &' TK-   UN1=',G12.3,'   UN2=',G12.5,'  UN3=',G12.5/
     &' TK-   EX1=',G12.3,'   EX2=',G12.5,'  EX3=',G12.5/
     &' TK-   EX4=',G12.3,'   EX5=',G12.5,'  EX6=',G12.5)
C
C....... RTN IF CURRENT STATE = PREVIOUS STATE
      IF ((PI.EQ.PS .OR. DI.EQ.DS) .AND. RNRL.NE.0 ) RETURN
C
C....... GET DIRECTION AND RTN IF CURRENT STATE IS WITHIN LIMITS...
      CALL IDRECT (IDR,IDRV,IDRVO,PI,PS,VI,VS)
      IF (A.GT.B .AND. PI.LE.A .AND. PI.GT.B .AND. IDR.EQ.IDRO) RETURN
      IF (A.LT.B .AND. PI.GT.A .AND. PI.LT.B .AND. IDR.EQ.IDRO) RETURN
C
C...... CALC EXCURSION RATIO AT EVERY ZERO CROSSING...
      IF (PI*PS.LE.0)CALL DNE(0,UPOS,EXCR, DY,DI,ESE,PSE,PSEOLD,CSE)
C
C...... INITILIZE DATA FOR FIRST CALL TO SUBROUTINE
 1    IF (RNRL.EQ.0) THEN
           RNRL=1
           UM1=-DC
           UM2=-PC
           UM3= DC
           UM4= PC
      ENDIF
C
C...... INITILIZE DATA
      NRL=RNRL
      UM(1,1)=UM1
      UM(1,2)=UM2
      UM(2,1)=UM3
      UM(2,2)=UM4
C
      IF (PI.GT.0) THEN
 901      ISGN =1
          JI=2
      ELSE IF (PI.LT.0) THEN
 902      ISGN =-1
          JI=1
      ELSE IF (DI.LT.DS) THEN
 903      ISGN =-1
          JI=1
      ELSE IF (DI.GT.DS) THEN
 904      ISGN =1
          JI=2
      ELSE
 905      ISGN =1
          JI=1
      ENDIF
C
      IF (DI.LE.DS) THEN
         IREVSL=-1
      ELSE
         IREVSL= 1
      ENDIF
C
      IF (IDR.LT.0) THEN
       IF (ISGN.GT.0) CALL DNE(1,UPOS,EXCR, DY,DI,ESE(2),PSE,PSEOLD,CSE)
       IF (ISGN.LT.0) CALL DNE(1,UNEG,EXCR,-DY,DI,ESE(3),PSE,PSEOLD,CSE)
      ENDIF
C
      GO TO ( 100, 200, 300, 400, 500, 600, 700, 800, 900,1000,
     1       1100,1200,1300,1400,1500,1600),NRL
C  ***** TAKEDA RULE  1.0 *****   ELASTIC STAGE
  100 IF (IDR) 130,130,110
  110 IF (ABS(PI).GE.PC) GO TO 200
      B=PI
      A=PC*ISGN
      S=QC
           SE=S
      NRL=1
              HRLN=1.11
       GO TO 2000
  130 B=0.
      A=PC*IREVSL
      IF (PI.NE.0) A=PC*SIGN(1.0,PI)
      S=QC
           SE=S
      NRL=1
              HRLN=1.2
       GO TO 2000
C  ***** TAKEDA RULE  2.0 *****  LOADING ON PRIMARY CURVE UP TO YEILD
  200 IF (IDR) 230,230,210
  210 IF (ABS(PI).GE.PY) GO TO 300
      B=PI
      A=PY*ISGN
      S=CY
           S1=(PI+(PC*ISGN))/(DI+DC*ISGN)
           SE=S1
      NRL=2
              HRLN=2.11
       GO TO 2000
  230 B=0.
      A=PI
      S1=(PI+(PC*ISGN))/(DI+DC*ISGN)
      S=S1
           SE=S1
      NRL=5
              HRLN=2.2
       GO TO 2000
C  ***** TAKEDA RULE  3.0 *****  LOADING ON PRIMARY CURVE AFTER YEILD
  300      BOTT=MAX(ABS(DI),ABS(UM1),ABS(UM3) )
           S1=CPY* SQRT(DY/BOTT)
           SE=S1
      IF (IDR) 330,330,310
  310 IF (ABS(PI).GE.PU) GO TO 320
      A=PU*ISGN
      B=PI
      S=YU
      NRL=3
              HRLN=3.11
       GO TO 2000
  320 A=PU*ISGN
      B=PI
      S=0.
      IF (MAXFLG.NE.0) S=YU
      NRL=3
              HRLN=3.12
       GO TO 2000
  330 B=0.
      A=PI
      BOTT=MAX(ABS(DI),ABS(UM1),ABS(UM3) )
      S1=CPY* SQRT(DY/BOTT)
      S=S1
      NRL=4
              HRLN=3.2
       GO TO 2000
C  ***** TAKEDA RULE  4.0 *****  UNLOADING FROM POINT UM  ON THE PRIMARY
C                                CURVE AFTER YEILDING
  400 IF (IDR) 430,440,410
  410 IF (ABS(PI).GE.ABS(UM(JI,2))) GO TO 420
      A=UM(JI,2)
      B=PI
      S=S1
      NRL=4
              HRLN=4.11
       GO TO 2000
  420 A=PU*ISGN
      B=PI
      S=YU
      NRL=3
              HRLN=4.12
       GO TO 2000
  430 A=PI
      B=0.
      S=S1
      NRL=4
              HRLN=4.2
       GO TO 2000
  440 IF (DI.LE.DS) IDRVO=1
      IF (DI.GT.DS) IDRVO=2
      IF (ABS(UM(IDRVO,1)).GT.DC) GO TO 450
      A=PC*IREVSL
      B=0.
      S=S1
      NRL=15
              HRLN=4.31
       GO TO 2000
  450 XO=DI
      XOUM=(0.-UM(IDRVO,2))/(XO-UM(IDRVO,1))
      XOY =(0.-PY*ISGN)/(XO-DY*ISGN)
      IF (PY.GT.ABS(UM(IDRVO,2)) .AND. XOY.GT.XOUM) GO TO 460
      A=UM(IDRVO,2)
      B=0.
      S2=XOUM
      S=S2
      NRL=6
              HRLN=4.32
       GO TO 2000
  460 UM(JI,1)=DY*ISGN
      UM(JI,2)=PY*ISGN
      A=UM(JI,2)
      B=0.
      S2=XOY
      S=S2
      NRL=6
              HRLN=4.33
       GO TO 2000
C  ***** TAKEDA RULE  5.0 *****   UNLOADING FROM POINT UM ON THE PRIMARY
C                                 CURVE BEFORE YEILDING
  500 IF (IDR) 530,540,510
  510 IF (ABS(PI).GE.ABS(UM(JI,2))) GO TO 200
      A=UM(JI,2)
      B=PI
      S=S1
      NRL=5
              HRLN=5.11
       GO TO 2000
  530 A=PI
      B=0.
      S=S1
      NRL=5
              HRLN=5.2
       GO TO 2000
  540 IF (DI.LE.DS) IDRV=2
      IF (DI.GT.DS) IDRV=1
      IF (ABS(UM(IDRV,1)).GT.DC) GO TO 450
      A=PC*IREVSL
      B=0.
      S=S1
      NRL=14
              HRLN=5.31
       GO TO 2000
C  ***** TAKEDA RULE  6.0 *****   LOADING TOWARDS POINT UM ON THE PRIMARY
C                                 CURVE.
  600 IF (IDR) 630,630,610
  610 IF (ABS(PI).GE.ABS(UM(JI,2))) GO TO 210
      A=UM(JI,2)
      B=PI
      XOUM=(PI-UM(JI,2))/(DI-UM(JI,1))
      S=XOUM
      NRL=6
              HRLN=6.11
       GO TO 2000
  630 UOD=DI
      UO=PI
      A=PI
      B=0.
      S=S1
      NRL=7
              HRLN=6.20
       GO TO 2000
C  ***** TAKEDA RULE  7.0 *****  UNLOADING FROM POINT UM AFTER RULE 6
  700 IF (IDR) 730,740,710
  710 IF (ABS(PI).GE.ABS(UO)) GO TO 720
      A=UO
      B=PI
      S=S1
      NRL=7
              HRLN=7.11
       GO TO 2000
  720 A=UM(JI,2)
      B=PI
      XOUM=(0.-UM(JI,2))/(XO-UM(JI,1))
      S=XOUM
      NRL=6
              HRLN=7.12
       GO TO 2000
  730 A=PI
```

```
      B=0.                                                              HYS03350
      S=S1                                                              HYS03360
      NRL=7                                                             HYS03370
            HRLN=7.2                                                    HYS03380
       GO TO 2000                                                       HYS03390
  740 A=UM(JI,2)                                                        HYS03400
      B=0.                                                              HYS03410
      X1UM=(PI-UM(JI,2))/(DI-UM(JI,1))                                  HYS03420
      S=X1UM                                                            HYS03430
      NRL=8                                                             HYS03440
            HRLN=7.3                                                    HYS03450
       GO TO 2000                                                       HYS03460
C  ***** TAKEDA RULE  8.0 *****   LOADING TOWARDS POINT UM ON THE PRIMARHYS03470
C                                 CURVE                               HYS03480
  800 IF (IDR) 830,830,810                                              HYS03490
  810 IF (ABS(PI).GE.ABS(UM(JI,2))) GO TO 210                           HYS03500
      A=UM(JI,2)                                                        HYS03510
      B=PI                                                              HYS03520
      X1UM=(PI-UM(JI,2))/(DI-UM(JI,1))                                  HYS03530
      S=X1UM                                                            HYS03540
      NRL=8                                                             HYS03550
            HRLN=8.11                                                   HYS03560
       GO TO 2000                                                       HYS03570
  830 U1=PI                                                             HYS03580
      U1D=DI                                                            HYS03590
      A=PI                                                              HYS03600
      B=0.                                                              HYS03610
      S=S1                                                              HYS03620
      NRL=9                                                             HYS03630
            HRLN=8.2                                                    HYS03640
       GO TO 2000                                                       HYS03650
C  ***** TAKEDA RULE  9.0 *****   UNLOADING FROM POINT U1 AFTER RULE 8  HYS03660
  900 IF (IDR) 930,940,910                                              HYS03670
  910 IF (ABS(PI).GE.ABS(U1       )) GO TO 810                          HYS03680
      A=U1                                                              HYS03690
      B=PI                                                              HYS03700
      S=S1                                                              HYS03710
      NRL=9                                                             HYS03720
            HRLN=9.11                                                   HYS03730
       GO TO 2000                                                       HYS03740
  930 A=PI                                                              HYS03750
      B=0.                                                              HYS03760
      S=S1                                                              HYS03770
      NRL=9                                                             HYS03780
            HRLN=9.2                                                    HYS03790
       GO TO 2000                                                       HYS03800
  940 A=U0                                                              HYS03810
      B=0.                                                              HYS03820
      X2U0=(PI-U0)/(DI-U0D)                                             HYS03830
      S=X2U0                                                            HYS03840
      NRL=10                                                            HYS03850
            HRLN=9.3                                                    HYS03860
       GO TO 2000                                                       HYS03870
C  ***** TAKEDA RULE 10.0 *****   LOADING TOWARDS POINT U0             HYS03880
 1000 IF (IDR) 1030,1030,1010                                           HYS03890
 1010 IF (ABS(PI).GE.ABS(U0       )) GO TO 610                         HYS03900
      A=U0                                                              HYS03910
      B=PI                                                              HYS03920
      X2U0=(PI-U0)/(DI-U0D)                                             HYS03930
      S=X2U0                                                            HYS03940
      NRL=10                                                            HYS03950
            HRLN=10.11                                                  HYS03960
       GO TO 2000                                                       HYS03970
 1030 U2=PI                                                             HYS03980
      A=PI                                                              HYS03990
      B=0.                                                              HYS04000
      S=S1                                                              HYS04010
      NRL=11                                                            HYS04020
            HRLN=10.2                                                   HYS04030
       GO TO 2000                                                       HYS04040
C  ***** TAKEDA RULE 11.0 *****   UNLOADING FROM POINT U2 AFTER RULE 10 HYS04050
 1100 IF (IDR) 1130,1140,1110                                           HYS04060
 1110 IF (ABS(PI).GE.ABS(U2       )) GO TO 1010                        HYS04070
      A=U2                                                              HYS04080
      B=PI                                                              HYS04090
      S=S1                                                              HYS04100
      NRL=11                                                            HYS04110
            HRLN=11.11                                                  HYS04120
       GO TO 2000                                                       HYS04130
 1130 A=PI                                                              HYS04140
      B=0.                                                              HYS04150
      S=S1                                                              HYS04160
      NRL=11                                                            HYS04170
            HRLN=11.2                                                   HYS04180
       GO TO 2000                                                       HYS04190
 1140 A=U1                                                              HYS04200
      B=0.                                                              HYS04210
      X3U1=(PI-U1)/(DI-U1D)                                             HYS04220
      S=X3U1                                                            HYS04230
      NRL=12                                                            HYS04240
            HRLN=11.3                                                   HYS04250
       GO TO 2000                                                       HYS04260
C  ***** TAKEDA RULE 12.0 *****   LOADING TOWARDS POINT U1             HYS04270
 1200 IF (IDR) 1230,1230,1210                                           HYS04280
 1210 IF (ABS(PI).GE.ABS(U1       )) GO TO  810                        HYS04290
      A=U1                                                              HYS04300
      B=PI                                                              HYS04310
      X3U1=(PI-U1)/(DI-U1D)                                             HYS04320
      S=X3U1                                                            HYS04330
      NRL=12                                                            HYS04340
            HRLN=12.11                                                  HYS04350
       GO TO 2000                                                       HYS04360
 1230 U3=PI                                                             HYS04370
      A=PI                                                              HYS04380
      B=0.                                                              HYS04390
      S=S1                                                              HYS04400
      NRL=13                                                            HYS04410
            HRLN=12.2                                                   HYS04420
       GO TO 2000                                                       HYS04430
C  ***** TAKEDA RULE 13.0 *****   UNLOADING FROM POINT U3 AFTER RULE 12  HYS04440
 1300 IF (IDR) 1330, 940,1310                                           HYS04450
 1310 IF (ABS(PI).GE.ABS(U3       )) GO TO 1210                        HYS04460
      A=U3                                                              HYS04470
      B=PI                                                              HYS04480
      S=S1                                                              HYS04490
      NRL=13                                                            HYS04500
            HRLN=13.11                                                  HYS04510
       GO TO 2000                                                       HYS04520
 1330 A=PI                                                              HYS04530
      B=0.                                                              HYS04540
      S=S1                                                              HYS04550
      NRL=13                                                            HYS04560
            HRLN=13.2                                                   HYS04570
       GO TO 2000                                                       HYS04580
C  ***** TAKEDA RULE 14.0 *****   LOADING IN THE UNCRACKED DIRECTION AFTEHYS04590
C                                 CRACKING IN THE OTHER DIRECTION.    HYS04600
 1400 IF (IDR) 1430,1440,1410                                           HYS04610
 1410 IF (ABS(PI).GE.     PC      ) GO TO  210                         HYS04620
      A=PC*ISGN                                                         HYS04630
      B=0.                                                              HYS04640
      S=S1                                                              HYS04650
      NRL=14                                                            HYS04660
            HRLN=14.11                                                  HYS04670
       GO TO 2000                                                       HYS04680
 1430 A=PI                                                              HYS04690
      B=0.                                                              HYS04700
      S=S1                                                              HYS04710
      NRL=14                                                            HYS04720
            HRLN=14.2                                                   HYS04730
       GO TO 2000                                                       HYS04740
 1440 A=UM(JI,2)                                                        HYS04750
      B=0.                                                              HYS04760
      S=S1                                                              HYS04770
      NRL=5                                                             HYS04780
            HRLN=14.3                                                   HYS04790
       GO TO 2000                                                       HYS04800
C  ***** TAKEDA RULE 15.0 *****   LOADING IN THE UNCRACKED DIRECTION AFTEHYS04810
C                                 YEILDING IN THE OTHER DIRECTION.    HYS04820
 1500 IF (IDR) 1530,1540,1510                                           HYS04830
 1510 IF (ABS(PI).GE.     PC      ) GO TO 1520                         HYS04840
      A=PC*ISGN                                                         HYS04850
      B=PI                                                              HYS04860
      S=S1                                                              HYS04870
      NRL=15                                                            HYS04880
            HRLN=15.11                                                  HYS04890
       GO TO 2000                                                       HYS04900
 1520 PQ=PC*ISGN                                                        HYS04910
      DQ=(PQ-PI)/S1+DI                                                  HYS04920
      QY=(PY*ISGN-PQ)/(DY*ISGN-DQ)                                      HYS04930
      A=PY*ISGN                                                         HYS04940
      B=PQ                                                              HYS04950
      S=QY                                                              HYS04960
      NRL=16                                                            HYS04970
            HRLN=15.12                                                  HYS04980
       GO TO 2000                                                       HYS04990
 1530 A=PI                                                              HYS05000
      B=0.                                                              HYS05010
      S=S1                                                              HYS05020
      NRL=15                                                            HYS05030
            HRLN=15.2                                                   HYS05040
       GO TO 2000                                                       HYS05050
 1540 IF (ABS(PI) .GE. ABS(UM(JI,2))) GO TO 1550                        HYS05060
      A=UM(JI,2)                                                        HYS05070
      B=0.                                                              HYS05080
      S=S1                                                              HYS05090
      NRL=4                                                             HYS05100
            HRLN=15.31                                                  HYS05110
       GO TO 2000                                                       HYS05120
 1550 A=PU*IASVSL                                                       HYS05130
      B=0.                                                              HYS05140
      S=YU                                                              HYS05150
      NRL=3                                                             HYS05160
            HRLN=15.32                                                  HYS05170
       GO TO 2000                                                       HYS05180
C  ***** TAKEDA RULE 16.0 *****   LOADING TOWARDS THE YEILD POINT AFTER HYS05190
C                                 RULE 15.                            HYS05200
 1600 IF (IDR) 1630,1630,1610                                           HYS05210
 1610 IF (ABS(PI).GE.     PY      ) GO TO 300                          HYS05220
      A=PY*ISGN                                                         HYS05230
      B=PI                                                              HYS05240
      S=QY                                                              HYS05250
      NRL=16                                                            HYS05260
            HRLN=16.11                                                  HYS05270
       GO TO 2000                                                       HYS05280
 1630 UM(JI,1)=DY                                                       HYS05290
      UM(JI,2)=PY                                                       HYS05300
      U0=PI                                                             HYS05310
      U0D=DI                                                            HYS05320
      A=PI                                                              HYS05330
      B=0.                                                              HYS05340
      S=S1                                                              HYS05350
      S2=QY                                                             HYS05360
      X0=DI-PI/S2                                                       HYS05370
      NRL= 7                                                            HYS05380
            HRLN=16.2                                                   HYS05390
       GO TO 2000                                                       HYS05400
C                                                                       HYS05410
C ======> CALCULATE MAXIMUN PREVIOUS DEFORMATION                        HYS05420
 2000     UM1=MIN(UM(1,1),DI)                                           HYS05430
          UM2=MIN(UM(1,2),PI)                                           HYS05440
          UM3=MAX(UM(2,1),DI)                                           HYS05450
          UM4=MAX(UM(2,2),PI)                                           HYS05460
          RNRL=NRL                                                      HYS05470
          IDRO=IDR                                                      HYS05480
          RF=0.                                                         HYS05490
          IF (IDR.GE.0) ACHNG=A                                         HYS05500
          IF (IDR.LT.0) ACHNG=B                                         HYS05510
      RETURN                                                            HYS05520
      END                                                               HYS05530
C$PROCESS SDUMP OPT(0) GOSTMT XREF                                      HYS00010
C********** HYST07 ********************************************************HYS00020
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    HYS00030
C           END DEBUG                                                   HYS00040
C                                                                       HYS00050
C     SU_BROU_TINE ELSPLS - ELASTO-PLASTIC BILINEAR HYSTERESIS MODEL    HYS00060
C                                                                       HYS00070
C     PC       = CURRENT LOAD                                           HYS00080
C     PEQ      = MODIFIED CURRENT LOAD                                  HYS00090
C     DC       = CURRENT DISPL.                                         HYS00100
C     PL       = LAST LOAD                                              HYS00110
C     DL       = LAST DISPL.                                            HYS00120
C     PY       = YEILD LOAD                                             HYS00130
C     PYEQ     = MODIFIED YEILD LOAD                                    HYS00140
C     S        = ELASTIC STIFFNESS                                      HYS00150
C     K        = CURRENT STIFFNESS (OUTPUT)                             HYS00160
C     RULE     = CURRENT RULE NUMBER, USED TO TRACE MODEL (OUTPUT)      HYS00170
C     RNRL     = NEXT RULE NUMBER (OUTPUT)                              HYS00180
C     VI       = NOT USED                                               HYS00190
C     VL       = NOT USED                                               HYS00200
C     KIE      = INELASTIC STIFFNESS                                    HYS00210
C                                                                       HYS00220
      SUBROUTINE HYST07(PC,DC,PL,DL,K,RULE,PSEOLD,A,B,                  HYS00230
     &     VI,VL,S,PY,KIE,DY,CSE,PRINT,ESE,PSE,RKEY,UPOS,UNEG,EXCR)     HYS00240
      REAL   K,KIE                                                      HYS00250
      LOGICAL PRINT,BTEST                                               HYS00260
      DIMENSION UPOS(3),UNEG(3),EXCR(6),ESE(3)                          HYS00270
                                                                        HYS00280
      KEY=RKEY                                                          HYS00290
      PC=K*(DC-DL)+PL                                                   HYS00300
      IF (PRINT) WRITE (6,1010) PC,DC,PL,DL,K,RULE,VI,CSE,VL,S,PY,KIE,  HYS00310
     &    A,B,KEY,ESE(3),PSE,PSEOLD,UPOS,UNEG,EXCR                      HYS00320
 1010 FORMAT (                                                          HYS00330
     &' ELSPLS-PC=',G13.5,'    DC=',G13.5,'    PL=',G13.5,'  DL=',G13.5/HYS00340
     &' EL-     K=',G13.5,'  RULE=',G13.5,'    VI=',G13.5,' CSE=',G13.5/HYS00350
     &' EL-    VL=',G13.5,'     S=',G13.5,'    PY=',G13.5,' KIE=',G13.5/HYS00360
     &' EL-     A=',G13.5,'     B=',G13.5/                             HYS00370
     &' EL-   KEY=',G13.5,'   ESE=',G13.5,'   PSE=',G13.5,' PSO=',G13.5/HYS00380
     &' EL-   UP1=',G13.5,'   UP2=',G13.5,'   UP3=',G13.5/             HYS00390
     &' EL-   UN1=',G13.5,'   UN2=',G13.5,'   UN3=',G13.5/             HYS00400
     &' EL-   EX1=',G13.5,'   EX2=',G13.5,'   EX3=',G13.5/             HYS00410
     &' EL-   EX4=',G13.5,'   EX5=',G13.5,'   EX6=',G13.5)             HYS00420
C                                                                       HYS00430
C...... CALC EXCURSION RATIO AT EVERY ZERO CROSSING...                  HYS00440
      IF(PC*PL.LE.0)CALL DNE(0.UPOS,EXCR, DY,DL,ESE,PSE,PSEOLD,CSE)     HYS00450
C                                                                       HYS00460
C                                                                       HYS00470
C------------ ELASTIC REGION                                           HYS00480
      IF (RULE.LE.1 .AND. ABS(PC).LT.PY)  THEN                          HYS00490
         K=S                                                            HYS00500
         RULE=1.                                                        HYS00510
         A= PI                                                          HYS00520
         B=-PY                                                          HYS00530
         PC=S*DC                                                        HYS00540
         CALL STRENG (ESE,PSE,PSEOLD,PC,PL,DC,DL,K,S )                 HYS00550
         KEY=0                                                          HYS00560
         RKEY=KEY                                                       HYS00570
         GO TO 1000                                                     HYS00580
      ENDIF                                                             HYS00590
C                                                                       HYS00600
C------------ INELASTIC REGION                                         HYS00610
      DIR=DC-DL                                                         HYS00620
      IF (DIR.EQ.0) RETURN                                              HYS00630
      PEQ =PC-KIE*DC                                                    HYS00640
      PEQL=PL-KIE*DL                                                    HYS00650
```

```
      PYEQ=PY*(1-KIE/S)                                                 HYS00660
      DVEL=VI*VL                                                        HYS00670
      IF (PRINT) WRITE (6,*) 'PEQ,PYEQ,DVEL,VI,VL:'                     HYS00680
      IF (PRINT) WRITE (6,*)  PEQ,PYEQ,DVEL,VI,VL                       HYS00690
C                                                                       HYS00700
C------ LOAD AT OR ABOVE POSITIVE YEILD                                 HYS00710
      IF (RULE.EQ.1) THEN                                               HYS00720
         IF (DIR.GT.0) THEN                                             HYS00730
C           ... POSITIVE YEILD ...                                      HYS00740
            CALL STRENG (ESE,PSE,PSEOLD, PY,FL, DY,DL,S,S )             HYS00750
            FL=PY                                                       HYS00760
            DL=DY                                                       HYS00770
            GO TO 210                                                   HYS00780
         ELSE                                                           HYS00790
C           ... NEGATIVE YEILD ...                                      HYS00800
            CALL STRENG (ESE,PSE,PSEOLD,-PY,FL,-DY,DL,S,S )             HYS00810
            FL=-PY                                                      HYS00820
            DL=-DY                                                      HYS00830
            GO TO 220                                                   HYS00840
         ENDIF                                                          HYS00850
      ELSE                                                              HYS00860
         IF (RULE.EQ.2.10) GO TO 210                                    HYS00870
         IF (RULE.EQ.2.20) GO TO 220                                    HYS00880
         IF (RULE.EQ.2.30) GO TO 230                                    HYS00890
      ENDIF                                                             HYS00900
      WRITE (6,*) 'INVALID RULE IN HYST07-BILINEAR HYSTERESIS MODEL'    HYS00910
      CALL SDUMP                                                        HYS00920
C                                                                       HYS00930
C------ LOAD AT OR ABOVE POSITIVE YEILD, ON BACKBONE CURVE              HYS00940
  210 IF (DIR.LE.0) GO TO 230                                           HYS00950
      K=KIE                                                             HYS00960
      RULE=2.10                                                         HYS00970
      FC=PYEQ+DC*KIE                                                    HYS00980
      B= FC                                                             HYS00990
      A= FC*10                                                          HYS01000
      CALL STRENG (ESE,PSE,PSEOLD,FC,FL,DC,DL,K,S )                    HYS01010
      KEY=1                                                             HYS01020
         RKEY=KEY                                                       HYS01030
      IF (DVEL.LT.0 ) THEN                                              HYS01040
           K=S                                                          HYS01050
           B= FC                                                        HYS01060
           A= FC                                                        HYS01070
      ENDIF                                                             HYS01080
      GO TO 1000                                                        HYS01090
C                                                                       HYS01100
C------ LOAD AT OR BELOW NEGATIVE YEILD, ON BACKBONE CURVE              HYS01110
  220 IF (DIR.GE.0) GO TO 230                                           HYS01120
      K=KIE                                                             HYS01130
      RULE=2.20                                                         HYS01140
      FC=-PYEQ+DC*KIE                                                   HYS01150
      B= FC                                                             HYS01160
      A= FC*10                                                          HYS01170
      CALL STRENG (ESE,PSE,PSEOLD,FC,FL,DC,DL,K,S )                    HYS01180
      KEY=2                                                             HYS01190
         RKEY=KEY                                                       HYS01200
      IF (DVEL.LT.0 ) THEN                                              HYS01210
           K=S                                                          HYS01220
           B= FC                                                        HYS01230
           A= FC                                                        HYS01240
      ENDIF                                                             HYS01250
      GO TO 1000                                                        HYS01260
C                                                                       HYS01270
C------ LOAD IN LINEAR RANGE BETWEEN BACKBONE CURVES                    HYS01280
  230 K=S                                                               HYS01290
      RULE=2.3                                                          HYS01300
      FC=S*(DC-DL) + FL                                                 HYS01310
      A= FC + PYEQ - PEQ                                                HYS01320
      B= FC - PYEQ - PEQ                                                HYS01330
      IF (FC.GT.A) THEN                                                 HYS01340
         DA=DC+(A-FC)/S                                                 HYS01350
         CALL STRENG (ESE,PSE,PSEOLD, A,FL,DA,DL,K,S )                  HYS01360
         DL=DA                                                          HYS01370
         FL=A                                                           HYS01380
         GO TO 210                                                      HYS01390
      ELSE IF (FC.LT.B) THEN                                            HYS01400
         DB=DC+(B-FC)/S                                                 HYS01410
         CALL STRENG (ESE,PSE,PSEOLD, B,FL,DB,DL,K,S )                  HYS01420
         DL=DB                                                          HYS01430
         FL=B                                                           HYS01440
         GO TO 220                                                      HYS01450
      ENDIF                                                             HYS01460
C                                                                       HYS01470
C........ CALC D & E'S BASED ON PEAK STRAIN ENERGY                      HYS01480
      IF (KEY.EQ.1) CALL DNE(1,UPOS,EXCR, DY,DL,ESE(2),PSE,PSEOLD,CSE)  HYS01490
      IF (KEY.EQ.2) CALL DNE(1,UNEG,EXCR,-DY,DL,ESE(3),PSE,PSEOLD,CSE)  HYS01500
C                                                                       HYS01510
C........ CALC STRAIN ENERGY FOR THIS STEP..                            HYS01520
C     PSEL=PSE                                                          HYS01530
      CALL STRENG (ESE,PSE,PSEOLD,FC,FL,DC,DL,K,S )                     HYS01540
C                                                                       HYS01550
C........ CALC PLACTIC STRAIN ENERGY AT ZERO CROSSING ...               HYS01560
C     IF (FC*FL.LE.0) THEN                                              HYS01570
C        DF=FL-FC                                                       HYS01580
C        IF (DF.NE.0) PSEOLD=PSE+ (PSEL-PSE)*FC/(FL-FC)                 HYS01590
C        IF (DF.EQ.0) PSEOLD=PSE                                        HYS01600
C     ENDIF                                                             HYS01610
C                                                                       HYS01620
      KEY=3                                                             HYS01630
         RKEY=KEY                                                       HYS01640
C                                                                       HYS01650
C                                                                       HYS01660
 1000 IF (PRINT) WRITE (6,1011) FC,K,A,B                                HYS01670
 1011 FORMAT                                                            HYS01680
     &(' EL-END-FC=',F8.3,'     K=',F8.3,'     A=',F8.3,'     B=',F8.3)  HYS01690
      IF (PRINT) WRITE (6,1010) FC,DC,FL,DL,K,RULE,VI,VL,S,PY,KIE,      HYS01700
     &    A,B,KEY,ESE,PSE,PSEOLD,UPOS,UNEG,EXCR                         HYS01710
      RETURN                                                            HYS01720
      END                                                               HYS01730
@PROCESS SDUMP GOSTMT XREF                                              HYS00010
C********** HYST08 ********************************************************HYS00020
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    HYS00030
C           END DEBUG                                                   HYS00040
C                                                                       HYS00050
C     GOEL TRUSS HYSTERESIS MODEL FOR ANGLE AND BOX SECTION             HYS00060
C                                                                       HYS00070
      SUBROUTINE HYST08(P,PLA,DB,DELA,V,VLA,EL,SLK,                     HYS00080
     &                  EM,AREA,RADG,YS,PY, SHAPE,                      HYS00090
     &                  CC,CYCLE,RULE,STIF,IRV,IDUC,                    HYS00100
     &                  HA,VA,HC,VC,RSN,DEMAX,REGIN,PREGIN,             HYS00110
     &                  PBOT,DBOT,PTOP,DTOP,PRINT,                      HYS00120
     &                  EKI, DUC, EXCR, DELY, ESE, PSE, PSEOLD, CSE)    HYS00130
      COMMON /IDSTEP/ ISTEP                                             HYS00140
      DIMENSION DUC(3), EXCR(6), ESE(3)                                 HYS00150
      LOGICAL REVRSC,PRINT                                              HYS00160
      REAL*4 IRV, IDUC                                                  HYS00170
      IF( IRV .EQ. 0.) REVRSC=.FALSE.                                   HYS00180
      IF( IRV .EQ. 1.) REVRSC=.TRUE.                                    HYS00190
      PI=3.14159926                                                     HYS00200
      ES=AREA*EM/EL                                                     HYS00210
      DELY=PY/ES                                                        HYS00220
      DEL=DE/DELY                                                       HYS00230
      PL=P/PY                                                           HYS00240
      ESR=SLK*EL/RADG                                                   HYS00250
      SLOP7=STIF/ES                                                     HYS00260
      DA=ABS(DE)                                                        HYS00270
C                                                                       HYS00280
      DP=P-PLA                                                          HYS00290
      DDE=DE-DELA                                                       HYS00300
      DDEL=DDE/DELY                                                     HYS00310
C ---- CALCULATE EXCURSION RATE AT EVERY ZERO CROSSING ---------------- HYS00320
C                                                                       HYS00330
      IF( P*PLA .LE. 0 ) THEN                                           HYS00340
         CALL DNE(0, DUC,EXCR,DELY, DA,ESE,PSE,PSEOLD,CSE)              HYS00350
      ENDIF                                                             HYS00360
C                                                                       HYS00370
C**************************************************************************HYS00380
C   DUC     ---- ELEMENT DUCTILITY RATE                                HYS00390
C   EXCR    ---- ELEMENT EXCURSION RATE                                HYS00400
C   DELY    ---- ELEMENT YIELDING DISPLACEMENT                         HYS00410
C   ESE     ---- ELEMENT ELASTIC STRAIN ENERGY                         HYS00420
C   PSE     ---- ELEMENT PLASTIC STRAIN ENERGY                         HYS00430
C   PSEOLD ---- ELEMENT PLASTIC STRAIN ENERGY AT LAST ZERO CROSSING    HYS00440
C   CSE     ---- ELEMENT CONSTANT STRAIN ENERGY                        HYS00450
C                                                                       HYS00460
C   DP      ---- ELEMENT AXIAL LOAD INCREMENT (KIPS)                   HYS00470
C   ESR     ---- EFFECTIVE SLENDER RATIO  (KL/R)                       HYS00480
C   YS      ---- YIELDING STRESS                                       HYS00490
C   AREA    ---- CROSS SECTION AREA (A)                                HYS00500
C   EM      ---- ELASTIC MODULUS  (E)                                  HYS00510
C   DP      ---- ELEMENT AXIAL LOAD INCREMENT (KIPS)                   HYS00520
C   P       ---- CURRENT AXIAL LOAD (KIPS)                             HYS00530
C   PL      ---- P - PY RATIO (P/PY)                                   HYS00540
C   PLA     ---- LAST AXIAL LOAD                                       HYS00550
C   DDEL    ---- DISPLACEMENT INCREMENT  (IN)                          HYS00560
C   DE      ---- CURRENT AXIAL DISPLACEMENT  (IN)                      HYS00570
C   DEMAX   ---- MAXIMUM AXIAL DISPLACEMENT  (IN)                      HYS00580
C   DA      ---- CURRENT ABSOLUTE AXIAL DISPLACEMENT  (IN)             HYS00590
C   DELA    ---- LAST AXIAL DISPLACEMENT                               HYS00600
C   DDE     ---- ELEMENT DISPLACEMENT INCREMENT                        HYS00610
C   DEL     ---- DEL=DE/DELY                                           HYS00620
C   DELY    ---- YIELDING DISPLACEMENT                                 HYS00630
C   V       ---- CURRENT VELOCITY                                      HYS00640
C   VLA     ---- LAST VELOCITY                                         HYS00650
C   ES      ---- ELEMENT ELASTIC STIFFNESS                             HYS00660
C   EKI     ---- UNLOADING EQUIVALENT STIFFNESS FOR CALCULATING EESE   HYS00670
C   PY      ---- TENSION YIELDING LOAD                                 HYS00680
C   EL      ---- ELEMENT LENGTH  (IN)                                  HYS00690
C   CC      ---- C SUB. C CONSTANT                                     HYS00700
C   PMAX    ---- BUCKING LOAD (KIPS)                                   HYS00710
C   DELMAX ---- BUCKING DISPLACEMENT (IN)                              HYS00720
C   HA,HB,..,HC1,HD1..,HF ---- P/PY - DE/DELY PLOT'S DE/DELY COORDINATE HYS00730
C   VA,VB,..,VC1,VD1..,VF ---- P/PY - DE/DELY PLOT'S P/PY COORDINATE   HYS00740
C   RSN     ---- RESIDUAL STRAIN                                       HYS00750
C   RULE  ---- RULE NUMBER                                             HYS00760
C   CYCLE   ---- NO. OF CYCLE                                          HYS00770
C   STIF    ---- ELEMENT STIFFNESS                                     HYS00780
C   REVRSC ---- LOGICAL INDEX FOR REVERSING CURVE OCCURRED             HYS00790
C   IDUC    ---- 1: HC LESS THAN 12,  0:HC=12                          HYS00800
C   REGIN  ---- REGIN 1: DE >  HA                                      HYS00810
C                 2: DE = (HH, HA)                                     HYS00820
C                 3: DE = (HB, HH)                                     HYS00830
C                 4: DE = (HC, HB)                                     HYS00840
C   PREGIN ---- PREVIOUS STEP REGIN ID NUMBER                          HYS00850
C   SHAPE  ---- 1: FOR BOX OR ANGLE SECTION                            HYS00860
C               2: FOR I SHAPE SECTION                                 HYS00870
C               3: FOR BOX SECTION CONSIDERING LOCAL BUCKLING          HYS00880
C                  NOTE: THIS IS ONLY FOR KL/R LESS THAN 40            HYS00890
C**************************************************************************HYS00900
C                                                                       HYS00910
C INITIAL BUCKLING LOAD                                                 HYS00920
C                                                                       HYS00930
      IF( ESR .LT. CC .AND. CYCLE .EQ. 1 ) THEN                         HYS00940
         PMAX=-(1.-(ESR**2/(2.*CC**2)))*YS*AREA                         HYS00950
      ELSE IF( ESR .GE. CC .AND. CYCLE .EQ. 1 ) THEN                    HYS00960
         PMAX=-PI*PI*AREA*EM/(ESR**2)                                   HYS00970
      ENDIF                                                             HYS00980
C                                                                       HYS00990
C RESIDUAL ELOGATION FACTOR RFAC                                        HYS01000
C                                                                       HYS01010
C                    /** FOR BOX AND ANGLE SECTION **/                 HYS01020
      IF(SHAPE .EQ. 1) THEN                                             HYS01030
         RFAC=1.75                                                      HYS01040
      ELSE IF(SHAPE .EQ. 3) THEN                                        HYS01050
         RFAC=0                                                         HYS01060
C                    /** FOR I SHAPE SECTION        **/                HYS01070
      ELSE IF(SHAPE .EQ. 2) THEN                                        HYS01080
         RFAC=1.4                                                       HYS01090
      ENDIF                                                             HYS01100
C                                                                       HYS01110
C GENERATE CONTROL POINTS COORDINATE                                    HYS01120
C                                                                       HYS01130
C  (1)  POINT A                                                         HYS01140
C                                                                       HYS01150
      IF( SHAPE .NE. 3 ) THEN                                           HYS01160
        IF( CYCLE .EQ. 1 ) THEN                                         HYS01170
           VA=PMAX/PY                                                   HYS01180
           HA=PMAX/(ES*DELY)                                            HYS01190
        ELSE IF( CYCLE .EQ. 2) THEN                                     HYS01200
          IF( ESR .LE. 40 ) THEN                                        HYS01210
            PMAX=-(23.4/ESR)*PY                                         HYS01220
          ELSE IF( ESR .GT. 40 ) THEN                                   HYS01230
            PMAX=-(30./ESR)*PY                                          HYS01240
          ENDIF                                                         HYS01250
        ELSE IF( CYCLE .GT. 2) THEN                                     HYS01260
         IF( ESR .LE. 40 ) THEN                                         HYS01270
           PMAX=-(18.5/ESR)*PY                                          HYS01280
         ELSE IF( ESR .GT. 40 ) THEN                                    HYS01290
           PMAX=-(25./ESR)*PY                                           HYS01300
         ENDIF                                                          HYS01310
        ENDIF                                                           HYS01320
      ELSE IF( SHAPE .EQ. 3 ) THEN                                      HYS01330
        IF( CYCLE .EQ. 1) THEN                                          HYS01340
           VA=PMAX/PY                                                   HYS01350
           HA=PMAX/(ES*DELY)                                            HYS01360
        ELSE IF( CYCLE .EQ. 2) THEN                                     HYS01370
           PMAX=PMAX                                                    HYS01380
        ELSE IF( CYCLE .EQ. 3) THEN                                     HYS01390
           PMAX=-(23.4/ESR)*PY                                          HYS01400
        ELSE IF( CYCLE .GT. 3) THEN                                     HYS01410
           PMAX=-(18.5/ESR)*PY                                          HYS01420
        ENDIF                                                           HYS01430
      ENDIF                                                             HYS01440
C                                                                       HYS01450
C  (2)  POINT B                                                         HYS01460
C                                                                       HYS01470
      HB=-5.                                                            HYS01480
      IF( ESR .LE. 40 ) THEN                                            HYS01490
        VB=-17./ESR                                                     HYS01500
      ELSE IF( ESR .GT. 40 ) THEN                                       HYS01510
C                    /** FOR BOX AND ANGLE SECTION **/                 HYS01520
        IF( SHAPE .EQ. 1 .OR. SHAPE .EQ. 3 ) THEN                       HYS01530
          VB=-18./ESR                                                   HYS01540
C                    /** FOR I SHAPE SECTION        **/                HYS01550
        ELSE IF( SHAPE .EQ. 2 ) THEN                                    HYS01560
          VB=-11.3/ESR                                                  HYS01570
        ENDIF                                                           HYS01580
      ENDIF                                                             HYS01590
C                                                                       HYS01600
C  (3)  POINT C                                                         HYS01610
C                                                                       HYS01620
      IF( IDUC .EQ. 0.) THEN                                            HYS01630
        HC=-12.                                                         HYS01640
        IF( ESR .LE. 40 ) THEN                                          HYS01650
          VC=-15./ESR                                                   HYS01660
        ELSE IF( ESR .GT. 40 ) THEN                                     HYS01670
C                    /** FOR BOX AND ANGLE SECTION **/                 HYS01680
          IF( SHAPE .EQ. 1 .OR. SHAPE .EQ. 3 ) THEN                     HYS01690
            VC=-12./ESR                                                 HYS01700
C                    /** FOR I SHAPE SECTION        **/                HYS01710
          ELSE IF( SHAPE .EQ. 2 ) THEN                                  HYS01720
            VC=-8.5/ESR                                                 HYS01730
          ENDIF                                                         HYS01740
        ENDIF                                                           HYS01750
      ENDIF                                                             HYS01760
```

```
C
C  (4)  POINT E
C
          HE=1.+(RSN*EL/DELT)
          VE=1.
C
C  (5)  POINT D1
C
      SLOPE=(VE-VC)/(HE-HC)
      HD1=(SLOPE*HC-VC)/(SLOPE+0.267949)
      VD1=-0.267949*HD1
C
C  (6)  POINT D
C
      IF( ESR .LE. 40 ) THEN
        HD=HD1*31./ESR
        VD=VD1*31./ESR
      ELSE IF( ESR .GT. 40 ) THEN
C                    /** FOR BOX AND ANGLE SECTION **/
        IF( SHAPE .EQ. 1 .OR. SHAPE .EQ. 3 ) THEN
          HD=HD1*60./ESR
          VD=VD1*60./ESR
C                    /** FOR I SHAPE SECTION        **/
        ELSE IF( SHAPE .EQ. 2 ) THEN
          HD=HD1*20./ESR
          VD=VD1*20./ESR
        ENDIF
      ENDIF
C
C DEFINE ENVELOPE SLOPE
C
      SLOP1=1.
      SLOP2=(VB-VA)/(HB-HA)
      SLOP3=(VC-VB)/(HC-HB)
      SLOP4=(VD-VC)/(HD-HC)
      SLOP5=(VE-VD)/(HE-HD)
      SLOP6=0.0001
C
C DEFINE ENVELOPE INTERMIDIATE POINTS
C
C  (1)  POINT H
C
      VH=(VB+SLOP2*(-VD+HD-HB))/(1.-SLOP2)
      HH=VH-VD+HD
C
C  (2)  POINT C1
C
      VC1=0.
      HC1=HC-(VC/SLOP4)
C
C  (3)  POINT (PREF,DREF) : CALCULATE IN RULE NO. 4
C
C
C  (4)  POINT (HREF,VREF) : CALCULATE IN RULE NO. 5
C
C--- DEFINE CURRENT REGIN ZONE NUMBER
      PREGIN=REGIN
      IF( DEL .GE. HA) REGIN=1
      IF( DEL .GE. HH .AND. DEL .LT. HA ) REGIN=2
      IF( DEL .GE. HB .AND. DEL .LT. HH ) REGIN=3
      IF( DEL .GE. HC .AND. DEL .LT. HB ) REGIN=4
C
      IF (PRINT) THEN
      WRITE(6,567)CYCLE,RULE,HA,VA,HB,VB,HC,VC,HC1,VC1,
     &        HD,VD,HD1,VD1,HH,VH,HE,VE
  567 FORMAT(5X,'GOEL HYSTERESIS ENVELOPE CONTROL POINTS....'/
     &       5X,'-----------------------------------------'/
     &       5X,'CYCLE     ----------',F10.4,/
     &       5X,'RULE      ----------',F10.4,//
     &       5X,'(HA, VA) ----------',1X,'( ',F10.4,',',F10.4,' )',/
     &       5X,'(HB, VB) ----------',1X,'( ',F10.4,',',F10.4,' )',/
     &       5X,'(HC, VC) ----------',1X,'( ',F10.4,',',F10.4,' )',/
     &       5X,'(HC1, VC1) --------',1X,'( ',F10.4,',',F10.4,' )',/
     &       5X,'(HD, VD) ----------',1X,'( ',F10.4,',',F10.4,' )',/
     &       5X,'(HD1, VD1) --------',1X,'( ',F10.4,',',F10.4,' )',/
     &       5X,'(HH, VH) ----------',1X,'( ',F10.4,',',F10.4,' )',/
     &       5X,'(HE, VE) ----------',1X,'( ',F10.4,',',F10.4,' )',/)
      ENDIF
C
C     GOEL'S HYSTORESIS LOOP RULES
C
C  (1) RULE 1
C
      IF( RULE .EQ. 1.) THEN
            IF( P. GE. PMAX .AND. P .LT. PT ) THEN
                  STIF=SLOP1*ES
                  EKI=SLOP1*ES
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ENDIF
            ELSE IF( P. GE. PT ) THEN
                  STIF=SLOP6
                  RULE=6
                  RSN=0
                  EKI=SLOP1*ES
            ELSE IF( P. LT. PMAX) THEN
C                 HA=DEL
C                 VA=PL
C                 SLOP2=(VB-VA)/(HB-HA)
C                 STIF=SLOP2*ES
C
                  DI=DE-HA*DELT
                  SLOP2=(VB-VA)/(HB-HA)
                  STIF=SLOP2*ES
                  P=PMAX+(STIF*DI)
                  PL=P/PT
C
                  RULE=2
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ENDIF
                  EKI=SLOP1*ES
            ENDIF
            CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)
            CALL DNE(1, DUC, EXCR, DELT, DA, ESE, PSE, PSEOLD, CSE )
      RETURN
      ENDIF
C
C  (2) RULE 2
C
      IF( RULE .EQ. 2.) THEN
          IF( DEL .GE. HH) THEN
                IF( DDE .LE. 0. ) THEN
                      IF(PREGIN .NE. REGIN ) REVRSC= .FALSE.
                      STIF=SLOP2*ES
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ELSE IF( DEMAX .GT. 0 ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ENDIF
C                 ------------------------------------------
                      IF(.NOT. REVRSC )THEN
                         EKI=SLOP1*ES
                      ELSE IF( REVRSC )THEN
                         EKI=ES*((PTOP-PL)/(DTOP-DEL))
                      ENDIF
C                 ------------------------------------------
                ELSE IF( DDE .GT. 0. )THEN
                   IF(.NOT. REVRSC )THEN
                      RULE=7
                      STIF=SLOP1*ES
C                 ------------------------------------------
                      PLA=PL-SLOP2*DDEL
                      DLA=DEL-DDEL
                      DBOT=DLA
                      PBOT=PLA
                      PL=PLA+SLOP1*DDEL
                      P=PL*PT
C
                      PTOP=VE+(SLOP5*(PBOT-VE-SLOP1*DBOT
     *                      +SLOP1*HE)/(SLOP5-SLOP1))
                      DTOP=(PBOT-VE+SLOP5*HE-SLOP1*DBOT)
     *                      /(SLOP5-SLOP1)
                      EKI=ES*SLOP1
                   ELSE IF( REVRSC ) THEN
                      RULE=12
                      DBOT=DEL
                      PBOT=P/PT
                      SLOP7=(PTOP-PBOT)/(DTOP-DBOT)
                      STIF=SLOP7*ES
                      EKI=ES*SLOP7
                   ENDIF
                ENDIF
          ELSE IF(DEL .LT. HH .AND. DEL .GE. HB ) THEN
                IF( DDE .LE. 0. ) THEN
                      IF(PREGIN .NE. REGIN ) REVRSC= .FALSE.
                      STIF=SLOP2*ES
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ELSE IF( DEMAX .GT. 0 ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ENDIF
C                 ------------------------------------------
                      IF(.NOT. REVRSC )THEN
                         EKI=SLOP1*ES
                      ELSE IF( REVRSC )THEN
                         EKI=ES*((PTOP-PL)/(DTOP-DEL))
                      ENDIF
C                 ------------------------------------------
                ELSE IF( DDE .GT. 0. )THEN
                    IF( .NOT. REVRSC ) THEN
                      RULE=8
                      STIF=SLOP1*ES
C                 ------------------------------------------
                      PLA=PL-SLOP2*DDEL
                      DLA=DEL-DDEL
                      DBOT=DLA
                      PBOT=PLA
                      PL=PLA+SLOP1*DDEL
                      P=PL*PT
C
                      PTOP=VD+(SLOP4*(PBOT-VD-SLOP1*DBOT
     *                      +SLOP1*HD)/(SLOP4-SLOP1))
                      DTOP=(PBOT-VD+SLOP4*HD-SLOP1*DBOT)
     *                      /(SLOP4-SLOP1)
                      EKI=ES*SLOP1
                   ELSE IF( REVRSC ) THEN
                      RULE=11
                      DBOT=DEL
                      PBOT=P/PT
                      PTOP=VD
                      DTOP=HD
                      SLOP7=(PTOP-PBOT)/(DTOP-DBOT)
                      STIF=SLOP7*ES
                      EKI=ES*SLOP7
                   ENDIF
                ENDIF
          ELSE IF( DEL .LT. HB) THEN
                IF(PREGIN .NE. REGIN ) REVRSC= .FALSE.
                RULE=3
                EKI=ES*SLOP1
          ENDIF
      IF( REVRSC ) IRV=1.
      IF ( .NOT. REVRSC) IRV=0.
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)
            IF( ABS(DEMAX) .LT. ABS(DE) ) THEN
              CALL DNE(1, DUC, EXCR, DELT, DA, ESE, PSE, PSEOLD, CSE )
              DEMAX=DE
            ENDIF
      RETURN
      ENDIF
C
C  (3) RULE 3
C
      IF( RULE .EQ. 3.) THEN
          IF( DEL .GT. HC) THEN
                IF( DDE .LE. 0. ) THEN
                      IF(PREGIN .NE. REGIN ) REVRSC= .FALSE.
                      STIF=SLOP3*ES
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ELSE IF( DEMAX .GT. 0 ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ENDIF
C                 ------------------------------------------
                      IF(.NOT. REVRSC )THEN
                         EKI=SLOP1*ES
                      ELSE IF( REVRSC )THEN
                         EKI=ES*((PTOP-PL)/(DTOP-DEL))
                      ENDIF
C                 ------------------------------------------
                ELSE IF( DDE .GT. 0. )THEN
                  IF( DEL .GT. -12. ) THEN
                    IF( .NOT. REVRSC )THEN
                      RULE=9
                      STIF=SLOP1*ES
C
                      PLA=PL-SLOP2*DDEL
                      DLA=DEL-DDEL
                      DBOT=DLA
                      PBOT=PLA
                      PL=PLA+SLOP1*DDEL
                      P=PL*PT
C
                      PTOP=VD+(SLOP4*(PBOT-VD-SLOP1*DBOT
     *                      +SLOP1*HD)/(SLOP4-SLOP1))
                      DTOP=(PBOT-VD+SLOP4*HD-SLOP1*DBOT)
     *                      /(SLOP4-SLOP1)
                      EKI=ES*SLOP1
                    ELSE IF( REVRSC ) THEN
                      RULE=10
                      DBOT=DEL
                      PBOT=P/PT
                      SLOP7=(PTOP-PBOT)/(DTOP-DBOT)
                      STIF=SLOP7*ES
                      EKI=ES*SLOP7
                    ENDIF
                  ELSE IF ( DEL .LE. -12. ) THEN
                     REVRSC= .FALSE.
                     RULE=4
                     STIF=SLOP4*ES
                     EKI=ES*SLOP4
                  ENDIF
                ENDIF
          ELSE IF( DEL .LE. HC ) THEN
           IF( DDE .LT. 0 ) THEN
             RULE=3
             HC=DEL
             VC=PL
             IDUC=1.
             STIF=-SLOP6*ES
             EKI=ES*((VD-VC)/(HD-HC))
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN
```

```
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC   HYS04610
                  ELSE IF( DEMAX .GT. 0 ) THEN                        HYS04620
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC   HYS04630
                  ENDIF                                               HYS04640
               ELSE IF( DDE .GT. 0 ) THEN                             HYS04650
                  RULE=4                                              HYS04660
                  STIF=SLOP4*ES                                       HYS04670
                  EKI=ES*SLOP4                                        HYS04680
               ENDIF                                                  HYS04690
            ENDIF                                                     HYS04700
         IF( REVRSC ) IRV=1.                                          HYS04710
         IF ( .NOT. REVRSC) IRV=0.                                    HYS04720
         CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)           HYS04730
               IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                     HYS04740
                  CALL DNE(1, DUC, EXCR, DELT, DA, ESE, PSE, PSEOLD, CSE ) HYS04750
                  DEMAX=DE                                            HYS04760
               ENDIF                                                  HYS04770
         RETURN                                                       HYS04780
         ENDIF                                                        HYS04790
C                                                                     HYS04800
C  (4) RULE 7                                                         HYS04810
C                                                                     HYS04820
         IF( RULE .EQ. 7.) THEN                                       HYS04830
            IF( DEL .GT. DBOT .AND. DEL .LT. DTOP .AND. P .GT. PMAX) THEN HYS04840
                  STIF=SLOP1*ES                                       HYS04850
                  EKI=ES*SLOP1                                        HYS04860
            ELSE IF(DEL .LT. DBOT .OR. P .LE. PMAX) THEN              HYS04870
               IF( P .LE. PMAX ) THEN                                 HYS04880
                     REVRSC= .FALSE.                                  HYS04890
                     PTOP=0                                           HYS04900
                     DTOP=0                                           HYS04910
                     PBOT=PL                                          HYS04920
                     DBOT=DEL                                         HYS04930
                     S1=PLA/PY                                        HYS04940
                     S2=PMAX/PY                                       HYS04950
                     HA=(DELA/DELY) + ((S2-S1)/(SLOP1))               HYS04960
                     VA=S2                                            HYS04970
                     SLOP2=(VB-VA)/(HB-HA)                            HYS04980
               ENDIF                                                  HYS04990
               STIF=SLOP2*ES                                          HYS05000
               RULE=2                                                 HYS05010
C                                                                     HYS05020
               PL=VA+ SLOP2*(DBOT-HA)                                 HYS05030
               P=PL*PY                                                HYS05040
C                                                                     HYS05050
               EKI=ES*((PTOP-PL)/(DTOP-DEL))                          HYS05060
            ELSE IF(DEL .GT. DTOP .AND. PL .LT. VE) THEN              HYS05070
               STIF=SLOP5*ES                                          HYS05080
               RULE=5                                                 HYS05090
               EKI=ES*SLOP1                                           HYS05100
            ELSE IF(DEL .GT. DTOP .AND. PL .GE. VE) THEN              HYS05110
               STIF=SLOP6*ES                                          HYS05120
               RULE=6                                                 HYS05130
               EKI=ES*SLOP1                                           HYS05140
            ENDIF                                                     HYS05150
         IF( REVRSC ) IRV=1.                                          HYS05160
         IF ( .NOT. REVRSC) IRV=0.                                    HYS05170
         CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)           HYS05180
C        CALL DNE(1, DUC, EXCR, DELT, DA, ESE, PSE, PSEOLD, CSE )     HYS05190
         RETURN                                                       HYS05200
         ENDIF                                                        HYS05210
C                                                                     HYS05220
C  (5) RULE 8                                                         HYS05230
C                                                                     HYS05240
         IF( RULE .EQ. 8.) THEN                                       HYS05250
            IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN              HYS05260
               STIF=SLOP1*ES                                          HYS05270
               EKI=ES*SLOP1                                           HYS05280
            ELSE IF(DEL .LT. DBOT ) THEN                              HYS05290
               STIF=SLOP2*ES                                          HYS05300
               RULE=2                                                 HYS05310
C                                                                     HYS05320
               PL=PBOT-(DBOT-DEL)*SLOP2                               HYS05330
               P=PL*PY                                                HYS05340
C                                                                     HYS05350
               EKI=ES*((PTOP-PL)/(DTOP-DEL))                          HYS05360
            ELSE IF(DEL .GT. DTOP ) THEN                              HYS05370
               STIF=SLOP4*ES                                          HYS05380
               RULE=4                                                 HYS05390
               EKI=ES*SLOP1                                           HYS05400
            ENDIF                                                     HYS05410
         IF( REVRSC ) IRV=1.                                          HYS05420
         IF ( .NOT. REVRSC) IRV=0.                                    HYS05430
         CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)           HYS05440
C        CALL DNE(1, DUC, EXCR, DELT, DA, ESE, PSE, PSEOLD, CSE )     HYS05450
         RETURN                                                       HYS05460
         ENDIF                                                        HYS05470
C                                                                     HYS05480
C  (6) RULE 9                                                         HYS05490
C                                                                     HYS05500
         IF( RULE .EQ. 9.) THEN                                       HYS05510
            IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN              HYS05520
               STIF=SLOP1*ES                                          HYS05530
               EKI=ES*SLOP1                                           HYS05540
            ELSE IF(DEL .LT. DBOT ) THEN                              HYS05550
               STIF=SLOP3*ES                                          HYS05560
               RULE=3                                                 HYS05570
C                                                                     HYS05580
               PL=PBOT-(DBOT-DEL)*SLOP3                               HYS05590
               P=PL*PY                                                HYS05600
C                                                                     HYS05610
               EKI=ES*((PTOP-PL)/(DTOP-DEL))                          HYS05620
            ELSE IF(DEL .GT. DTOP ) THEN                              HYS05630
               STIF=SLOP4*ES                                          HYS05640
               RULE=4                                                 HYS05650
               EKI=ES*SLOP1                                           HYS05660
            ENDIF                                                     HYS05670
         IF( REVRSC ) IRV=1.                                          HYS05680
         IF ( .NOT. REVRSC) IRV=0.                                    HYS05690
         CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)           HYS05700
C        CALL DNE(1, DUC, EXCR, DELT, DA, ESE, PSE, PSEOLD, CSE )     HYS05710
         RETURN                                                       HYS05720
         ENDIF                                                        HYS05730
C                                                                     HYS05740
C  (7) RULE 4                                                         HYS05750
C                                                                     HYS05760
         IF( RULE .EQ. 4.) THEN                                       HYS05770
            IF (DEL .GE. HC1 .AND. DEL .LT. HD ) THEN                 HYS05780
               IF(DDE .GE. 0. ) THEN                                  HYS05790
                  STIF=SLOP4*ES                                       HYS05800
                  EKI=ES*SLOP1                                        HYS05810
               ELSE IF( DDE .LT. 0. ) THEN                            HYS05820
                  PREF=VD+(SLOP4*(VB-VD-SLOP1*HB                      HYS05830
     *                 +SLOP1*HD)/(SLOP4-SLOP1))                    HYS05840
                  DREF=(VB-VD+SLOP4*HD-SLOP1*HB)/(SLOP4-SLOP1)        HYS05850
                  IF( DEL .LT. DREF) THEN                             HYS05860
                     PTOP=P/PY                                        HYS05870
                     DTOP=DEL                                         HYS05880
                     DBOT=(SLOP3*HB-VB+PTOP-SLOP1*DEL)/(SLOP3-SLOP1)  HYS05890
                     PBOT=VB+SLOP3*(DBOT-HB)                          HYS05900
                     STIF=SLOP1*ES                                    HYS05910
                     RULE=9                                           HYS05920
                     REVRSC= .TRUE.                                   HYS05930
                     EKI=ES*SLOP1                                     HYS05940
                  ELSE IF( DEL .GE. DREF) THEN                        HYS05950
                     PTOP=P/PY                                        HYS05960
                     DTOP=DEL                                         HYS05970
                     DBOT=(SLOP2*HA-VA+PTOP-SLOP1*DEL)/(SLOP2-SLOP1)  HYS05980
                     PBOT=VA+SLOP2*(DBOT-HA)                          HYS05990
                     STIF=SLOP1*ES                                    HYS06000
                     RULE=8                                           HYS06010
                     REVRSC= .TRUE.                                   HYS06020
                     EKI=ES*SLOP1                                     HYS06030
                  ENDIF                                               HYS06040
               ENDIF                                                  HYS06050
            ELSE IF( DEL .GT. HD .AND. DDE .GT. 0. ) THEN             HYS06060
               STIF=SLOP5*ES                                          HYS06070
               RULE=5                                                 HYS06080
               EKI=ES*SLOP1                                           HYS06090
            ELSE IF( DEL .LT. HC1 .AND. DEL .GE. HC ) THEN            HYS06100
               REVRSC= .FALSE.                                        HYS06110
               RULE=4                                                 HYS06120
               STIF=SLOP4*ES                                          HYS06130
               EKI=ES*SLOP4                                           HYS06140
            ELSE IF( DEL .LT. HC ) THEN                               HYS06150
               REVRSC= .FALSE.                                        HYS06160
               RULE=3                                                 HYS06170
               HC=DEL                                                 HYS06180
               VC=PL                                                  HYS06190
               IDUC=1.                                                HYS06200
               STIF=-SLOP6*ES                                         HYS06210
               EKI=ES*((VD-VC)/(HD-HC))                               HYS06220
            ENDIF                                                     HYS06230
         IF( REVRSC ) IRV=1.                                          HYS06240
         IF ( .NOT. REVRSC) IRV=0.                                    HYS06250
         CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)           HYS06260
               IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                     HYS06270
                  CALL DNE(1, DUC, EXCR, DELT, DA, ESE, PSE, PSEOLD, CSE ) HYS06280
                  DEMAX=DE                                            HYS06290
               ENDIF                                                  HYS06300
         RETURN                                                       HYS06310
         ENDIF                                                        HYS06320
C                                                                     HYS06330
C  (8) RULE 5                                                         HYS06340
C                                                                     HYS06350
         IF( RULE .EQ. 5.) THEN                                       HYS06360
            IF (DEL .LT. HE ) THEN                                    HYS06370
               IF(DDE .GE. 0. ) THEN                                  HYS06380
                  STIF=SLOP5*ES                                       HYS06390
                  EKI=ES*SLOP1                                        HYS06400
               ELSE IF( DDE .LT. 0. ) THEN                            HYS06410
                     BREF=VA-SLOP1*HA                                 HYS06420
                     HREF=(VD-SLOP5*HD-BREF)/(SLOP1-SLOP5)            HYS06430
                     IF( DEL .GE. HREF ) CYCLE=CYCLE+1                HYS06440
                     PTOP=P/PY                                        HYS06450
                     DTOP=DEL                                         HYS06460
                     DBOT=(SLOP2*HA-VA+PTOP-SLOP1*DEL)/(SLOP2-SLOP1)  HYS06470
                     PBOT=VA+SLOP2*(DBOT-HA)                          HYS06480
                     STIF=SLOP1*ES                                    HYS06490
                     RULE=7                                           HYS06500
                     REVRSC= .TRUE.                                   HYS06510
                     EKI=ES*SLOP1                                     HYS06520
               ENDIF                                                  HYS06530
            ELSE IF( DEL .GE. HE .AND. DDE .GT. 0. ) THEN             HYS06540
               STIF=SLOP6                                             HYS06550
               RULE=6                                                 HYS06560
               EKI=ES*SLOP1                                           HYS06570
            ENDIF                                                     HYS06580
         IF( REVRSC ) IRV=1.                                          HYS06590
         IF ( .NOT. REVRSC) IRV=0.                                    HYS06600
         CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)           HYS06610
               IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                     HYS06620
                  CALL DNE(1, DUC, EXCR, DELT, DA, ESE, PSE, PSEOLD, CSE ) HYS06630
                  DEMAX=DE                                            HYS06640
               ENDIF                                                  HYS06650
         RETURN                                                       HYS06660
         ENDIF                                                        HYS06670
C                                                                     HYS06680
C  (9) RULE 6                                                         HYS06690
C                                                                     HYS06700
         IF( RULE .EQ. 6.) THEN                                       HYS06710
            IF(DDE .GE. 0.) THEN                                      HYS06720
               STIF=SLOP6                                             HYS06730
               EKI=ES*SLOP1                                           HYS06740
            ELSE IF(DDE .LT. 0.) THEN                                 HYS06750
               STIF=SLOP1*ES                                          HYS06760
               RULE=1                                                 HYS06770
               CYCLE=CYCLE+1.                                         HYS06780
               REVRSC= .FALSE.                                        HYS06790
               EKI=ES*SLOP1                                           HYS06800
            ENDIF                                                     HYS06810
         IF( REVRSC ) IRV=1.                                          HYS06820
         IF ( .NOT. REVRSC) IRV=0.                                    HYS06830
         CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)           HYS06840
               IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                     HYS06850
                  CALL DNE(1, DUC, EXCR, DELT, DA, ESE, PSE, PSEOLD, CSE ) HYS06860
                  DEMAX=DE                                            HYS06870
               ENDIF                                                  HYS06880
         RETURN                                                       HYS06890
         ENDIF                                                        HYS06900
C                                                                     HYS06910
C  (10) RULE 10                                                       HYS06920
C                                                                     HYS06930
         IF( RULE .EQ. 10.) THEN                                      HYS06940
            IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN              HYS06950
               STIF=SLOP7*ES                                          HYS06960
               EKI=ES*SLOP7                                           HYS06970
            ELSE IF(DEL .LT. DBOT ) THEN                              HYS06980
               STIF=SLOP3*ES                                          HYS06990
               RULE=7                                                 HYS07000
               REVRSC= .FALSE.                                        HYS07010
               PTOP=0                                                 HYS07020
               DTOP=0                                                 HYS07030
               PBOT=0                                                 HYS07040
               DBOT=0                                                 HYS07050
               EKI=ES*((PTOP-PL)/(DTOP-DEL))                          HYS07060
            ELSE IF(DEL .GT. DTOP ) THEN                              HYS07070
               STIF=SLOP4*ES                                          HYS07080
               RULE=4                                                 HYS07090
               EKI=ES*SLOP1                                           HYS07100
            ENDIF                                                     HYS07110
         IF( REVRSC ) IRV=1.                                          HYS07120
         IF ( .NOT. REVRSC) IRV=0.                                    HYS07130
         CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)           HYS07140
C        CALL DNE(1, DUC, EXCR, DELT, DA, ESE, PSE, PSEOLD, CSE )     HYS07150
         RETURN                                                       HYS07160
         ENDIF                                                        HYS07170
C                                                                     HYS07180
C  (11) RULE 11                                                       HYS07190
C                                                                     HYS07200
         IF( RULE .EQ. 11.) THEN                                      HYS07210
            IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN              HYS07220
               STIF=SLOP7*ES                                          HYS07230
               EKI=ES*SLOP7                                           HYS07240
            ELSE IF(DEL .LT. DBOT ) THEN                              HYS07250
               STIF=SLOP2*ES                                          HYS07260
               RULE=2                                                 HYS07270
               REVRSC= .FALSE.                                        HYS07280
               PTOP=0                                                 HYS07290
               DTOP=0                                                 HYS07300
               PBOT=0                                                 HYS07310
               DBOT=0                                                 HYS07320
               EKI=ES*((VD-PL)/(HD-DEL))                              HYS07330
            ELSE IF(DEL .GT. DTOP ) THEN                              HYS07340
               STIF=SLOP5*ES                                          HYS07350
               RULE=5                                                 HYS07360
               EKI=ES*SLOP1                                           HYS07370
            ENDIF                                                     HYS07380
         IF( REVRSC ) IRV=1.                                          HYS07390
         IF ( .NOT. REVRSC) IRV=0.                                    HYS07400
         CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)           HYS07410
C        CALL DNE(1, DUC, EXCR, DELT, DA, ESE, PSE, PSEOLD, CSE )     HYS07420
         RETURN                                                       HYS07430
         ENDIF                                                        HYS07440
```

```
C                                                                       HYS07450
C  (12) RULE 12                                                         HYS07460
C                                                                       HYS07470
      IF( RULE .EQ. 12.) THEN                                           HYS07480
          IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                  HYS07490
              STIF=SLOP7*ES                                             HYS07500
              EKI=ES*SLOP7                                              HYS07510
          ELSE IF(DEL .LT. DBOT ) THEN                                  HYS07520
              STIF=SLOP2*ES                                             HYS07530
              RULE=2                                                    HYS07540
              REVRSC= .FALSE.                                           HYS07550
              PTOP=0                                                    HYS07560
              DTOP=0                                                    HYS07570
              PBOT=0                                                    HYS07580
              DBOT=0                                                    HYS07590
              EKI=ES*((PTOP-PL)/(DTOP-DEL))                             HYS07600
          ELSE IF(DEL .GT. DTOP ) THEN                                  HYS07610
              STIF=SLOP5*ES                                             HYS07620
              RULE=5                                                    HYS07630
              EKI=ES*SLOP1                                              HYS07640
          ENDIF                                                         HYS07650
      ENDIF                                                             HYS07660
      IF( REVRSC ) IRV=1.                                               HYS07670
      IF ( .NOT. REVRSC) IRV=0.                                         HYS07680
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)                HYS07690
C     CALL DNE(1, DUC, EXCR, DELT, DA, ESE, PSE, PSEOLD, CSE )          HYS07700
      RETURN                                                            HYS07710
      END                                                               HYS07720
C******* HYST09 ***************************************************** HYS00010
C     DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                          HYS00020
C     END DEBUG                                                         HYS00030
C                                                                       HYS00040
C     PINO-SUAREZ TOWER LONG DIRECTION GIRDER MOMENT-ROTATION           HYS00050
C     HYSTERESIS MODEL BASED ON HYSTERESIS LOOP OF T2A GIRDER           HYS00060
C                                                                       HYS00070
      SUBROUTINE HYST09(MOPT,STIF,P,X,PLA,XLA,DP,DX,                    HYS00080
     &            RV,RULE,PTOP,DTOP,PBOT,DBOT,                          HYS00090
     &            SPTOP,SDTOP,SPBOT,SDBOT,                              HYS00100
     &            SSPTOP,SSDTOP,SSPBOT,SSDBOT,                          HYS00110
     &            HA,VA,HB,VB,HD,VD,HE,VE,HAP,VAP,HBP,VBP,              HYS00120
     &            SSHD,SSVD,SSHE,SSVE,                                  HYS00130
     &            SSSPTP,SSSDTP,SSSPBT,SSSDBT,                          HYS00140
     &            SHD,SVD,SHE,SVE,PRINT,                                HYS00150
     &            EKI,DUC,EXCR,DELT,ESE,PSE,PSEOLD,CSE,                 HYS00160
     &            DEMAX,FACH,FACV,FAC,DUCMAX)                           HYS00170
      DIMENSION DUC(3),EXCR(6),ESE(3)                                   HYS00180
      LOGICAL HIGH,PRINT                                                HYS00190
C                                                                       HYS00200
C ***************************************************************** HYS00210
C   DUC    ---- ELEMENT DUCTILITY RATE                                 HYS00220
C   EXCR   ---- ELEMENT EXCURSION RATE                                 HYS00230
C   ESE    ---- ELEMENT ELASTIC STRAIN RATE                           HYS00240
C   PSE    ---- ELEMENT PLASTIC STRAIN RATE                           HYS00250
C   PSEOLD ---- ELEMENT PLASTIC STRAIN RATE AT LAST ZERO CROSSING     HYS00260
C   CSE    ---- ELEMENT CONSTANT STRAIN ENERGY                        HYS00270
C   STIF   ---- ELEMENT STIFFNESS                                     HYS00280
C   P      ---- CURRENT ELEMENT INTERNAL FORCE                        HYS00290
C   X      ---- CURRENT ELEMENT INTERNAL DISPLACEMENT                 HYS00300
C   PLA    ---- PREVIOUS ELEMENT INTERNAL FORCE                       HYS00310
C   XLA    ---- PREVIOUS ELEMENT INTERNAL DISPLACEMENT                HYS00320
C   DP     ---- ELEMENT INTERNAL FORCE INCREMENT                      HYS00330
C   DX     ---- ELEMENT INTERNAL DISPLACEMENT INCREMENT               HYS00340
C   HIGH   ---- LOGICAL INDEX, IF P .GT. 0 THEN HIGH=.TRUE.          HYS00350
C   RULE   ---- RULE NUMBER                                           HYS00360
C   PTOP   ---- REVERSED LOADING POINT AT TOP ENVELOPE                HYS00370
C   DTOP   ---- REVERSED DISP. POINT AT TOP ENVELOPE                  HYS00380
C   PBOT   ---- REVERSED LOADING POINT AT BOTTOM ENVELOPE             HYS00390
C   DBOT   ---- REVERSED DISP. POINT AT BOTTOM ENVELOPE               HYS00400
C   SPTOP  ---- REVERSED LOADING POINT AT TOP SMALL LOOP              HYS00410
C   SDTOP  ---- REVERSED DISP. POINT AT TOP SMALL LOOP                HYS00420
C   SPBOT  ---- REVERSED LOADING POINT AT BOTTOM SMALL LOOP           HYS00430
C   SDBOT  ---- REVERSED DISP. POINT AT BOTTOM SMALL LOOP             HYS00440
C   SSPTOP ---- REVERSED LOADING POINT AT TOP SECOND SMALL LOOP       HYS00450
C   SSDTOP ---- REVERSED DISP. POINT AT TOP SECOND SMALL LOOP         HYS00460
C   SSPBOT ---- REVERSED LOADING POINT AT BOTTOM SECOND SMALL LOOP    HYS00470
C   SSDBOT ---- REVERSED DISP. POINT AT BOTTOM SCOND SMALL LOOP       HYS00480
C   SSSPTP ---- REVERSED LOADING POINT AT TOP THIRD OR ABOVE SMALL LOOP HYS00490
C   SSSDTP ---- REVERSED DISP. POINT AT TOP THIRD OR ABOVE SMALL LOOP  HYS00500
C   SSSPBT ---- REVERSED LOADING POINT AT BOT. THIRD OR ABOVE SMALL LOO HYS00510
C   SSSDBT ---- REVERSED DISP. POINT AT BOT. THIRD OR ABOVE  SMALL LOOP HYS00520
C   HA,VA  ---- FIRST BUCKLING POINT AT TOP ENVELOPE                  HYS00530
C   HAP,VAP---- FIRST BUCKLING POINT AT BOTTOM ENVELOPE               HYS00540
C   HB,VB  ---- TOP ENVELOPE CONTROL POINT                            HYS00550
C   HBP,VBP---- BOTTOM ENVELOPE CONTROL POINT                         HYS00560
C   HD,VD  ---- FIRST SMALL LOOP CONTROL POINT                        HYS00570
C   HE,VE  ---- FIRST SMALL LOOP CONTROL POINT                        HYS00580
C   SHD,SVD --- SECOND SMALL LOOP CONTROL POINT                       HYS00590
C   SHE,SVE --- SECOND SMALL LOOP CONTROL POINT                       HYS00600
C   SSHD,SSVD - THIRD OR ABOVE SMALL LOOP CONTROL POINT               HYS00610
C   SSHE,SSVE - THIRD OR ABOVE SMALL LOOP CONTROL POINT               HYS00620
C   DUCMAX    - FAILURE DUCTILITY                                     HYS00630
C               .GE. 1 FAILURE DUCTILITY CONSIDERED                   HYS00640
C               .LT. 1 FAILURE DUCTILITY NOT CONSIDERED               HYS00650
C ***************************************************************** HYS00660
C                                                                       HYS00670
C ****************************                                      HYS00680
C * CHECK FAILURE DUCTILITY **                                      HYS00690
C ****************************                                      HYS00700
      IF( DUC(1) .GE. DUCMAX .AND. DUCMAX .GE. 1) THEN                  HYS00710
        P=0                                                             HYS00720
        STIF=0.001                                                      HYS00730
        RETURN                                                          HYS00740
      ENDIF                                                             HYS00750
C                                                                       HYS00760
      IF( RV .EQ. 1 ) HIGH=.FALSE.                                      HYS00770
      IF( RV .EQ. 0 ) HIGH=.TRUE.                                       HYS00780
C                                                                       HYS00790
C ***********************************                                   HYS00800
C * DEFINE ENVELOPE CONTROL POINTS *                                   HYS00810
C ***********************************                                   HYS00820
      HA1=0.0042                                                        HYS00830
      VA1=2700.                                                         HYS00840
      HB1=HA1+0.0023                                                    HYS00850
      VB1=VA1-465                                                       HYS00860
      HAP1=-0.0042                                                      HYS00870
      VAP1=-2700.                                                       HYS00880
      HBP1=HAP1-0.0023                                                  HYS00890
      VBP1=VAP1+465                                                     HYS00900
      IF( MOPT .EQ. 2 ) THEN                                            HYS00910
         PTOP=999999                                                    HYS00920
         PBOT=999999                                                    HYS00930
C                                                                       HYS00940
      FACH=HA/HA1                                                       HYS00950
      FACV=VA/VA1                                                       HYS00960
      FAC=FACV/FACH                                                     HYS00970
      HA=HA1*FACH                                                       HYS00980
      VA=VA1*FACV                                                       HYS00990
      HB=HB1*FACH                                                       HYS01000
      VB=VB1*FACV                                                       HYS01010
      HAP=HAP1*FACH                                                     HYS01020
      VAP=VAP1*FACV                                                     HYS01030
      HBP=HBP1*FACH                                                     HYS01040
      VBP=VBP1*FACV                                                     HYS01050
      ENDIF                                                             HYS01060
C                                                                       HYS01070
C     IF( P .GT. 0 ) HIGH=.TRUE.                                        HYS01080
C     IF( P .LT. 0 ) HIGH=.FALSE.                                       HYS01090
      IF( P .GE. 0 .AND. PLA .GE. 0 ) HIGH=.TRUE.                       HYS01100
      IF( P .LE. 0 .AND. PLA .LE. 0 ) HIGH=.FALSE.                      HYS01110
      IF( HIGH) RV=0                                                    HYS01120
      IF( .NOT. HIGH) RV=1                                              HYS01130
C                                                                       HYS01140
C ***********************                                             HYS01150
C * DEFINE ENVELOPE SLOPE *                                           HYS01160
C ***********************                                             HYS01170
      SLOP1=(VA1/HA1)*FAC                                               HYS01180
      SLOP2=-202324*FAC                                                 HYS01190
      SLOP3=-100000*FAC                                                 HYS01200
C                                                                       HYS01210
C     IF(ABS(X) .GT. 0.0126*FACH) SLOP3=SLOP3*0.01                      HYS01220
C                                                                       HYS01230
      IF (PRINT) THEN                                                   HYS01240
      WRITE(6,567)RULE,HA,VA,HB,VB,HAP,VAP,HBP,VBP,                     HYS01250
     &          SLOP1,SLOP2,SLOP3                                       HYS01260
  567 FORMAT(5X,'PINO-SUAREZ TOWER LONG DIR. M-R LOOP CONTROL POINTS..',HYS01270
     &       5X,'-----------------------------------------------',      HYS01280
     &       5X,'RULE    ----------',1X,G10.4,/                        HYS01290
     &       5X,'(HA, VA) ----------',1X,'( ',F10.4,',',F10.4,' )',/   HYS01300
     &       5X,'(HB, VB) ----------',1X,'( ',F10.4,',',F10.4,' )',/   HYS01310
     &       5X,'(HAP,VAP) ---------',1X,'( ',F10.4,',',F10.4,' )',/   HYS01320
     &       5X,'(HBP,VBP) ---------',1X,'( ',F10.4,',',F10.4,' )',/   HYS01330
     &       5X,'SLOP1   ----------',1X,G10.4,/                        HYS01340
     &       5X,'SLOP2   ----------',1X,G10.4,/                        HYS01350
     &       5X,'SLOP3   ----------',1X,G10.4,/)                       HYS01360
      ENDIF                                                             HYS01370
C                                                                       HYS01380
C ************************                                            HYS01390
C * HYSTERESIS LOOP RULES *                                           HYS01400
C ************************                                            HYS01410
C--RULE 1:                                                            HYS01420
      IF( RULE .EQ. 1 ) THEN                                            HYS01430
        IF(P .GE. VA .OR. P .LE. VAP ) THEN                             HYS01440
          RULE=2                                                        HYS01450
          STIF=SLOP2                                                    HYS01460
C         *** FOR UNBALANCE FORCE ***                                   HYS01470
          IF( P .GE. VA) THEN                                           HYS01480
              UP1= P-VA                                                 HYS01490
              UD = UP1/SLOP1                                            HYS01500
              UP2= -UD*SLOP2                                            HYS01510
              P  = P-(UP1+UP2)                                          HYS01520
              IF( P .LT. 0.1*VA) P=0.1*VA                               HYS01530
              RV=0                                                      HYS01540
          ELSE IF( P .LE. VAP) THEN                                     HYS01550
              UP1= P-VAP                                                HYS01560
              UD = UP1/SLOP1                                            HYS01570
              UP2= -UD*SLOP2                                            HYS01580
              P  = P-(UP1+UP2)                                          HYS01590
              IF( P .GT. 0.1*VAP) P=0.1*VAP                             HYS01600
              RV=1                                                      HYS01610
          ENDIF                                                         HYS01620
C                                                                       HYS01630
        ELSE                                                            HYS01640
          RULE=1                                                        HYS01650
          STIF=SLOP1                                                    HYS01660
        ENDIF                                                           HYS01670
        IF( DEMAX .LT. ABS(X) ) THEN                                    HYS01680
            DEMAX=ABS(X)                                                HYS01690
            DUC(1)=DEMAX/(HA1*FACH)                                     HYS01700
        ENDIF                                                           HYS01710
        RETURN                                                          HYS01720
      ENDIF                                                             HYS01730
C                                                                       HYS01740
C---RULE 2:                                                           HYS01750
      IF( RULE .EQ. 2 ) THEN                                            HYS01760
        IF( HIGH .AND. DX .GT. 0 ) THEN                                 HYS01770
          IF( X .LT. HB ) THEN                                          HYS01780
            RULE=2                                                      HYS01790
            STIF=SLOP2                                                  HYS01800
            IF( P .LT. 0 )P=0.1*VA                                      HYS01810
            IF( P .LE. 0.1*VA) STIF=SLOP3*0.01                          HYS01820
          ELSE IF( X .GE. HB ) THEN                                     HYS01830
            RULE=3                                                      HYS01840
            STIF=SLOP3                                                  HYS01850
            IF( P .LT. 0 )P=0.1*VA                                      HYS01860
            IF( P .LE. 0.1*VA) STIF=SLOP3*0.01                          HYS01870
C           *** FOR UNBALANCE FORCE ***                                 HYS01880
            IF( XLA .LT. HB) THEN                                       HYS01890
              UD=X-HB                                                   HYS01900
              UP=STIF*UD                                                HYS01910
              P=VB+UP                                                   HYS01920
            ENDIF                                                       HYS01930
C                                                                       HYS01940
          ENDIF                                                         HYS01950
        ELSE IF( .NOT. HIGH .AND. DX .LT. 0 ) THEN                      HYS01960
          IF( X .GT. HBP ) THEN                                         HYS01970
            RULE=2                                                      HYS01980
            STIF=SLOP2                                                  HYS01990
            IF( P .GT. 0 )P=0.1*VAP                                     HYS02000
            IF( P .GE. 0.1*VAP) STIF=SLOP3*0.01                         HYS02010
          ELSE IF( X .LE. HBP ) THEN                                    HYS02020
            RULE=3                                                      HYS02030
            STIF=SLOP3                                                  HYS02040
            IF( P .GT. 0 )P=0.1*VAP                                     HYS02050
            IF( P .GE. 0.1*VAP) STIF=SLOP3*0.01                         HYS02060
C           *** FOR UNBALANCE FORCE ***                                 HYS02070
            IF( XLA .GT. HBP) THEN                                      HYS02080
              UD=X-HBP                                                  HYS02090
              UP=STIF*UD                                                HYS02100
              P=VBP+UP                                                  HYS02110
            ENDIF                                                       HYS02120
C                                                                       HYS02130
          ENDIF                                                         HYS02140
        ELSE IF( HIGH .AND. DX .LE. 0 ) THEN                            HYS02150
          PTOP=PLA                                                      HYS02160
          DTOP=XLA                                                      HYS02170
          IF( PBOT .EQ. 999999 ) THEN                                   HYS02180
C           STIF=-52030208*(ABS(DTOP)-ABS(HA)+0.0042)+856026            HYS02190
            IF((ABS(DTOP/FACH)-ABS(HA1)+0.0042) .LE. 0.0126)THEN        HYS02200
              STIF=(-52030208*(ABS(DTOP/FACH)-ABS(HA1)+0.0042)          HYS02210
     &              +856026)*FAC                                        HYS02220
            ELSE                                                        HYS02230
              STIF=(-52030208*0.0126 + 856026)*FAC                      HYS02240
            ENDIF                                                       HYS02250
            RULE=8                                                      HYS02260
            VAP=-2700*FACV                                              HYS02270
            HAP=DTOP-(2700*FACV+PTOP)/STIF                              HYS02280
            VBP=VAP+465*FACV                                            HYS02290
            HBP=HAP-0.0023*FACH                                         HYS02300
C           *** FOR UNBALANCE FORCE ***                                 HYS02310
            UD=X-XLA                                                    HYS02320
                                                                        HYS02330
            UP=STIF*UD                                                  HYS02340
            P=PLA+UP                                                    HYS02350
            IF( P .LT. VAP) P=VAP                                       HYS02360
          ELSE IF( PBOT .NE. 999999 ) THEN                              HYS02370
            CALL LOOP(VA1,HA1,PTOP,DTOP,PBOT,DBOT,HD,VD,HE,VE,          HYS02380
     &                SM1,SM2,FACH,FACV,FAC)                            HYS02390
            STIF=(PTOP-VD)/(DTOP-HD)                                    HYS02400
            RULE=4                                                      HYS02410
C           *** FOR UNBALANCE FORCE ***                                 HYS02420
            UD=X-XLA                                                    HYS02430
            UP=STIF*UD                                                  HYS02440
            P=PLA+UP                                                    HYS02450
            IF( P .LT. PBOT) P=PBOT                                     HYS02460
          ENDIF                                                         HYS02470
        ELSE IF( .NOT. HIGH .AND. DX .GE. 0 ) THEN                      HYS02480
          PBOT=PLA                                                      HYS02490
          DBOT=XLA                                                      HYS02500
          IF( PTOP .EQ. 999999 ) THEN                                   HYS02510
C           STIF=-52030208*(ABS(DBOT)-ABS(HAP)+0.0042)+856026           HYS02520
            IF((ABS(DBOT/FACH)-ABS(HAP1)+0.0042) .LE. 0.0126)THEN       HYS02530
              STIF=(-52030208*(ABS(DBOT/FACH)-ABS(HAP1)+0.0042)         HYS02540
     &              +856026)*FAC                                        HYS02550
            ELSE                                                        HYS02560
```

```
                STIF=(-52030208*0.0126 + 856026 )*FAC
              ENDIF
              RULE=8
              VA=2700*FACV
              HA=DBOT+(2700*FACV-PBOT)/STIF
              VB=VA-465*FACV
              HB=HA+0.0023*FACH
C             *** FOR UNBALANCE FORCE ***
              UD=X-XLA
              UP=STIF*UD
              P=PLA+UP
              IF( P .GT. VA) P=VA
            ELSE IF( PTOP .NE. 999999 ) THEN
              CALL LOOP(VA1,HA1,PBOT,DBOT,PTOP,DTOP,HD,VD,HE,VE,
     &                  SM1,SM2,FACH,FACV,FAC)
              STIF=(PBOT-VD)/(DBOT-HD)
              RULE=4
C             *** FOR UNBALANCE FORCE ***
              UD=X-XLA
              UP=STIF*UD
              P=PLA+UP
              IF( P .GT. PTOP) P=PTOP
            ENDIF
          ENDIF
          IF( DEMAX .LT. ABS(X) ) THEN
              DEMAX=ABS(X)
              DUC(1)=DEMAX/(HA1*FACH)
          ENDIF
          RETURN
          ENDIF
C
C---RULE 3;
        IF( RULE .EQ. 3 ) THEN
          IF( HIGH .AND. DX .GT. 0 ) THEN
              RULE=3
              STIF=SLOP3
              IF( P .LT. 0 )P=0.1*VA
              IF(P .LE. 0.1*VA) STIF=SLOP3*0.01
          ELSE IF( .NOT. HIGH .AND. DX .LT. 0 ) THEN
              RULE=3
              STIF=SLOP3
              IF( P .GT. 0 )P=0.1*VAP


              IF(P .GE. 0.1*VAP) STIF=SLOP3*0.001
          ELSE IF( HIGH .AND. DX .LE. 0 ) THEN
              PTOP=PLA
              DTOP=XLA
              IF( PBOT .EQ. 999999 ) THEN
                IF((ABS(DTOP/FACH)-ABS(HA1)+0.0042) .LE. 0.0126)THEN
                  STIF=(-52030208*(ABS(DTOP/FACH)-ABS(HA1)+0.0042)
     &                 +856026)*FAC
                ELSE
                  STIF=(-52030208*0.0126 + 856026)*FAC
                ENDIF
                RULE=8
                VAP=-2700*FACV
                HAP=DTOP-(2700*FACV+PTOP)/STIF
                VBP=VAP+465*FACV
                HBP=HAP-0.0023*FACH
C               *** FOR UNBALANCE FORCE ***
                UD=X-XLA
                UP=STIF*UD
                P=PLA+UP
                IF( P .LT. VAP) P=VAP
              ELSE IF( PBOT .NE. 999999 ) THEN
                CALL LOOP(VA1,HA1,PTOP,DTOP,PBOT,DBOT,HD,VD,HE,VE,
     &                    SM1,SM2,FACH,FACV,FAC)
                STIF=(PTOP-VD)/(DTOP-HD)
                RULE=4
C               *** FOR UNBALANCE FORCE ***
                UD=X-XLA
                UP=STIF*UD
                P=PLA+UP
                IF( P .LT. PBOT) P=PBOT
              ENDIF
          ELSE IF( .NOT. HIGH .AND. DX .GE. 0 ) THEN
              PBOT=PLA
              DBOT=XLA
              IF( PTOP .EQ. 999999 ) THEN
C               STIF=-52030208*(ABS(DBOT)-ABS(HAP)+0.0042)+856026
                IF((ABS(DBOT/FACH)-ABS(HAP1)+0.0042) .LE. 0.0126)THEN
                 STIF=(-52030208*(ABS(DBOT/FACH)-ABS(HAP1)+0.0042)
     &                +856026)*FAC
                ELSE
                 STIF=(-52030208*0.0126 + 856026)*FAC
                ENDIF
                RULE=8
                VA=2700*FACV
                HA=DBOT+(2700*FACV-PBOT)/STIF
                VB=VA-465*FACV
                HB=HA+0.0023*FACH
C               *** FOR UNBALANCE FORCE ***
                UD=X-XLA
                UP=STIF*UD
                P=PLA+UP
                IF( P .GT. VA) P=VA
              ELSE IF( PTOP .NE. 999999 ) THEN
                CALL LOOP(VA1,HA1,PBOT,DBOT,PTOP,DTOP,HD,VD,HE,VE,
     &                    SM1,SM2,FACH,FACV,FAC)
                STIF=(PBOT-VD)/(DBOT-HD)
                RULE=4
C               *** FOR UNBALANCE FORCE ***
                UD=X-XLA
                UP=STIF*UD
                P=PLA+UP
                IF( P .GT. PTOP) P=PTOP
              ENDIF
          ENDIF


          IF( DEMAX .LT. ABS(X) ) THEN
              DEMAX=ABS(X)
              DUC(1)=DEMAX/(HA1*FACH)
          ENDIF
          RETURN
          ENDIF
C
C---RULE 4, 7, 8, 9, 10;
        IF( RULE .GT. 3 ) THEN
            CALL HYST9B(MOPT,STIF,P,X,PLA,XLA,DP,DX,
     &              RV,RULE,PTOP,DTOP,PBOT,DBOT,
     &              SPTOP,SDTOP,SPBOT,SDBOT,
     &              SSPTOP,SSDTOP,SSPBOT,SSDBOT,
     &              HA,VA,HB,VB,HD,VD,HE,VE,HAP,VAP,HBP,VBP,
     &              SSHD,SSVD,SSHE,SSVE,
     &              SSSPTP,SSSDTP,SSSPBT,SSSDBT,
     &              SHD,SVD,SHE,SVE,PRINT,
     &              EKI,DUC,EXCR,DELT,ESE,PSE,PSEOLD,CSE,
     &              DEMAX,FACH,FACV,FAC,HIGH,SLOP1,SLOP2,SLOP3,
     &              HA1,VA1,HB1,VB1,HAP1,VAP1,HBP1,VBP1 )
          ENDIF
C
          RETURN
          END
C
          SUBROUTINE HYST9B(MOPT,STIF,P,X,PLA,XLA,DP,DX,
     &              RV,RULE,PTOP,DTOP,PBOT,DBOT,
     &              SPTOP,SDTOP,SPBOT,SDBOT,
     &              SSPTOP,SSDTOP,SSPBOT,SSDBOT,
     &              HA,VA,HB,VB,HD,VD,HE,VE,HAP,VAP,HBP,VBP,
     &              SSHD,SSVD,SSHE,SSVE,
     &              SSSPTP,SSSDTP,SSSPBT,SSSDBT,
     &              SHD,SVD,SHE,SVE,PRINT,
     &              EKI,DUC,EXCR,DELT,ESE,PSE,PSEOLD,CSE,
     &              DEMAX,FACH,FACV,FAC,HIGH,SLOP1,SLOP2,SLOP3,
     &              HA1,VA1,HB1,VB1,HAP1,VAP1,HBP1,VBP1 )
          DIMENSION DUC(3),EXCR(6),ESE(3)
          LOGICAL HIGH,PRINT
C
  400 IF( RULE .EQ. 4 ) THEN
          IF( HD .GT. HE .AND. X .GT. DBOT .AND. DX .LE. 0) THEN
            IF( X .GT. HD) THEN
                STIF=(PTOP-VD)/(DTOP-HD)
C               *** FOR UNBALANCED FORCE ***
                UD=X-HD
                UP=STIF*UD
                P=VD+UP
            ELSE IF( X .LE. HD .AND. X .GT. HE) THEN
                STIF=(VD-VE)/(HD-HE)
C               *** FOR UNBALANCED FORCE ***
                UD=X-HE
                UP=STIF*UD
                P=VE+UP
            ELSE IF( X .LE. HE .AND. X .GT. DBOT) THEN
                STIF=(VE-PBOT)/(HE-DBOT)
C               *** FOR UNBALANCED FORCE ***
                UD=X-DBOT
                UP=STIF*UD
                P=PBOT+UP
            ENDIF
            RULE=4
          ELSE IF( HD .GT. HE .AND. X .LE. DBOT .AND. DX .LE. 0 ) THEN
            IF( X .GT. HBP) THEN
               RULE=2
               CALL SRV(P,RV)
               STIF=SLOP2
C              *** FOR UNBALANCE FORCE ***
               UD=X-DBOT
               UP=STIF*UD
               P=PBOT+UP
            ELSE IF( X .LE. HBP) THEN
               RULE=3
               CALL SRV(P,RV)
               STIF=SLOP3
               IF( P .GT. 0.1*VAP .OR. PBOT .GT. 0.1*VAP)STIF=SLOP3*0.01
C              *** FOR UNBALANCE FORCE ***
               UD=X-DBOT
               UP=STIF*UD
               P=PBOT+UP
            ENDIF
          ELSE IF( HD .GT. HE .AND. DX .GT. 0 ) THEN
            SPBOT=PLA
            SDBOT=XLA
            SPTOP=PTOP
            SDTOP=DTOP
            CALL SLOOP(SPBOT,SDBOT,PTOP,DTOP,SHD,SVD,SHE,SVE,
     &                 SM1,SM2,FACH,FACV,FAC)
            STIF=(SPBOT-SVD)/(SDBOT-SHD)
            RULE=7
C           *** FOR UNBALANCE FORCE ***
            UD=X-XLA
            UP=STIF*UD
            P=PLA+UP
C           *** FOR UNBALANCE FORCE ***
          ELSE IF( HD .LT. HE .AND. X .LT. DTOP .AND. DX .GE. 0) THEN
            IF( X .LT. HD) THEN
                STIF=(PBOT-VD)/(DBOT-HD)
C               *** FOR UNBALANCED FORCE ***
                UD=X-HD
                UP=STIF*UD
                P=VD+UP
            ELSE IF( X .GE. HD .AND. X .LT. HE) THEN
               STIF=(VD-VE)/(HD-HE)
C              *** FOR UNBALANCED FORCE ***
               UD=X-HE
               UP=STIF*UD
               P=VE+UP
            ELSE IF( X .GE. HE .AND. X .LT. DTOP) THEN
               STIF=(VE-PTOP)/(HE-DTOP)
C              *** FOR UNBALANCED FORCE ***
               UD=X-DTOP
               UP=STIF*UD
               P=PTOP+UP
            ENDIF
            RULE=4
          ELSE IF( HD .LT. HE .AND. X .GE. DTOP .AND. DX .GE. 0 ) THEN
            IF( X .LT. HB) THEN
               RULE=2
               CALL SRV(P,RV)
               STIF=SLOP2
C              *** FOR UNBALANCE FORCE ***
               UD=X-DTOP
               UP=STIF*UD
               P=PTOP+UP
            ELSE IF( X .GE. HB) THEN
               RULE=3
               CALL SRV(P,RV)
               STIF=SLOP3
               IF( P .LT. 0.1*VA  .OR. PTOP .LT. 0.1*VA )STIF=SLOP3*0.01
C              *** FOR UNBALANCE FORCE ***


               UD=X-DTOP
               UP=STIF*UD
               P=PTOP+UP
            ENDIF
          ELSE IF( HD .LT. HE .AND. DX .LT. 0 ) THEN
            SPBOT=PBOT
            SDBOT=DBOT
            SPTOP=PLA
            SDTOP=XLA
            CALL SLOOP(SPTOP,SDTOP,PBOT,DBOT,SHD,SVD,SHE,SVE,
     &                 SM1,SM2,FACH,FACV,FAC)
            STIF=(SPTOP-SVD)/(SDTOP-SHD)
            RULE=7
C           *** FOR UNBALANCE FORCE ***
            UD=X-XLA
            UP=STIF*UD
            P=PLA+UP
          ENDIF
          IF( DEMAX .LT. ABS(X) ) THEN
              DEMAX=ABS(X)
              DUC(1)=DEMAX/(HA1*FACH)
          ENDIF
          RETURN
        ENDIF
C
C---RULE 7;
  700 IF( RULE .EQ. 7 ) THEN
          IF( SHD .GT. SHE .AND. X .GT. SDBOT .AND. DX .LE. 0) THEN
            IF( X .GT. SHD) THEN
                STIF=(SPTOP-SVD)/(SDTOP-SHD)
C               *** FOR UNBALANCED FORCE ***
                UD=X-SHD
                UP=STIF*UD
                P=SVD+UP
            ELSE IF(X .LT. SHD .AND. X .GT. SHE) THEN
                STIF=(SVD-SVE)/(SHD-SHE)
C               *** FOR UNBALANCED FORCE ***
                UD=X-SHE
                UP=STIF*UD
                P=SVE+UP
            ELSE IF(X .LT. SHE .AND. X .GT. SDBOT) THEN
```

```
              STIF=(SVE-SPBOT)/(SHE-SDBOT)                                      HYS05340
C             *** FOR UNBALANCED FORCE ***                                      HYS05350
              UD=X-SDBOT                                                        HYS05360
              UP=STIF*UD                                                        HYS05370
              P=SPBOT+UP                                                        HYS05380
            ENDIF                                                               HYS05390
            RULE=7                                                              HYS05400
          ELSE IF(SHD .GT. SHE .AND. X .LE. SDBOT .AND. DX .LE. 0 ) THEN        HYS05410
            IF( X .GT. HBP) THEN                                                HYS05420
              RULE=2                                                            HYS05430
              CALL SRV(P,RV)                                                    HYS05440
              STIF=SLOP2                                                        HYS05450
C             *** FOR UNBALANCE FORCE ***                                       HYS05460
              UD=X-SDBOT                                                        HYS05470
              UP=STIF*UD                                                        HYS05480
              P=SPBOT+UP                                                        HYS05490
            ELSE IF( X .LE. HBP) THEN                                           HYS05500
              RULE=3                                                            HYS05510
              CALL SRV(P,RV)                                                    HYS05520
              STIF=SLOP3                                                        HYS05530
             IF( P .GT. 0.1*VAP .OR. SPBOT .GT. 0.1*VAP)STIF=SLOP3*0.01         HYS05540
C             *** FOR UNBALANCE FORCE ***                                       HYS05550
              UD=X-SDBOT                                                        HYS05560
              UP=STIF*UD                                                        HYS05570
              P=SPBOT+UP                                                        HYS05580
            ENDIF                                                               HYS05590
          ELSE IF(SHD .GT. SHE .AND. DX .GT. 0 ) THEN                           HYS05600
            RULE=9                                                              HYS05610
            SSPTOP=SPTOP                                                        HYS05620
            SSDTOP=SDTOP                                                        HYS05630
            SSPBOT=PLA                                                          HYS05640
            SSDBOT=XLA                                                          HYS05650
            CALL SLOOP(SSPBOT,SSDBOT,SSPTOP,SSDTOP,SSHD,SSVD,SSHE,SSVE,        HYS05660
     &                 SM1,SM2,FACH,FACV,FAC)                                   HYS05670
            STIF=(SSPBOT-SSVD)/(SSDBOT-SSHD)                                    HYS05680
C           *** FOR UNBALANCE FORCE ***                                         HYS05690
            UD=X-XLA                                                            HYS05700
            UP=STIF*UD                                                          HYS05710
            P=PLA+UP                                                            HYS05720
C                                                                               HYS05730
          ELSE IF( SHD .LT. SHE .AND. X .LT. SDTOP .AND. DX .GE. 0) THEN        HYS05740
            IF(X .LT. SHD) THEN                                                 HYS05750
              STIF=(SPBOT-SVD)/(SDBOT-SHD)                                      HYS05760
C             *** FOR UNBALANCED FORCE ***                                      HYS05770
              UD=X-SHD                                                          HYS05780
              UP=STIF*UD                                                        HYS05790
              P=SVD+UP                                                          HYS05800
            ELSE IF(X .GE. SHD .AND. X .LT. SHE) THEN                           HYS05810
              STIF=(SVD-SVE)/(SHD-SHE)                                          HYS05820
C             *** FOR UNBALANCED FORCE ***                                      HYS05830
              UD=X-SHE                                                          HYS05840
              UP=STIF*UD                                                        HYS05850
              P=SVE+UP                                                          HYS05860
            ELSE IF(X .GE. SHE .AND. X .LT. SDTOP) THEN                         HYS05870
              STIF=(SVE-SPTOP)/(SHE-SDTOP)                                      HYS05880
C             *** FOR UNBALANCED FORCE ***                                      HYS05890
              UD=X-SDTOP                                                        HYS05900
              UP=STIF*UD                                                        HYS05910
              P=SPTOP+UP                                                        HYS05920
            ENDIF                                                               HYS05930
            RULE=7                                                              HYS05940
          ELSE IF(SHD .LT. SHE .AND. X .GE. SDTOP .AND. DX .GE. 0 ) THEN        HYS05950
            IF( X .LT. HB) THEN                                                 HYS05960
              RULE=2                                                            HYS05970
              CALL SRV(P,RV)                                                    HYS05980
              STIF=SLOP2                                                        HYS05990
C             *** FOR UNBALANCE FORCE ***                                       HYS06000
              UD=X-SDTOP                                                        HYS06010
              UP=STIF*UD                                                        HYS06020
              P=SPTOP+UP                                                        HYS06030
            ELSE IF( X .GE. HB) THEN                                            HYS06040
              RULE=3                                                            HYS06050
              CALL SRV(P,RV)                                                    HYS06060
              STIF=SLOP3                                                        HYS06070
             IF( P .LT. 0.1*VA .OR. SPTOP .LT. 0.1*VA)STIF=SLOP3*0.01           HYS06080
C             *** FOR UNBALANCE FORCE ***                                       HYS06090
              UD=X-SDTOP                                                        HYS06100
              UP=STIF*UD                                                        HYS06110
              P=SPTOP+UP                                                        HYS06120
            ENDIF                                                               HYS06130
          ELSE IF(SHD .LT. SHE .AND. DX .LT. 0 ) THEN                           HYS06140
            RULE=9                                                              HYS06150
            SSPTOP=PLA                                                          HYS06160
            SSDTOP=XLA                                                          HYS06170
            SSPBOT=SPBOT                                                        HYS06180
            SSDBOT=SDBOT                                                        HYS06190
            CALL SLOOP(SSPTOP,SSDTOP,SSPBOT,SSDBOT,SSHD,SSVD,SSHE,SSVE,        HYS06200
     &                 SM1,SM2,FACH,FACV,FAC)                                   HYS06210
            STIF=(SSPTOP-SSVD)/(SSDTOP-SSHD)                                    HYS06220


C             *** FOR UNBALANCE FORCE ***                                       HYS06230
              UD=X-XLA                                                          HYS06240
              UP=STIF*UD                                                        HYS06250
              P=PLA+UP                                                          HYS06260
            ENDIF                                                               HYS06270
            IF( DEMAX .LT. ABS(X) ) THEN                                        HYS06280
                DEMAX=ABS(X)                                                    HYS06290
                DUC(1)=DEMAX/(HA1*FACH)                                         HYS06300
            ENDIF                                                               HYS06310
            RETURN                                                              HYS06320
          ENDIF                                                                 HYS06330
C                                                                               HYS06340
C---RULE 8                                                                      HYS06350
        IF( RULE .EQ. 8 ) THEN                                                  HYS06360
          IF( PTOP .EQ. 999999 ) THEN                                           HYS06370
            S1=VA                                                               HYS06380
          ELSE IF( PTOP .NE. 999999 ) THEN                                      HYS06390
            S1=PTOP                                                             HYS06400
          ENDIF                                                                 HYS06410
          IF( PBOT .EQ. 999999 ) THEN                                           HYS06420
            S2=VAP                                                              HYS06430
          ELSE IF( PBOT .NE. 999999 ) THEN                                      HYS06440
            S2=PBOT                                                             HYS06450
          ENDIF                                                                 HYS06460
          IF( P .LT. S1 .AND. P .GT. S2 ) THEN                                  HYS06470
            RULE=8                                                              HYS06480
            STIF=STIF                                                           HYS06490
          ELSE IF( P .LE. S2 ) THEN                                             HYS06500
            IF( PBOT .EQ. 999999 ) THEN                                         HYS06510
C             *** FOR UNBALANCE FORCE ***                                       HYS06520
              UP1=P-S2                                                          HYS06530
              UD=UP1/STIF                                                       HYS06540
              UP2=-UD*SLOP2                                                     HYS06550
              P=P-(UP1+UP2)                                                     HYS06560
              HAP=X-UD                                                          HYS06570
              VAP=S2                                                            HYS06580
C                                                                               HYS06590
              RULE=2                                                            HYS06600
              CALL SRV(P,RV)                                                    HYS06610
              STIF=SLOP2                                                        HYS06620
              HBP=HAP-0.0023*FACH                                               HYS06630
              VBP=VAP+465*FACV                                                  HYS06640
            ELSE IF( PBOT .NE. 999999 ) THEN                                    HYS06650
              IF( X .GT. HBP) THEN                                              HYS06660
                RULE=2                                                          HYS06670
                CALL SRV(P,RV)                                                  HYS06680
                STIF=SLOP2                                                      HYS06690
C               *** FOR UNBALANCED FORCE ***                                    HYS06700
                UD=X-DBOT                                                       HYS06710
                UP=STIF*UD                                                      HYS06720
                P=PBOT+UP                                                       HYS06730
              ELSE IF( X .LE. HBP) THEN                                         HYS06740
                RULE=3                                                          HYS06750
                CALL SRV(P,RV)                                                  HYS06760
                STIF=SLOP3                                                      HYS06770
                IF( P .GT. 0.1*VAP .OR. PBOT .GT. 0.1*VAP)                      HYS06780
     &          STIF=SLOP3*0.01                                                HYS06790
C               *** FOR UNBALANCED FORCE ***                                    HYS06800
                UD=X-DBOT                                                       HYS06810
                UP=STIF*UD                                                      HYS06820
                P=PBOT+UP                                                       HYS06830
              ENDIF                                                             HYS06840
            ENDIF                                                               HYS06850
C           ELSE IF( P .GE. S1 ) THEN                                           HYS06860
                                                                                HYS06870
              IF( PTOP .EQ. 999999 ) THEN                                       HYS06880
C               *** FOR UNBALANCE FORCE ***                                     HYS06890
                UP1=P-S1                                                        HYS06900
                UD=UP1/STIF                                                     HYS06910
                UP2=-UD*SLOP2                                                   HYS06920
                P=P-(UP1+UP2)                                                   HYS06930
                HA=X-UD                                                         HYS06940
                VA=S1                                                           HYS06950
C                                                                               HYS06960
                RULE=2                                                          HYS06970
                CALL SRV(P,RV)                                                  HYS06980
                STIF=SLOP2                                                      HYS06990
                HB=HA+0.0023*FACH                                               HYS07000
                VB=VA-465*FACV                                                  HYS07010
              ELSE IF( PTOP .NE. 999999 ) THEN                                  HYS07020
                IF( X .LT. HB) THEN                                             HYS07030
                  RULE=2                                                        HYS07040
                  CALL SRV(P,RV)                                                HYS07050
                  STIF=SLOP2                                                    HYS07060
C                 *** FOR UNBALANCED FORCE ***                                  HYS07070
                  UD=X-DTOP                                                     HYS07080
                  UP=STIF*UD                                                    HYS07090
                  P=PTOP+UP                                                     HYS07100
                ELSE IF( X .GE. HB) THEN                                        HYS07110
                  RULE=3                                                        HYS07120
                  CALL SRV(P,RV)                                                HYS07130
                  STIF=SLOP3                                                    HYS07140
                  IF( P .LT. 0.1*VA .OR. PTOP .LT. 0.1*VA)                      HYS07150
     &            STIF=SLOP3*0.01                                             HYS07160
C                 *** FOR UNBALANCED FORCE ***                                  HYS07170
                  UD=X-DTOP                                                     HYS07180
                  UP=STIF*UD                                                    HYS07190
                  P=PTOP+UP                                                     HYS07200
                ENDIF                                                           HYS07210
              ENDIF                                                             HYS07220
            ENDIF                                                               HYS07230
            IF( DEMAX .LT. ABS(X) ) THEN                                        HYS07240
                DEMAX=ABS(X)                                                    HYS07250
                DUC(1)=DEMAX/(HA1*FACH)                                         HYS07260
            ENDIF                                                               HYS07270
            RETURN                                                              HYS07280
          ENDIF                                                                 HYS07290
C                                                                               HYS07300
C---RULE 9:                                                                     HYS07310
  900 IF( RULE .EQ. 9 ) THEN                                                    HYS07320
        IF( SSHD .GT. SSHE .AND. X .GT. SSDBOT .AND. DX .LE. 0) THEN            HYS07330
          IF( X .GT. SSHD) THEN                                                 HYS07340
            STIF=(SSPTOP-SSVD)/(SSDTOP-SSHD)                                    HYS07350
C           *** FOR UNBALANCED FORCE ***                                        HYS07360
            UD=X-SSHD                                                           HYS07370
            UP=STIF*UD                                                          HYS07380
            P=SSVD+UP                                                           HYS07390
          ELSE IF(X .LT. SSHD .AND. X .GT. SSHE) THEN                           HYS07400
            STIF=(SSVD-SSVE)/(SSHD-SSHE)                                        HYS07410
C           *** FOR UNBALANCED FORCE ***                                        HYS07420
            UD=X-SSHE                                                           HYS07430
            UP=STIF*UD                                                          HYS07440
            P=SSVE+UP                                                           HYS07450
          ELSE IF(X .LT. SSHE .AND. X .GT. SSDBOT) THEN                         HYS07460
            STIF=(SSVE-SSPBOT)/(SSHE-SSDBOT)                                    HYS07470
C           *** FOR UNBALANCED FORCE ***                                        HYS07480
            UD=X-SSDBOT                                                         HYS07490
            UP=STIF*UD                                                          HYS07500
            P=SSPBOT+UP                                                         HYS07510
          ENDIF                                                                 HYS07520


          RULE=9                                                                HYS07530
        ELSE IF(SSHD .GT. SSHE .AND.                                            HYS07540
     &        X .LE. SSDBOT .AND. DX .LE. 0 ) THEN                              HYS07550
            RULE=4                                                              HYS07560
            GO TO 400                                                           HYS07570
        ELSE IF(SSHD .GT. SSHE .AND. DX .GT. 0 ) THEN                           HYS07580
          RULE=10                                                               HYS07590
          SSSPTP=SSPTOP                                                         HYS07600
          SSSDTP=SSDTOP                                                         HYS07610
          SSSPBT=PLA                                                            HYS07620
          SSSDBT=XLA                                                            HYS07630
          STIF=(SSSPBT-SSSPTP)/(SSSDBT-SSSDTP)                                  HYS07640
C         *** FOR UNBALANCE FORCE ***                                           HYS07650
          UD=X-XLA                                                              HYS07660
          UP=STIF*UD                                                            HYS07670
          P=PLA+UP                                                              HYS07680
C                                                                               HYS07690
        ELSE IF( SSHD .LT. SSHE .AND.                                           HYS07700
     &        X .LT. SSDTOP .AND. DX .GE. 0) THEN                               HYS07710
          IF(X .LT. SSHD) THEN                                                  HYS07720
            STIF=(SSPBOT-SSVD)/(SSDBOT-SSHD)                                    HYS07730
C           *** FOR UNBALANCED FORCE ***                                        HYS07740
            UD=X-SSHD                                                           HYS07750
            UP=STIF*UD                                                          HYS07760
            P=SSVD+UP                                                           HYS07770
          ELSE IF(X .GE. SSHD .AND. X .LT. SSHE) THEN                           HYS07780
            STIF=(SSVD-SSVE)/(SSHD-SSHE)                                        HYS07790
C           *** FOR UNBALANCED FORCE ***                                        HYS07800
            UD=X-SSHE                                                           HYS07810
            UP=STIF*UD                                                          HYS07820
            P=SSVE+UP                                                           HYS07830
          ELSE IF(X .GE. SSHE .AND. X .LT. SSDTOP) THEN                         HYS07840
            STIF=(SSVE-SSPTOP)/(SSHE-SSDTOP)                                    HYS07850
C           *** FOR UNBALANCED FORCE ***                                        HYS07860
            UD=X-SSDTOP                                                         HYS07870
            UP=STIF*UD                                                          HYS07880
            P=SSPTOP+UP                                                         HYS07890
          ENDIF                                                                 HYS07900
          RULE=9                                                                HYS07910
        ELSE IF(SSHD .LT. SSHE .AND.                                            HYS07920
     &        X .GE. SSDTOP .AND. DX .GE. 0 ) THEN                              HYS07930
            RULE=4                                                              HYS07940
            GO TO 400                                                           HYS07950
        ELSE IF(SSHD .LT. SSHE .AND. DX .LT. 0 ) THEN                           HYS07960
          RULE=10                                                               HYS07970
          SSSPTP=PLA                                                            HYS07980
          SSSDTP=XLA                                                            HYS07990
          SSSPBT=SSPBOT                                                         HYS08000
          SSSDBT=SSDBOT                                                         HYS08010
          STIF=(SSSPBT-SSSPTP)/(SSSDBT-SSSDTP)                                  HYS08020
C         *** FOR UNBALANCE FORCE ***                                           HYS08030
          UD=X-XLA                                                              HYS08040
          UP=STIF*UD                                                            HYS08050
          P=PLA+UP                                                              HYS08060
        ENDIF                                                                   HYS08070
        IF( DEMAX .LT. ABS(X) ) THEN                                            HYS08080
            DEMAX=ABS(X)                                                        HYS08090
            DUC(1)=DEMAX/(HA1*FACH)                                             HYS08100
        ENDIF                                                                   HYS08110
```

```
        RETURN                                                          HYS08120
      ENDIF                                                             HYS08130
C                                                                       HYS08140
C                                                                       HYS08150
C---RULE 10:                                                            HYS08160
      IF( RULE .EQ. 10 ) THEN                                           HYS08170
        IF( SSHD .GT. SSHE ) THEN                                       HYS08180
          IF( X .LT. SSSDTP .AND. X .GT. SSSDBT ) THEN                  HYS08190
            RULE=10                                                     HYS08200
            STIF=(SSSPBT-SSSPTP)/(SSSDBT-SSSDTP)                        HYS08210
          ELSE IF( X .LT. SSSDBT ) THEN                                 HYS08220
            RULE=9                                                      HYS08230
            GO TO 900                                                   HYS08240
          ELSE IF( X .GT. SSSDTP ) THEN                                 HYS08250
            RULE=7                                                      HYS08260
            GO TO 700                                                   HYS08270
          ENDIF                                                         HYS08280
        ELSE IF( SSHD .LT. SSHE ) THEN                                  HYS08290
          IF( X .LT. SSSDTP .AND. X .GT. SSSDBT ) THEN                  HYS08300
            RULE=10                                                     HYS08310
            STIF=(SSSPBT-SSSPTP)/(SSSDBT-SSSDTP)                        HYS08320
          ELSE IF( X .GT. SSSDTP ) THEN                                 HYS08330
            RULE=9                                                      HYS08340
            GO TO 900                                                   HYS08350
          ELSE IF( X .LT. SSSDBT ) THEN                                 HYS08360
            RULE=7                                                      HYS08370
            GO TO 700                                                   HYS08380
          ENDIF                                                         HYS08390
        ENDIF                                                           HYS08400
        IF( DEMAX .LT. ABS(X) ) THEN                                    HYS08410
          DEMAX=ABS(X)                                                  HYS08420
          DUC(1)=DEMAX/(HA1*FACH)                                       HYS08430
        ENDIF                                                           HYS08440
        RETURN                                                          HYS08450
      ENDIF                                                             HYS08460
C                                                                       HYS08470
      END                                                               HYS08480
C                                                                       HYS08490
C.PA                                                                    HYS08500
      SUBROUTINE LOOP(VA1,HA1,VC,HC,VB,HB,HD,VD,HE,VE,                  HYS08510
     &                SM1,SM2,FACH,FACV,FAC)                            HYS08520
C                                                                       HYS08530
C     VA1=VA/FACV                                                       HYS08540
C     HA1=HA/FACH                                                       HYS08550
      VC1=VC/FACV                                                       HYS08560
      HC1=HC/FACH                                                       HYS08570
      VB1=VB/FACV                                                       HYS08580
      HB1=HB/FACH                                                       HYS08590
C                                                                       HYS08600
      SM1=(VC1-VB1)/(HC1-HB1)                                           HYS08610
      IF( (ABS(HC1)-ABS(HA1)+0.0042) .LE. 0.0126 ) THEN                 HYS08620
          SM2=-52030208*(ABS(HC1)-ABS(HA1)+0.0042)+856026               HYS08630
      ELSE IF( (ABS(HC1)-ABS(HA1)+0.0042) .GT. 0.0126 ) THEN            HYS08640
          SM2=-52030208*0.0126 + 856026                                 HYS08650
      ENDIF                                                             HYS08660
      FACT1=.5                                                          HYS08670
      FACT2=.1                                                          HYS08680
      HL=HC1-HB1                                                        HYS08690
      VL=VC1-VB1                                                        HYS08700
      H1=HB1+HL*FACT1                                                   HYS08710
      V1=VB1+VL*FACT1                                                   HYS08720
      H2=HB1+HL*FACT2                                                   HYS08730
      V2=VB1+VL*FACT2                                                   HYS08740
      HD1=(V1+(H1/SM1)-VC1+SM2*HC1)/(SM2+1/SM1)                         HYS08750
      VD1=SM2*HD1+VC1-SM2*HC1                                           HYS08760
      HE1=HD1-HL*(FACT1-FACT2)                                          HYS08770
      VE1=VD1-VL*(FACT1-FACT2)                                          HYS08780
      HD=HD1*FACH                                                       HYS08790
      VD=VD1*FACV                                                       HYS08800
      HE=HE1*FACH                                                       HYS08810
      VE=VE1*FACV                                                       HYS08820
C                                                                       HYS08830
      RETURN                                                            HYS08840
      END                                                               HYS08850
C                                                                       HYS08860
C.PA                                                                    HYS08870
      SUBROUTINE SLOOP(VC,HC,VB,HB,HD,VD,HE,VE,                         HYS08880
     &                 SM1,SM2,FACH,FACV,FAC)                           HYS08890
C                                                                       HYS08900
      VC1=VC/FACV                                                       HYS08910
      HC1=HC/FACH                                                       HYS08920
      VB1=VB/FACV                                                       HYS08930
      HB1=HB/FACH                                                       HYS08940
C                                                                       HYS08950
      SM1=(VC1-VB1)/(HC1-HB1)                                           HYS08960
      SM2=447160                                                        HYS08970
      FACT1=.4                                                          HYS08980
      FACT2=.1                                                          HYS08990
      HL=HC1-HB1                                                        HYS09000
      VL=VC1-VB1                                                        HYS09010
      H1=HB1+HL*FACT1                                                   HYS09020
      V1=VB1+VL*FACT1                                                   HYS09030
      H2=HB1+HL*FACT2                                                   HYS09040
      V2=VB1+VL*FACT2                                                   HYS09050
      HD1=(V1+(H1/SM1)-VC1+SM2*HC1)/(SM2+1/SM1)                         HYS09060
      VD1=SM2*HD1+VC1-SM2*HC1                                           HYS09070
      HE1=HD1-HL*(FACT1-FACT2)                                          HYS09080
      VE1=VD1-VL*(FACT1-FACT2)                                          HYS09090
      HD=HD1*FACH                                                       HYS09100
      VD=VD1*FACV                                                       HYS09110
      HE=HE1*FACH                                                       HYS09120
      VE=VE1*FACV                                                       HYS09130
C                                                                       HYS09140
      RETURN                                                            HYS09150
      END                                                               HYS09160
C                                                                       HYS09170
C                                                                       HYS09180
      SUBROUTINE SRV(P,RV)                                              HYS09190
      IF( P .GT. 0 ) THEN                                               HYS09200
        RV=0                                                            HYS09210
      ELSE IF( P .LT. 0 ) THEN                                          HYS09220
        RV=1                                                            HYS09230
      ENDIF                                                             HYS09240
C                                                                       HYS09250
      RETURN                                                            HYS09260
      END                                                               HYS09270
C******* HYST11 ************************************************       HYS09010
C     DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                          HYS00020
C     END DEBUG                                                         HYS00030
C                                                                       HYS00040
C     PINO-SUAREZ TOWER SHORT DIRECTION GIRDER MOMENT-ROTATION          HYS00050
C     HYSTERESIS MODEL BASED ON HYSTERESIS LOOP OF T2A GIRDER           HYS00060
C                                                                       HYS00070
      SUBROUTINE HYST11(MOPT,STIF,P,X,PLA,XLA,DP,DX,                    HYS00080
     &                  RV,RULE,PTOP,DTOP,PBOT,DBOT,                    HYS00090
     &                  HA,VA,RATIO,PRINT,                              HYS00100
     &                  SKI,DUC,EXCR,DELT,ESE,PSE,PSEOLD,CSE,DEMAX,DUCMAX)  HYS00110
      DIMENSION DUC(3),EXCR(6),ESE(3)                                   HYS00120
      LOGICAL HIGH,PRINT                                                HYS00130
C                                                                       HYS00140
C ***************************************************************       HYS00150
C   DUC    ---- ELEMENT DUCTILITY RATE                                 HYS00160
C   EXCR   ---- ELEMENT EXCURSION RATE                                 HYS00170
C   ESE    ---- ELEMENT ELASTIC STRAIN RATE                           HYS00180
C   PSE    ---- ELEMENT PLASTIC STRAIN RATE                           HYS00190
C   PSEOLD ---- ELEMENT PLASTIC STRAIN RATE AT LAST ZERO CROSSING      HYS00200
C   CSE    ---- ELEMENT CONSTANT STRAIN ENERGY                        HYS00210
C   RATIO  ---- STRAIN HARTEN RATIO TO ELASTIC STIFFNESS              HYS00220
C   STIF   ---- ELEMENT STIFFNESS                                      HYS00230
C   P      ---- CURRENT ELEMENT INTERNAL FORCE                        HYS00240
C   X      ---- CURRENT ELEMENT INTERNAL DISPLACEMENT                 HYS00250
C   PLA    ---- PREVIOUS ELEMENT INTERNAL FORCE                       HYS00260
C   XLA    ---- PREVIOUS ELEMENT INTERNAL DISPLACEMENT                HYS00270
C   DP     ---- ELEMENT INTERNAL FORCE INCREMENT                      HYS00280
C   DX     ---- ELEMENT INTERNAL DISPLACEMENT INCREMENT               HYS00290
C   HIGH   ---- LOGICAL INDEX, IF P .GT. 0 THEN HIGH=.TRUE.           HYS00300
C   RULE   ---- RULE NUMBER                                           HYS00310
C   PTOP   ---- REVERSED LOADING POINT AT TOP ENVELOPE                HYS00320
C   DTOP   ---- REVERSED DISP. POINT AT TOP ENVELOPE                  HYS00330
C   PBOT   ---- REVERSED LOADING POINT AT BOTTOM ENVELOPE             HYS00340
C   DBOT   ---- REVERSED DISP. POINT AT BOTTOM ENVELOPE               HYS00350
C   HA,VA  ---- FIRST BUCKLING POINT AT TOP ENVELOPE                  HYS00360
C   DUCMAX ---- FAILURE DUCTILITY                                     HYS00370
C               .GE. 1 FAILURE DUCTILITY CONSIDERED                   HYS00380
C               .LT. 1 FAILURE DUCTILITY NOT CONSIDERED               HYS00390
C ***************************************************************     HYS00400
C                                                                       HYS00410
C ---------------------------                                          HYS00420
C = CHECK FAILURE DUCTILITY ==                                          HYS00430
C ---------------------------                                          HYS00440
      IF( DUC(1) .GE. DUCMAX .AND. DUCMAX .GE. 1 ) THEN                 HYS00450
        P=0                                                             HYS00460
        STIF=0.001                                                      HYS00470
        RETURN                                                          HYS00480
      ENDIF                                                             HYS00490
C                                                                       HYS00500
        IF( RV .EQ. 1 ) HIGH=.FALSE.                                    HYS00510
        IF( RV .EQ. 0 ) HIGH=.TRUE.                                     HYS00520
C                                                                       HYS00530
C ---------------------------------                                     HYS00540
C = DEFINE ENVELOPE CONTROL POINTS =                                    HYS00550
C ---------------------------------                                     HYS00560
        IF( MOPT .EQ. 2 ) THEN                                          HYS00570
          PTOP=VA                                                       HYS00580
          PBOT=-VA                                                      HYS00590
          DTOP=HA                                                       HYS00600
          DBOT=-HA                                                      HYS00610
        ENDIF                                                           HYS00620
C                                                                       HYS00630
      IF( P .GT. 0 ) HIGH=.TRUE.                                        HYS00640
      IF( P .LT. 0 ) HIGH=.FALSE.                                       HYS00650
      IF( HIGH) RV=0                                                    HYS00660
      IF( .NOT. HIGH) RV=1                                              HYS00670
C                                                                       HYS00680
C -------------------------                                            HYS00690
C = DEFINE ENVELOPE SLOPE =                                             HYS00700
C -------------------------                                            HYS00710
      SLOP1=VA/HA                                                       HYS00720
      SLOP2=RATIO*SLOP1                                                 HYS00730
C                                                                       HYS00740
      IF (PRINT) THEN                                                   HYS00750
      WRITE(6,567)RULE,HA,VA,                                           HYS00760
     &            SLOP1,SLOP2                                           HYS00770
  567 FORMAT(5X,'PINO-SUAREZ TOWER SHORT DIR. M-R LOOP CONTROL POINTS.',HYS00780
     &  5X,'-------------------------------------------------------',HYS00790
     &       5X,'RULE    ----------=',1X,G10.4,/                       HYS00800
     &       5X,'(HA, VA) ----------=',1X,'( ',F10.4,',',F10.4,' )',/  HYS00810
     &       5X,'SLOP1    ----------=',1X,G10.4,/                      HYS00820
     &       5X,'SLOP2    ----------=',1X,G10.4 )                      HYS00830
      ENDIF                                                             HYS00840
C                                                                       HYS00850
C ------------------------                                             HYS00860
C = HYSTERESIS LOOP RULES =                                             HYS00870
C ------------------------                                             HYS00880
C--RULE 1:                                                              HYS00890
      IF( RULE .EQ. 1 ) THEN                                            HYS00900
        IF(P .GE. PTOP .OR. P .LE. PBOT ) THEN                          HYS00910
          RULE=2                                                        HYS00920
          STIF=SLOP2                                                    HYS00930
C         *** UNBALANCED FORCE ***                                      HYS00940
          IF( P .GE. PTOP ) THEN                                        HYS00950
            UP1=P-PTOP                                                  HYS00960
            UD=UP1/SLOP1                                                HYS00970
            UP2=-UD*SLOP2                                               HYS00980
            P=P-(UP1+UP2)                                               HYS00990
          ELSE IF( P .LE. PBOT ) THEN                                   HYS01000
            UP1=P-PBOT                                                  HYS01010
            UD=UP1/SLOP1                                                HYS01020
            UP2=-UD*SLOP2                                               HYS01030
            P=P-(UP1+UP2)                                               HYS01040
          ENDIF                                                         HYS01050
C                                                                       HYS01060
        ELSE                                                            HYS01070
          RULE=1                                                        HYS01080
          STIF=SLOP1                                                    HYS01090
        ENDIF                                                           HYS01100
        IF( DEMAX .LT. ABS(X) ) THEN                                    HYS01110
          DEMAX=ABS(X)                                                  HYS01120
          DUC(1)=DEMAX/HA                                               HYS01130
        ENDIF                                                           HYS01140
        RETURN                                                          HYS01150
      ENDIF                                                             HYS01160
C                                                                       HYS01170
C---RULE 2:                                                             HYS01180
      IF( RULE .EQ. 2 ) THEN                                            HYS01190
        IF( HIGH .AND. DX .GT. 0 ) THEN                                 HYS01200
          RULE=2                                                        HYS01210
          STIF=SLOP2                                                    HYS01220
        ELSE IF( .NOT. HIGH .AND. DX .LT. 0 ) THEN                      HYS01230
          RULE=2                                                        HYS01240
          STIF=SLOP2                                                    HYS01250
        ELSE IF( HIGH .AND. DX .LE. 0 ) THEN                            HYS01260
          PTOP=PLA                                                      HYS01270
          DTOP=XLA                                                      HYS01280
          RULE=1                                                        HYS01290
          STIF=SLOP1                                                    HYS01300
          PBOT=PTOP - 2*VA                                              HYS01310
          DBOT=DTOP - (PTOP-PBOT)/SLOP1                                 HYS01320
C         *** UNBALANCED FORCE ***                                      HYS01330
          UD=X-DTOP                                                     HYS01340
          UP=UD*SLOP1                                                   HYS01350
          P=PTOP+UP                                                     HYS01360
        ELSE IF( .NOT. HIGH .AND. DX .GE. 0 ) THEN                      HYS01370
          PBOT=PLA                                                      HYS01380
          DBOT=XLA                                                      HYS01390
          RULE=1                                                        HYS01400
          STIF=SLOP1                                                    HYS01410
          PTOP=PBOT + 2*VA                                              HYS01420
          DTOP=DBOT + (PTOP-PBOT)/SLOP1                                 HYS01430
C         *** UNBALANCED FORCE ***                                      HYS01440
          UD=X-DBOT                                                     HYS01450
          UP=UD*SLOP1                                                   HYS01460
          P=PBOT+UP                                                     HYS01470
        ENDIF                                                           HYS01480
        IF( DEMAX .LT. ABS(X) ) THEN                                    HYS01490
          DEMAX=ABS(X)                                                  HYS01500
          DUC(1)=DEMAX/HA                                               HYS01510
        ENDIF                                                           HYS01520
        RETURN                                                          HYS01530
        ENDIF                                                           HYS01540
C                                                                       HYS01550
        END                                                             HYS01560
C****** HYST12 ************************************************         HYS00010
C     DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                          HYS00020
C     END DEBUG                                                         HYS00030
C                                                                       HYS00040
C       THIS SUBROUTINE IS USED FOR NONLINEAR BEAM-COLUMN ANALYSIS,     HYS00050
C       WHICH CAN ANALYSIS THE POSTBUCKING, UNLOADING, AND RELOADING    HYS00060
C       HYSTERISTIC LOOPS.                                              HYS00070
C                                                                       HYS00080
C     ---------------------------------------------------------------   HYS00090
      SUBROUTINE HYST12( IOPT ,  NSEG,   IS,   EM,   LIBN,   HH,   UU,HYS00100
     &                   WW   ,  ZZ  , INSB, INEH,   ST  , IREV1,IREV2,HYS00110
     &                   IREVJ,  IREV4,IECOP,ECCX0,  ECCY0,  NUMP,     HYS00120
```

```
     &                    ELS  ,  SMALL,RATIX0,RATIY0,  TOTA,  IMD,      HYS00130
     &                    ISPE ,  DELTP2,S2    ,SP    ,ISTART, PRINT,   HYS00140
     &                    IELNO, ISTIF, IAUTO,IMATER,                    HYS00150
     &                    RATIO3, IR, G, QRNEE, RNEEE,                   HYS00160
C                                                                         HYS00170
     &                   RNP,RIFM, BETA, ECX, ECY, U, V, U0, V0, GCD,    HYS00180
     &                    EL, D1, DD, D00, TL, XLOC, FLOC, FGLOB, XGLO,  HYS00190
     &                    E0, E1, E2, TE, PR, TIR, SEASAT, STF,RMDIAG,   HYS00200
     &                    SEASAT, TPJT, DP, DUMMT, PSK, ECCX, ECCY,      HYS00210
     &                    FF1,FACTOR,DPCOPY,ASAT,E,DE,                   HYS00220
     &                    A0, B ,STR,DEP,EOP,STRP,TEP,EREF,SREF,FJL,FLP) HYS00230
C    &                    EKI,DUC,EXCR,DELY,ESE,PSE,PSEOLD,CSE)          HYS00240
C                                                                         HYS00250
C                                                                         HYS00260
C  IOPT   = 1 : FORMULATE THE INITIAL MEMBER STIFFNESS                    HYS00270
C         = 2 : FORMULATE THE NONLINEAR MEMBER STIFFNESS                  HYS00280
C  RATIX0 = INITIAL IMPERFECTION OF THE MEMBER IN X DIRECTION             HYS00290
C  RATIY0 = INITIAL IMPERFECTION OF THE MEMBER IN Y DIRECTION             HYS00300
C  IECOP  = ECCENTRIC LOAD OPTION                                         HYS00310
C  WW     = ECCENTRIC DISTANCE IN X DIRECTION                             HYS00320
C  ZZ     = ECCENTRIC DISTANCE IN Y DIRECTION                             HYS00330
C  NUMP   = TOTAL SECTIONAL POINT ON THE SEGMENT CROSS-SECTION            HYS00340
C  ASAT   = MEMBER GLOBAL STIFFNESS MATRIX                                HYS00350
C  SEASAT = CONDENSED MATRIX                                              HYS00360
C  DUMMT  = RESIDUAL MATRIX                                               HYS00370
C  DD     = MEMBER'S DISPLACEMENT INCREMENT AT EACH D.O.F.                 HYS00380
C  NSEG   = TOTAL NO OF SEGMENTS IN THE MEMBER                            HYS00390
C  IDOF   = TOTAL D.O.F. OF THE MEMBER                                    HYS00400
C  ELS    = TOTAL LENGTH OF THE MEMBER                                    HYS00410
C  LIBN   = THE CROSS-SECTION LIBRARY NO.                                 HYS00420
C  FLP    = FLAG FOR JUDING LOCAL BUCKLING                                HYS00430
C           0  : NO LOCAL BUCKLING                                        HYS00440
C           1-4: LOCAL BUCKLING OCCURRED                                  HYS00450
C  IELNO  = THE STABILITY MEMBER NO.                                      HYS00460
C  NOMEM  = SYSTEM MEMBER NO.                                             HYS00470
C  RNEEE  = INDEX FOR LIMITING NORM OF DISPLACEMENT                       HYS00480
C           0 : EEE .LT. QRNEE                                            HYS00490
C           1 : EEE .GT. QRNEE                                            HYS00500
C  ECX    = DISTANCE FROM REFERENCE COORDINATE (U,V,W) ORIGIN TO          HYS00510
C           ECCENTRIC LOAD LOCATION IN U DIRECTION                        HYS00520
C  ECY    = DISTANCE FROM REFERENCE COORDINATE (U,V,W) ORIGIN TO          HYS00530
C           ECCENTRIC LOAD LOCATION IN V DIRECTION                        HYS00540
C                                                                         HYS00550
      IMPLICIT REAL(A-H,O-Z)                                              HYS00560
      DIMENSION NP(  32,12), IFM(     198  ), BETA(NSEG),ECX(2),ECY(2)    HYS00570
      DIMENSION U(NSEG,NUMP),V(NSEG,NUMP),U0(NUMP),V0(NUMP)               HYS00580
      DIMENSION GCD(NSEG,2,3),D1((NSEG+1)*6,1),DD((NSEG+1)*6,1),          HYS00590
     &          DP((NSEG+1)*6,1),ASAT((NSEG+1)*6,(NSEG+1)*6)              HYS00600
      DIMENSION D00((NSEG+1)*6,1),EL(NSEG), TL(NSEG), XLOC(NSEG,12)       HYS00610
      DIMENSION FLOC(NSEG,12),FGLOB(NSEG,12),XGLO(NSEG,12),E0(NSEG,NUMP)HYS00620
      DIMENSION E1(NSEG,NUMP),E2(NSEG,NUMP),TE(NUMP),PR(NSEG,9)           HYS00630
      DIMENSION E(NUMP),DE(NSEG,NUMP)                                     HYS00640
      DIMENSION TIR(NSEG,9),SK(12,12),RTSKR(12,12),                       HYS00650
     &          SEASAT((NSEG+1)*6,(NSEG+1)*6)                             HYS00660
      DIMENSION STF((NSEG+1)*6*((NSEG+1)*6+1)/2),STR(NSEG,NUMP)           HYS00670
      DIMENSION MDIAG(    199     ),DPCOPY((NSEG+1)*6,1)                  HYS00680
      DIMENSION FACTOR((NSEG+1)*6),SEASAT(12,12),TPJT(12,12),TEMP(12)     HYS00690
      DIMENSION DUMMT((NSEG+1)*6,(NSEG+1)*6)                              HYS00700
      DIMENSION R(12,12), PSK(NSEG,12,12), PGLO(12,12)                    HYS00710
      DIMENSION ECCX(2),ECCY(2),A0(NUMP),B(NUMP),FF1((NSEG+1)*6,1)        HYS00720
      DIMENSION DEP(NSEG,NUMP),EOP(NSEG,NUMP),                            HYS00730
     &          STRP(NSEG,NUMP),TEP(NSEG,NUMP),                           HYS00740
     &          EREF(NSEG,NUMP),SREF(NSEG,NUMP)                           HYS00750
      DIMENSION DUC(3),EXCR(6),ESE(3)                                     HYS00760
      DIMENSION RNP(NSEG,12),RIFM((NSEG+1)*6),RMDIAG((NSEG+1)*6+1)       HYS00770
      DIMENSION FJL(NUMP)                                                 HYS00780
      LOGICAL PRINT, BUGS                                                 HYS00790
      REAL INSB, INEH                                                     HYS00800
C                                                                         HYS00810
C                                                                         HYS00820
      BUGS=.FALSE.                                                        HYS00830
      FLP=0                                                               HYS00840
      GO TO (740, 740, 200),IOPT                                          HYS00850
  740 CONTINUE                                                            HYS00860
C                                                                         HYS00870
C DEFINE PARAMETERS                                                       HYS00880
C                                                                         HYS00890
C                                                                         HYS00900
      IDOF=(NSEG+1)*6                                                     HYS00910
      EY=YS/EM                                                            HYS00920
C                                                                         HYS00930
C DEFINE LOCAL BUCKLING FLAG                                              HYS00940
C                                                                         HYS00950
      DO 666 IK=1,NUMP                                                    HYS00960
  666    FJL(IK)=0                                                        HYS00970
C                                                                         HYS00980
C DEFINE SEGMENT GLOBAL AND LOCAL D.O.F. RELATION                         HYS00990
C                                                                         HYS01000
      NP(1,1)=IDOF-12+1                                                   HYS01010
      NP(1,2)=IDOF-12+2                                                   HYS01020
      NP(1,3)=IDOF-12+3                                                   HYS01030
      NP(1,4)=IDOF-12+4                                                   HYS01040
      NP(1,5)=IDOF-12+5                                                   HYS01050
      NP(1,6)=IDOF-12+6                                                   HYS01060
      NP(NSEG,7)=IDOF-12+7                                                HYS01070
      NP(NSEG,8)=IDOF-12+8                                                HYS01080
      NP(NSEG,9)=IDOF-12+9                                                HYS01090
      NP(NSEG,10)=IDOF-12+10                                              HYS01100
      NP(NSEG,11)=IDOF-12+11                                              HYS01110
      NP(NSEG,12)=IDOF-12+12                                              HYS01120
C                                                                         HYS01130
      IF ( NSEG .EQ. 1 ) GO TO 3060                                       HYS01140
      DO 304 I=1,NSEG-1                                                   HYS01150
        DO 304 J=7,12                                                     HYS01160
          K=I*6-6                                                         HYS01170
          NP(I,J)=K+(J-6)                                                 HYS01180
  304 CONTINUE                                                            HYS01190
      DO 306 I=2,NSEG                                                     HYS01200
        DO 306 J=1,6                                                      HYS01210
          K=I*6-6                                                         HYS01220
          NP(I,J)=K+(J-6)                                                 HYS01230
  306 CONTINUE                                                            HYS01240
C                                                                         HYS01250
 3060 CONTINUE                                                            HYS01260
C                                                                         HYS01270
C DEFINE FORCE-MOMENT INDEX                                               HYS01280
C                                                                         HYS01290
C                                                                         HYS01300
      DO 9012 I=1,IDOF                                                    HYS01310
        IFM(I)=0                                                          HYS01320
 9012 CONTINUE                                                            HYS01330
          IFM(IDOF-12+4)=1                                                HYS01340
          IFM(IDOF-12+5)=1                                                HYS01350
          IFM(IDOF-12+6)=1                                                HYS01360
          IFM(IDOF-12+10)=1                                               HYS01370
          IFM(IDOF-12+11)=1                                               HYS01380
          IFM(IDOF-12+12)=1                                               HYS01390
      IF ( IDOF .EQ. 12 ) GO TO 4535                                      HYS01400
      DO 308 I=1,NSEG-1                                                   HYS01410
          J=I*6                                                           HYS01420
          IFM(J-2)=1                                                      HYS01430
          IFM(J-1)=1                                                      HYS01440
          IFM(J  )=1                                                      HYS01450
  308 CONTINUE                                                            HYS01460
 4535 CONTINUE                                                            HYS01470
C                                                                         HYS01480
C DEFINE SECTION POINTS AND SECTIONAL LOCAL COORDINATES                   HYS01490
C                                                                         HYS01500
      CALL LIB(LIBN,HH,UU,WW,ZZ,INEB,INEH,ST,                             HYS01510
     &         IREV1,IREV2,IREV3,IREV4,IECOP,IELNO,                       HYS01520
     &         U0,V0,A0,B,ECCX0,ECCY0,NUMP)                               HYS01530
      DO 50 I=1,NSEG                                                      HYS01540
          BETA(I)=0.                                                      HYS01550
          DO 50 J=1,NUMP                                                  HYS01560
             U(I,J)=U0(J)                                                 HYS01570
             V(I,J)=V0(J)                                                 HYS01580
   50 CONTINUE                                                            HYS01590
      IF( IECOP .EQ. 1 ) THEN                                             HYS01600
             ECX(1)=ECCX0                                                 HYS01610
             ECX(2)=ECCX0                                                 HYS01620
             ECY(1)=ECCY0                                                 HYS01630
             ECY(2)=ECCY0                                                 HYS01640
      ENDIF                                                               HYS01650
      IF( PRINT) THEN                                                     HYS01660
      WRITE(6,543)IELNO                                                   HYS01670
  543 FORMAT('1','THE STABILITY MEMBER ',I5,                              HYS01680
     *       ' SECTIONAL LOCAL COORDINATES:')                             HYS01690
      WRITE(6,544)                                                        HYS01700
  544 FORMAT(1X,' SEC. ELEMENT      U          V       AREA       LENGTH') HYS01710
      DO 545 I=1,NUMP                                                     HYS01720
      WRITE(6,546)I,U0(I),V0(I),A0(I),B(I)                                HYS01730
  546 FORMAT(I5,2X,3G10.4,2X,1P,G10.4)                                    HYS01740
  545 CONTINUE                                                            HYS01750
      ENDIF                                                               HYS01760
C                                                                         HYS01770
C READ EACH NODAL POINTS' GLOBAL COORDINATES                              HYS01780
C                                                                         HYS01790
      DO 203 I=1,NSEG                                                     HYS01800
        DO 203 J=1,2                                                      HYS01810
          DO 203 K=1,3                                                    HYS01820
            GCD(I,J,K)=0.                                                 HYS01830
  203 CONTINUE                                                            HYS01840
      TEMPP=0                                                             HYS01850
      DO 307 I=1,NSEG                                                     HYS01860
        IF(NSEG .EQ. 1 ) THEN                                             HYS01870
             SEL=ELS                                                      HYS01880
             EL(1)=SEL                                                    HYS01890
             GCD(I,1,3)=TEMPP                                             HYS01900
             GCD(I,2,3)=TEMPP+SEL                                         HYS01910
        ELSE IF( I .EQ. 1 .OR. I .EQ. ((NSEG-2)/2)+1                      HYS01920
     &           .OR. I .EQ. ((NSEG-2)/2)+2 .OR. I .EQ. NSEG              HYS01930
     &           .AND. NSEG .NE. 1 ) THEN                                 HYS01940
             SEL=SMALL                                                    HYS01950
             EL(I)=SEL                                                    HYS01960
             GCD(I,1,3)=TEMPP                                             HYS01970
             GCD(I,2,3)=TEMPP+SEL                                         HYS01980
             TEMPP=TEMPP+SEL                                              HYS01990
        ELSE IF( I .NE. 1 .OR. I .NE. ((NSEG-2)/2)+1                      HYS02000
     &           .OR. I .NE. ((NSEG-2)/2)+2 .OR. I .NE. NSEG              HYS02010
     &           .AND. NSEG .NE. 1 ) THEN                                 HYS02020
             SEL=(ELS-4*SMALL)/(NSEG-4)                                   HYS02030
             EL(I)=SEL                                                    HYS02040
             GCD(I,1,3)=TEMPP                                             HYS02050
             GCD(I,2,3)=TEMPP+SEL                                         HYS02060
             TEMPP=TEMPP+SEL                                              HYS02070
        ENDIF                                                             HYS02080
  307 CONTINUE                                                            HYS02090
      IF( PRINT) THEN                                                     HYS02100
      WRITE(6,537)IELNO                                                   HYS02110
  537 FORMAT('1','STABILITY ELEMENT',I5, ' SEGMENT',                      HYS02120
     &       ' END'S COORDINATES IN LONGITUDINAL DIRECTION')              HYS02130
      WRITE(6,538)                                                        HYS02140
  538 FORMAT(1X,' SEGMENT     A-END      B-END')                          HYS02150
      DO 539 I=1,NSEG                                                     HYS02160
      WRITE(6,540)I,GCD(I,1,3),GCD(I,2,3)                                 HYS02170
  540 FORMAT(1X,I5,2G15.6)                                                HYS02180
  539 CONTINUE                                                            HYS02190
      ENDIF                                                               HYS02200
C                                                                         HYS02210
C INITIALIZE THE MEMBER GLOBAL DISPLACEMENT AT EACH D.O.F                 HYS02220
C                                                                         HYS02230
      DO 157 I=1,IDOF                                                     HYS02240
         D1(I,1)=0                                                        HYS02250
         DD(I,1)=0                                                        HYS02260
         DPCOPY(I,1)=0                                                    HYS02270
  157 CONTINUE                                                            HYS02280
C                                                                         HYS02290
C INITIALIZE THE MEMBER STIFFNESS MATRIX                                  HYS02300
C                                                                         HYS02310
      DO 8866 I=1,IDOF                                                    HYS02320
      DO 8866 J=1,IDOF                                                    HYS02330
        ASAT(I,J)=0.                                                      HYS02340
 8866 CONTINUE                                                            HYS02350
C                                                                         HYS02360
C GENERATE THE INITIAL IMPERFECTION DEFLECTION                            HYS02370
C                                                                         HYS02380
      NDOF=IDOF                                                           HYS02390
      PI=3.1415926                                                        HYS02400
      DO 110 I=1,NDOF                                                     HYS02410
        D00(I,1)=0.                                                       HYS02420
  110 CONTINUE                                                            HYS02430
      DO 115 I=1,NSEG                                                     HYS02440
        D00(NP(I,1),1)=(RATIX0*ELS)*SIN(PI*GCD(I,1,3)/ELS)                HYS02450
        D00(NP(I,7),1)=(RATIX0*ELS)*SIN(PI*GCD(I,2,3)/ELS)                HYS02460
        D00(NP(I,2),1)=(RATIY0*ELS)*SIN(PI*GCD(I,1,3)/ELS)                HYS02470
        D00(NP(I,8),1)=(RATIY0*ELS)*SIN(PI*GCD(I,2,3)/ELS)                HYS02480
  115 CONTINUE                                                            HYS02490
      IF(PRINT) THEN                                                      HYS02500
      WRITE(6,534)IELNO                                                   HYS02510
  534 FORMAT(//1X,'THE INITIAL IMPERFECTION OF ELEMENT ',I5,':')          HYS02520
      WRITE(6,734)                                                        HYS02530
  734 FORMAT(1X,' DOF.          IMPERFECTION')                            HYS02540
      DO 535 I=1,IDOF                                                     HYS02550
      WRITE(6,536)I,D00(I,1)                                              HYS02560
  536 FORMAT(1X,I5,8X,1P,G15.6)                                           HYS02570
  535 CONTINUE                                                            HYS02580
      ENDIF                                                               HYS02590
C                                                                         HYS02600
C GENERATE THE SEGMENT LENGTH (INCLUDING THE INITIAL IMPERFECTION         HYS02610
C                                                                         HYS02620
      DO 912 I=1,NSEG                                                     HYS02630
        EL(I)=SQRT(EL(I)**2+(D00(NP(I,8),1)                               HYS02640
     $          -D00(NP(I,2),1))**2                                       HYS02650
     &          +(D00(NP(I,7),1)-D00(NP(I,1),1))**2 )                     HYS02660
        TL(I)=EL(I)                                                       HYS02670
  912 CONTINUE                                                            HYS02680
      IF(PRINT) THEN                                                      HYS02690
      WRITE(6,915)                                                        HYS02700
  915 FORMAT(//1X,'THE SEGMENT LENGTH:')                                  HYS02710
      WRITE(6,916)                                                        HYS02720
  916 FORMAT(1X,'SEGMENT    LENGTH')                                      HYS02730
      WRITE(6,917)(I,EL(I),I=1,NSEG)                                      HYS02740
  917 FORMAT(I5,F15.6)                                                    HYS02750
      ENDIF                                                               HYS02760
C                                                                         HYS02770
C INITIALIZE THE LOCAL DISPLACEMENT AND LOCAL INTERNAL FORCES             HYS02780
C AND GLOBAL INTERNAL FORCE                                               HYS02790
C                                                                         HYS02800
             DO 329 I=1,NSEG                                              HYS02810
               DO 329 J=1,12                                              HYS02820
                  XLOC(I,J)=0.                                            HYS02830
                  FLOC(I,J)=0.                                            HYS02840
                  FGLOB(I,J)=0.                                           HYS02850
                  XGLO(I,J)=0.                                            HYS02860
  329        CONTINUE                                                     HYS02870
C                                                                         HYS02880
C INITIALIZE THE SECTIONAL STRAIN AND BETA VALUES                         HYS02890
C                                                                         HYS02900
      DO 325 I=1,NSEG                                                     HYS02910
           BETA(I)=0.                                                     HYS02920
           DO 325 J=1,NUMP                                                HYS02930
                E0(I,J)=0                                                 HYS02940
                E1(I,J)=EY                                                HYS02950
                E2(I,J)=-EY                                               HYS02960
```

```
                TE(J)*EM                                                        HTS02970
  325 CONTINUE                                                                  HTS02980
C                                                                               HTS02990
      DO 112 I=1,NSEG                                                           HTS03000
C                                                                               HTS03010
C FINE THE ROTATION MATRIX DUE TO INITIAL IMPERFECTION EFFECT                   HTS03020
C                                                                               HTS03030
              PR(I,1)=1.                                                        HTS03040
              PR(I,2)=0.                                                        HTS03050
              PR(I,3)=0.                                                        HTS03060
              PR(I,4)=0.                                                        HTS03070
              PR(I,5)=1.                                                        HTS03080
              PR(I,6)=0.                                                        HTS03090
              PR(I,7)=0.                                                        HTS03100
              PR(I,8)=0.                                                        HTS03110
              PR(I,9)=1.                                                        HTS03120
C                                                                               HTS03130
               XA=D1(NP(I,1),1)+GCD(I,1,1)+D00(NP(I,1),1)                       HTS03140
               YA=D1(NP(I,2),1)+GCD(I,1,2)+D00(NP(I,2),1)                       HTS03150
               ZA=D1(NP(I,3),1)+GCD(I,1,3)+D00(NP(I,3),1)                       HTS03160
               ROXA=D1(NP(I,4),1)+D00(NP(I,4),1)                                HTS03170
               ROYA=D1(NP(I,5),1)+D00(NP(I,5),1)                                HTS03180
               ROZA=D1(NP(I,6),1)+D00(NP(I,6),1)                                HTS03190
               XB=D1(NP(I,7),1)+GCD(I,2,1)+D00(NP(I,7),1)                       HTS03200
               YB=D1(NP(I,8),1)+GCD(I,2,2)+D00(NP(I,8),1)                       HTS03210
               ZB=D1(NP(I,9),1)+GCD(I,2,3)+D00(NP(I,9),1)                       HTS03220
               ROXB=D1(NP(I,10),1)+D00(NP(I,10),1)                              HTS03230
               ROYB=D1(NP(I,11),1)+D00(NP(I,11),1)                              HTS03240
               ROZB=D1(NP(I,12),1)+D00(NP(I,12),1)                              HTS03250
                    DXA=DD(NP(I,1),1)                                           HTS03260
                    DYA=DD(NP(I,2),1)                                           HTS03270
                    DZA=DD(NP(I,3),1)                                           HTS03280
                    DROXA=DD(NP(I,4),1)                                         HTS03290
                    DROYA=DD(NP(I,5),1)                                         HTS03300
                    DROZA=DD(NP(I,6),1)                                         HTS03310
                    DXB=DD(NP(I,7),1)                                           HTS03320
                    DYB=DD(NP(I,8),1)                                           HTS03330
                    DZB=DD(NP(I,9),1)                                           HTS03340
                    DROXB=DD(NP(I,10),1)                                        HTS03350
                    DROYB=DD(NP(I,11),1)                                        HTS03360
                    DROZB=DD(NP(I,12),1)                                        HTS03370
                    IF(PRINT) THEN                                              HTS03380
                    WRITE(6,8088)                                               HTS03390
                    WRITE(6,8089)XA,YA,ZA,ROXA,ROYA,ROZA,                       HTS03400
     *                          XB,YB,ZB,ROXB,ROYB,ROZB,                        HTS03410
     *                          DXA,DYA,DZA,DROXA,DROYA,DROZA,                  HTS03420
     *                          DXB,DYB,DZB,DROXB,DROYB,DROZB                   HTS03430
                    ENDIF                                                       HTS03440
C                                                                               HTS03450
C VARIABLES FOR THE ROTATIONAL MATRIX DUE TO INITIAL IMPERFECTION               HTS03460
C                                                                               HTS03470
      UX0=PR(I,1)                                                               HTS03480
      UY0=PR(I,2)                                                               HTS03490
      UZ0=PR(I,3)                                                               HTS03500
      VX0=PR(I,4)                                                               HTS03510
      VY0=PR(I,5)                                                               HTS03520
      VZ0=PR(I,6)                                                               HTS03530
      WX0=PR(I,7)                                                               HTS03540
      WY0=PR(I,8)                                                               HTS03550
      WZ0=PR(I,9)                                                               HTS03560
C                                                                               HTS03570
C CALCULATE ROTATIONAL MATRIX OF SEGMENT                                        HTS03580
C                                                                               HTS03590
      CALL ROMA (XA,YA,ZA,ROXA,ROYA,ROZA,DXA,DYA,DZA,DROXA,DROYA,DROZA,         HTS03600
     *           XB,YB,ZB,ROXB,ROYB,ROZB,DXB,DYB,DZB,DROXB,DROYB,DROZB,         HTS03610
     *           ROWA0,ROWB0,VX0,VY0,VZ0,WX0,WY0,WZ0,I,                         HTS03620
     *           R,PR,BETA,TL,PRINT,NSEG,IELNO)                                 HTS03630
C                                                                               HTS03640
              TIR(I,1)=R(1,1)                                                   HTS03650
              TIR(I,2)=R(1,2)                                                   HTS03660
              TIR(I,3)=R(1,3)                                                   HTS03670
              TIR(I,4)=R(2,1)                                                   HTS03680
              TIR(I,5)=R(2,2)                                                   HTS03690
              TIR(I,6)=R(2,3)                                                   HTS03700
              TIR(I,7)=R(3,1)                                                   HTS03710
              TIR(I,8)=R(3,2)                                                   HTS03720
              TIR(I,9)=R(3,3)                                                   HTS03730
C                                                                               HTS03740
C GENERATE THE INITIAL ROTATIONAL MATRIX OR PREVIOUS ROTATIONAL                 HTS03750
C MATRIX                                                                        HTS03760
C                                                                               HTS03770
C                                                                               HTS03780
C FIND SEGMENTS' PRINCIPAL AXIS AND MATERIAL PROPERTIES                         HTS03790
C ( SUBROUTINE STRATE AND JUDEL ARE CALLED BY PRINC )                           HTS03800
C                                                                               HTS03810
       CALL PRINC(0., 0., 0., I,PRINT,ST,TOTA,LIBN,                             HTS03820
     &            EM,ET,G,YS,IOPT,NUMP,NSEG,                                    HTS03830
     &            EA,EIU,EIV,GKT,                                               HTS03840
     &            IREV1,IREV2,IREV3,IREV4,UC0,VC0,IELNO,IMATER,                 HTS03850
     &            A0,B,BETA,E0,E1,E2,U,V,U0,V0 ,TE,E,DE,                        HTS03860
     &            INEB,INEH,FJL,                                                HTS03870
     &            RATIO3,IR,STR,DEP,EDP,STRP,TEP,EREF,SREF,FLP)                 HTS03880
C                                                                               HTS03890
C CALCULATE ECCENTRICITIES OF TWO END-SEGMENTS                                  HTS03900
C                                                                               HTS03910
      IF( I .EQ. 1 .OR. I .EQ. NSEG ) THEN                                      HTS03920
         IF( IECOP .EQ. 1 ) THEN                                                HTS03930
            CALL ECCENT(NSEG,UC0,VC0,I,0. ,0. ,ECX,                             HTS03940
     &                  ECY, ECCX, ECCY,PRINT  )                                HTS03950
         ENDIF                                                                  HTS03960
      ENDIF                                                                     HTS03970
C                                                                               HTS03980
C CALCULATE INITIAL ROTATION MATRIX OF SEGMENT                                  HTS03990
C                                                                               HTS04000
               XA=D1(NP(I,1),1)+GCD(I,1,1)+D00(NP(I,1),1)                       HTS04010
               YA=D1(NP(I,2),1)+GCD(I,1,2)+D00(NP(I,2),1)                       HTS04020
               ZA=D1(NP(I,3),1)+GCD(I,1,3)+D00(NP(I,3),1)                       HTS04030
               ROXA=D1(NP(I,4),1)+D00(NP(I,4),1)                                HTS04040
               ROYA=D1(NP(I,5),1)+D00(NP(I,5),1)                                HTS04050
               ROZA=D1(NP(I,6),1)+D00(NP(I,6),1)                                HTS04060
               XB=D1(NP(I,7),1)+GCD(I,2,1)+D00(NP(I,7),1)                       HTS04070
               YB=D1(NP(I,8),1)+GCD(I,2,2)+D00(NP(I,8),1)                       HTS04080
               ZB=D1(NP(I,9),1)+GCD(I,2,3)+D00(NP(I,9),1)                       HTS04090
               ROXB=D1(NP(I,10),1)+D00(NP(I,10),1)                              HTS04100
               ROYB=D1(NP(I,11),1)+D00(NP(I,11),1)                              HTS04110
               ROZB=D1(NP(I,12),1)+D00(NP(I,12),1)                              HTS04120
                    DXA=DD(NP(I,1),1)                                           HTS04130
                    DYA=DD(NP(I,2),1)                                           HTS04140
                    DZA=DD(NP(I,3),1)                                           HTS04150
                    DROXA=DD(NP(I,4),1)                                         HTS04160
                    DROYA=DD(NP(I,5),1)                                         HTS04170
                    DROZA=DD(NP(I,6),1)                                         HTS04180
                    DXB=DD(NP(I,7),1)                                           HTS04190
                    DYB=DD(NP(I,8),1)                                           HTS04200
                    DZB=DD(NP(I,9),1)                                           HTS04210
                    DROXB=DD(NP(I,10),1)                                        HTS04220
                    DROYB=DD(NP(I,11),1)                                        HTS04230
                    DROZB=DD(NP(I,12),1)                                        HTS04240
                    IF(PRINT) THEN                                              HTS04250
                    WRITE(6,8088)                                               HTS04260
                    WRITE(6,8089)XA,YA,ZA,ROXA,ROYA,ROZA,                       HTS04270
     *                          XB,YB,ZB,ROXB,ROYB,ROZB,                        HTS04280
     *                          DXA,DYA,DZA,DROXA,DROYA,DROZA,                  HTS04290
     *                          DXB,DYB,DZB,DROXB,DROYB,DROZB                   HTS04300
                    ENDIF                                                       HTS04310
C                                                                               HTS04320
C VARIABLES FOR THE INITIAL ROTATIONAL MATRIX                                   HTS04330
C                                                                               HTS04340
      UX0=TIR(I,1)                                                              HTS04350
      UY0=TIR(I,2)                                                              HTS04360
      UZ0=TIR(I,3)                                                              HTS04370
      VX0=TIR(I,4)                                                              HTS04380
      VY0=TIR(I,5)                                                              HTS04390
      VZ0=TIR(I,6)                                                              HTS04400
      WX0=TIR(I,7)                                                              HTS04410
      WY0=TIR(I,8)                                                              HTS04420
      WZ0=TIR(I,9)                                                              HTS04430
C                                                                               HTS04440
C CALCULATE ROTATIONAL MATRIX OF SEGMENT                                        HTS04450
C                                                                               HTS04460
      CALL ROMA (XA,YA,ZA,ROXA,ROYA,ROZA,DXA,DYA,DZA,DROXA,DROYA,DROZA,         HTS04470
     *           XB,YB,ZB,ROXB,ROYB,ROZB,DXB,DYB,DZB,DROXB,DROYB,DROZB,         HTS04480
     *           ROWA0,ROWB0,VX0,VY0,VZ0,WX0,WY0,WZ0,I,                         HTS04490
     *           R,PR,BETA,TL,PRINT,NSEG,IELNO)                                 HTS04500
      DEC=0.                                                                    HTS04510
      DCU=0.                                                                    HTS04520
      DCV=0.                                                                    HTS04530
C                                                                               HTS04540
      IF(PRINT) THEN                                                            HTS04550
      WRITE(6,5321)                                                             HTS04560
 532 FORMAT(1X,'SEGMENT ',I5,' THE INITIAL ROTATIONAL MATRIX:')                 HTS04570
      WRITE(6,533)(TIR(I,J),J=1,9)                                              HTS04580
 533 FORMAT(3G15.6)                                                             HTS04590
      ENDIF                                                                     HTS04600
C                                                                               HTS04610
C FORMULATE THE SEGMENT'S LOACL STIFFNESS MATRIX                                HTS04620
C                                                                               HTS04630
       CALL ELESTI( NSEG,I,IELNO,PRINT,ISTIF,                                   HTS04640
     &              EA,EIU,EIV,GKT,EL,FLOC,PSK,SK )                             HTS04650
C                                                                               HTS04660
C TRANSFER SEGMENT'S LOCAL STIFFNESS TO GLOBAL STIFFNESS                        HTS04670
C                                                                               HTS04680
      DO 707 II=1,12                                                            HTS04690
      DO 707 JJ=1,12                                                            HTS04700
        DUMMY(II,JJ)=0                                                          HTS04710
      DO 707 KK=1,12                                                            HTS04720
  707   DUMMY(II,JJ)=DUMMY(II,JJ)+SK(II,KK)*R(KK,JJ)                           HTS04730
C                                                                               HTS04740
      DO 708 II=1,12                                                            HTS04750
      DO 708 JJ=1,12                                                            HTS04760
        RTSKR(II,JJ)=0                                                          HTS04770
      DO 708 KK=1,12                                                            HTS04780
  708   RTSKR(II,JJ)=RTSKR(II,JJ)+R(KK,II)*DUMMY(KK,JJ)                        HTS04790
C                                                                               HTS04800
C ASSEMBLING SEGMENT GLOBAL STIFFNESS TO MEMBER                                 HTS04810
C CENTROID STIFFNESS MATRIX                                                     HTS04820
C                                                                               HTS04830
      DO 85 J=1,12                                                              HTS04840
      DO 85 JJ=1,12                                                             HTS04850
        ASAT(NP(I,J),NP(I,JJ))=ASAT(NP(I,J),NP(I,JJ))+RTSKR(J,JJ)              HTS04860
   85 CONTINUE                                                                  HTS04870
C                                                                               HTS04880
  112 CONTINUE                                                                  HTS04890
C                                                                               HTS04900
C PRINT OUT THE STABILITY MEMBER STIFFNESS MATRIX                               HTS04910
C                                                                               HTS04920
      IF( PRINT ) THEN                                                          HTS04930
      WRITE(6,1200)IELNO                                                        HTS04940
 1200 FORMAT('1',' THE STABILITY MEMBER',I3,' ASAT MATRIX:'/)                   HTS04950
      DO 1300 I=1,IDOF                                                          HTS04960
      WRITE(6,1340)(ASAT(I,J),J=1,IDOF)                                         HTS04970
 1340 FORMAT(1X,6G12.6)                                                         HTS04980
 1300 CONTINUE                                                                  HTS04990
      ENDIF                                                                     HTS05000
C                                                                               HTS05010
C MEMORIZE THE STABILITY ELEMENT STIFFNESS                                      HTS05020
C                                                                               HTS05030
      DO 986 I=1,IDOF                                                           HTS05040
        DO 986 J=1,IDOF                                                         HTS05050
          EEASAT(I,J)=ASAT(I,J)                                                 HTS05060
  986 CONTINUE                                                                  HTS05070
C                                                                               HTS05080
C CONVERT ASAT MATRIX TO HALF STORAGE AND BANDED MODE                           HTS05090
C                                                                               HTS05100
      NDOST=IDOF                                                                HTS05110
      JSTOR=NDOST*(NDOST+1)/2                                                   HTS05120
C                                                                               HTS05130
      CALL SKY(ASAT,STF,NDOST,JSTOR,IDOF,MDIAG,IMD)                             HTS05140
C                                                                               HTS05150
C FIND CONDENSED STIFFNESS MATRIX CORRESPONDING TO TWO-END 12 DOF               HTS05160
C                                                                               HTS05170
      IF ( IDOF .EQ. 12 ) GO TO 4532                                            HTS05180
      L1=1                                                                      HTS05190
      L2=IDOF-12                                                                HTS05200
      CALL GAUSS2(1, PRINT,IDOF,IMD,NDOST,1,L1,L2,MDIAG,STF,DPCOPY,            HTS05210
     $            FACTOR,JSTOR)                                                 HTS05220
C                                                                               HTS05230
 4532 CONTINUE                                                                  HTS05240
C                                                                               HTS05250
C ============================================================                  HTS05260
C CONDENSE ASAT MATRIX TO GET CONDENSED MATRIX CORRESPONDING TO                 HTS05270
C TWO ENDS TWELVE DOF DIRECTIONS                                                HTS05280
C ============================================================                  HTS05290
C                                                                               HTS05300
      L1=IDOF-12+1                                                              HTS05310
      L2=IDOF                                                                   HTS05320
      CALL CONVER(PRINT,STF,EEASAT,NDOST,JSTOR,IDOF,IMD,                        HTS05330
     $            12,L1,L2,MDIAG)                                               HTS05340
C                                                                               HTS05350
      IF( IECOP .EQ. 1 ) THEN                                                   HTS05360
        CALL TRANK(EEASAT,ECCX,ECCY,TPJT)                                       HTS05370
        IF(PRINT) THEN                                                          HTS05380
          WRITE(6,*)'1',' THE FINAL EEASAT MATRIX AFTER CALL TRANK:'            HTS05390
          CALL PRIN( EEASAT,12,12,12, 12)                                       HTS05400
        ENDIF                                                                   HTS05410
      ENDIF                                                                     HTS05420
C                                                                               HTS05430
      DO 365 I=1,NSEG                                                           HTS05440
      DO 365 J=1,12                                                             HTS05450
  365   RNP(I,J)=NP(I,J)                                                        HTS05460
      DO 326 I=1,(NSEG+1)*6                                                     HTS05470
  326   RIFM(I)=IFM(I)                                                          HTS05480
      DO 327 I=1,(NSEG+1)*6+1                                                   HTS05490
  327   RMDIAG(I)=MDIAG(I)                                                      HTS05500
C                                                                               HTS05510
      RETURN                                                                    HTS05520
C                                                                               HTS05530
C------------------- ( IOPT =2 ) ------------------------------------------    HTS05540
  200 CONTINUE                                                                  HTS05550
C                                                                               HTS05560
C DEFINE PARAMETERS                                                             HTS05570
C                                                                               HTS05580
      IDOF=(NSEG+1)*6                                                           HTS05590
      NDOF=IDOF                                                                 HTS05600
      NDOST=IDOF                                                                HTS05610
      JSTOR=NDOST*(NDOST+1)/2                                                   HTS05620
      EY=YS/EM                                                                  HTS05630
C     RNEEE=0                                                                   HTS05640
      IF( ISTART .LT. 5 ) THEN                                                  HTS05650
         ISTART=ISTART+1                                                        HTS05660
      ELSE                                                                      HTS05670
         ISTART=5                                                               HTS05680
      ENDIF                                                                     HTS05690
C                                                                               HTS05700
      DO 425 I=1,NSEG                                                           HTS05710
      DO 425 J=1,12                                                             HTS05720
  425   NP(I,J)=RNP(I,J)                                                        HTS05730
      DO 426 I=1,(NSEG+1)*6                                                     HTS05740
        FACTOR(I)=0                                                             HTS05750
  426   IFM(I)=RIFM(I)                                                          HTS05760
      DO 427 I=1,(NSEG+1)*6+1                                                   HTS05770
  427   MDIAG(I)=RMDIAG(I)                                                      HTS05780
                                                                                HTS05790
                                                                                HTS05800
```

```
C
C ***************************************
C ** INITIALIZE THE MEMBER'S ASAT MATRIX **
C ***************************************
C
      DO 17 I=1,NDOF
      DO 17 J=1,NDOF
        ASAT(I,J)=0.
   17 CONTINUE
C
C DISPLACEMENT INCREMENT TRANSFORMATION DUE TO ECCENTRIC LOADS
C
      IF( IECOP .EQ. 1 ) THEN
        DO 913 I=1,12
          TEMP(I)=DD(I,1)
  913   CONTINUE
        DO 525 II=1,12
          DD(II,1)=0.
          DO 525 KK=1,12
            DD(II,1)=DD(II,1)+TPJT(II,KK)*TEMP(KK)
  525   CONTINUE
      ENDIF
C
C CONVERT MEMBER END DISPLACEMENT INCREMENT TO THE CORRESPONDING
C MEMBER'S DOF DIRECTION
C
        DO 908 I=1,12
          TEMP(I)=DD(I,1)
  908   CONTINUE
      DO 837 I=1,12
        DD(IDOF-12+I,1)=TEMP(I)
  837 CONTINUE
C
C BACK SUBSTITUTE TO GET MEMBER'S INTERNAL DOF DISPLACEMENT INCREMENT
C
      L1=IDOF-12+1
      L2=IDOF
C
      DO 838 I=L1,L2
  838   FACTOR(I)=DD(I,1)
      IMD=MDIAG(IDOF+1)-1
C
      CALL GAUSS2(4, PRINT,IDOF,IMD,NDOST,1,L1,L2,MDIAG,STF,
     $            DPCOPT,FACTOR,JSTOR)
C
      DO 901 I=1,L1-1
        DP(I,1)=0
  901   DPCOPT(I,1)=0
      DO 902 I=L1,L2
  902   DP(I,1)=DPCOPT(I,1)
C
C     CALL GAUSS2(1, PRINT,IDOF,IMD,NDOST,1,L1,L2,MDIAG,STF,
C    $            DPCOPT,FACTOR,JSTOR)
C     CALL GAUSS2(3, PRINT,IDOF,IMD,NDOST,1,1,IDOF,MDIAG,STF,
C    $            DPCOPT,FACTOR,JSTOR)
C
      DO 848 I=IDOF-12+1,IDOF
  848   FACTOR(I)=DD(I,1)
C
      IF ( IDOF .EQ. 12 ) GO TO 4544
C
C     ------------------------------
C     === BACK SUBSTITUTE BEGIN ===
C     ------------------------------
      L1=1
      L2=IDOF-12
      CALL GAUSS2(5, PRINT,IDOF,IMD,NDOST,1,1,L2,MDIAG,STF,
     $            DPCOPT,FACTOR,JSTOR)
C
 4544 CONTINUE
C
      DO 903 I=1,IDOF
  903   DD(I,1)=FACTOR(I)
C
C
      DO 219 I=1,NSEG
      DO 219 J=1,12
        XGLO(I,J)=DD(NP(I,J),1)
  219 CONTINUE
C
C ADD GLOBAL DISPLACEMENT RATE TO TOTAL DISPLACEMENT
C
      DO 1109 I=1,IDOF
        D1(I,1)=D1(I,1)+DD(I,1)
 1109 CONTINUE
C
C PRINT OUT STABILITY MEMBER INTERNAL DISPLACEMENT
C
      IF( PRINT) THEN
        WRITE(6,8855)IELNO
 8855   FORMAT('1','THE STABILITY MEMBER ',I3,' INTERNAL DISPLACEMENT:')
        WRITE(6,873)(D1(I,1),I=1,IDOF)
  873   FORMAT(1X,6G13.6)
      ENDIF
C
C FIND EACH SEGMENT INTERNAL FORCE IN GLOBAL DIRECTIONS
C
C
C ADJUST THE INITIAL IMPERFECTION DEFLECTION FOR NEW CYCLE, ISPE=1
C
      IF( ISPE .EQ. 1 ) THEN
      PI=3.14159265
      DO 210 I=1,NDOF
        DD0(I,1)=0.
  210 CONTINUE
      ELS=0.
      DO 211 I=1,NSEG
        SZA=D1(NP(I,3),1)+GCD(I,1,3)
        SZB=D1(NP(I,9),1)+GCD(I,2,3)
        EL(I)=SZB-SZA
  211   ELS=ELS+EL(I)
C
C     ------------------------------------------------------------------
C     ASSUME INITIAL IMPERFECTION RATIO BASED ON MEMBER MIDDLE POINT
C     DISPLACEMENT
C     ------------------------------------------------------------------
        IDUM=((NSEG/2)-1)*6 + 1
        RATIX0=D1(IDUM,1)/ELS
        RATIY0=D1(IDUM+1,1)/ELS
C
      DO 215 I=1,NSEG
        DD0(NP(I,1),1)=(RATIX0*EL6)*SIN(PI*
     &  (D1(NP(I,3),1)+GCD(I,1,3)-D1(NP(1,3),1))/ELS)
        DD0(NP(I,7),1)=(RATIX0*ELS)*SIN(PI*
     &  (D1(NP(I,9),1)+GCD(I,2,3)-D1(NP(1,3),1))/ELS)
        DD0(NP(I,2),1)=(RATIY0*ELS)*SIN(PI*
     &  (D1(NP(I,3),1)+GCD(I,1,3)-D1(NP(1,3),1))/ELS)
        DD0(NP(I,8),1)=(RATIY0*ELS)*SIN(PI*
     &  (D1(NP(I,9),1)+GCD(I,2,3)-D1(NP(1,3),1))/ELS)
  215 CONTINUE
      IF(PRINT) THEN
        WRITE(6,534)
        WRITE(6,734)
        DO 235 I=1,IDOF
          WRITE(6,536)I,DD0(I,1)
  235   CONTINUE
      ENDIF
C
C GENERATE THE SEGMENT LENGTH (INCLUDING THE INITIAL IMPERFECTION
C
      DO 922 I=1,NSEG
        EL(I)=SQRT(EL(I)**2+(DD0(NP(I,8),1)
     $        -DD0(NP(I,2),1))**2
     &        +(DD0(NP(I,7),1)-DD0(NP(I,1),1))**2 )
        TL(I)=EL(I)
  922 CONTINUE
      WRITE(6,915)
      WRITE(6,916)
      WRITE(6,917)(I,EL(I),I=1,NSEG)
      ENDIF
C
C ***********************************************************
C     FIND EACH SEGMENT INTERNAL FORCE IN GLOBAL DIRECTIONS
C ***********************************************************
      DO 90 I=1,NSEG
        CALL CLDAF(R,PR,I,NSEG,XGLO,PSK,SK,
     &             FGLO,FGLOB,PRINT)
C
C FIND SEGMENT DEFORMATION AND INCREMENTS IN MEMBER COORDINATES
C DIRECTIONS
C
        IF ( ISPE .EQ. 1 ) THEN
          XA=0.                    +GCD(I,1,1)+DD0(NP(I,1),1)
          YA=0.                    +GCD(I,1,2)+DD0(NP(I,2),1)
          ZA=D1(NP(I,3),1)+GCD(I,1,3)+DD0(NP(I,3),1)
          ROXA=0.                  +DD0(NP(I,4),1)
          ROYA=0.                  +DD0(NP(I,5),1)
          ROZA=D1(NP(I,6),1)+DD0(NP(I,6),1)
          XB=0.                    +GCD(I,2,1)+DD0(NP(I,7),1)
          YB=0.                    +GCD(I,2,2)+DD0(NP(I,8),1)
          ZB=D1(NP(I,9),1)+GCD(I,2,3)+DD0(NP(I,9),1)
          ROXB=0.                  +DD0(NP(I,10),1)
          ROYB=0.                  +DD0(NP(I,11),1)
          ROZB=D1(NP(I,12),1)+DD0(NP(I,12),1)
              DXA=0.
              DYA=0.
              DZA=DD(NP(I,3),1)
              DROXA=0.
              DROYA=0.
              DROZA=DD(NP(I,6),1)
              DXB=0.
              DYB=0.
              DZB=DD(NP(I,9),1)
              DROXB=0.
              DROYB=0.
              DROZB=DD(NP(I,12),1)
              D1(NP(I,1),1)=0
              D1(NP(I,2),1)=0
              D1(NP(I,4),1)=0
              D1(NP(I,5),1)=0
              D1(NP(I,7),1)=0
              D1(NP(I,8),1)=0
              D1(NP(I,10),1)=0
              D1(NP(I,11),1)=0
        ELSE IF ( ISPE .EQ. 0 ) THEN
          XA=D1(NP(I,1),1)+GCD(I,1,1)+DD0(NP(I,1),1)
          YA=D1(NP(I,2),1)+GCD(I,1,2)+DD0(NP(I,2),1)
          ZA=D1(NP(I,3),1)+GCD(I,1,3)+DD0(NP(I,3),1)
          ROXA=D1(NP(I,4),1)+DD0(NP(I,4),1)
          ROYA=D1(NP(I,5),1)+DD0(NP(I,5),1)
          ROZA=D1(NP(I,6),1)+DD0(NP(I,6),1)
          XB=D1(NP(I,7),1)+GCD(I,2,1)+DD0(NP(I,7),1)
          YB=D1(NP(I,8),1)+GCD(I,2,2)+DD0(NP(I,8),1)
          ZB=D1(NP(I,9),1)+GCD(I,2,3)+DD0(NP(I,9),1)
          ROXB=D1(NP(I,10),1)+DD0(NP(I,10),1)
          ROYB=D1(NP(I,11),1)+DD0(NP(I,11),1)
          ROZB=D1(NP(I,12),1)+DD0(NP(I,12),1)
              DXA=DD(NP(I,1),1)
              DYA=DD(NP(I,2),1)
              DZA=DD(NP(I,3),1)
              DROXA=DD(NP(I,4),1)
              DROYA=DD(NP(I,5),1)
              DROZA=DD(NP(I,6),1)
              DXB=DD(NP(I,7),1)
              DYB=DD(NP(I,8),1)
              DZB=DD(NP(I,9),1)
              DROXB=DD(NP(I,10),1)
              DROYB=DD(NP(I,11),1)
              DROZB=DD(NP(I,12),1)
         ENDIF
              IF(PRINT) THEN
              WRITE(6,8088)
 8088         FORMAT('1','XA,YA,ZA,ROXA,ROYA,ROZA,',
     *                   'XB,YB,ZB,ROXB,ROYB,ROZB,',
     *                   'DXA,DYA,DZA,DROXA,DROYA,DROZA,',
     *                   'DXB,DYB,DZB,DROXB,DROYB,DROZB'/)
              WRITE(6,8089)XA,YA,ZA,ROXA,ROYA,ROZA,
     *                     XB,YB,ZB,ROXB,ROYB,ROZB,
     *                     DXA,DYA,DZA,DROXA,DROYA,DROZA,
     *                     DXB,DYB,DZB,DROXB,DROYB,DROZB
 8089         FORMAT(1X,6F15.6)
              ENDIF
C
C VARIABLES FOR THE INITIAL ROTATIONAL MATRIX
C
      UX0=TIR(1,1)
      UY0=TIR(1,2)
      UZ0=TIR(1,3)
      VX0=TIR(1,4)
      VY0=TIR(1,5)
      VZ0=TIR(1,6)
      WX0=TIR(1,7)
      WY0=TIR(1,8)
      WZ0=TIR(1,9)
C
C CALCULATE SEGMENT CURVATURE RATE IN PRINCIPAL AXES DIRECTIONS
C
      CALL CURVA( NSEG,I,DEC,DCU,DCV,PRINT,R,EL,PR,
     &            DXA,DYA,DZA,DROXA,DROYA,DROZA,
     &            DXB,DYB,DZB,DROXB,DROYB,DROZB )
C     WRITE(6,*)I,' SEGMENT, DEC,DCU,DCV: ',DEC,DCU,DCV
C
C CALCULATE EACH SEGMEN'S PRINCIPAL AXIS AND MATERIAL PROPERTIES
C
        CALL PRINC(DEC, DCU, DCV, I,PRINT,ST,TOTA,LIBN,
     &             EM,ET,G,TS,IOPT,NUMP,NSEG,
     &             EA,EIU,EIV,GKT,
     &             IREV1,IREV2,IREV3,IREV4,UC0,VC0,IELNO,IMATER,
     &             A0,B,BETA,E0,E1,E2,U,V,U0,V0,TE,E,DE,
     &             INEB,INEH,FJL,
     &             RATIO3,IR,STR,DEP,EOP,STRP,TEP,EREF,SREF,FLP)
C
C CALCULATE ECCENTRICITIES OF TWO END-SEGMENTS
C
      IF( I .EQ. 1 .OR. I .EQ. NSEG ) THEN
        IF( IECOP .EQ. 1 ) THEN
 888      FORMAT(1X,1P,6I5.4)
          CALL ECCENT(NSEG,UC0,VC0,I,ROZA ,ROZB,ECX,
     &                ECY, ECCX, ECCY,PRINT  )
        ENDIF
      ENDIF
C
C     CALCULATE SEGMENT CURRENT ROTATIONAL MATRIX
C
      CALL ROMA (XA,YA,ZA,ROXA,ROYA,ROZA,DXA,DYA,DZA,DROXA,DROYA,DROZA,
     *           XB,YB,ZB,ROXB,ROYB,ROZB,DXB,DYB,DZB,DROXB,DROYB,DROZB,
     *           ROWA0,ROWB0,VX0,VY0,VZ0,WX0,WY0,WZ0,I,
     *           R,PR,BETA,TL,PRINT,NSEG,IELNO)
C
C CALCULATE EACH SEGMENT LOCAL FORCE
C
      CALL CSLF(R,NSEG,I,FLOC,FGLO, PRINT)
C
C CALCULATE ELEMENT STIFFNESS MATRIX
```

```
C                                                                       HYS08650
      CALL ELESTI( NSEG,I,IELNO,PRINT,ISTIF,                            HYS08660
     &             EA,EIU,EIV,GKT,EL,FLOC,PSK,SK )                      HYS08670
C                                                                       HYS08680
C CALCULATE SYSTEM STIFFNESS MATRIX                                     HYS08690
C                                                                       HYS08700
      DO 807 II=1,12                                                    HYS08710
      DO 807 JJ=1,12                                                    HYS08720
        DUMMY(II,JJ)=0                                                  HYS08730
      DO 807 KK=1,12                                                    HYS08740
  807   DUMMY(II,JJ)=DUMMY(II,JJ)+SK(II,KK)*R(KK,JJ)                     HYS08750
C                                                                       HYS08760
      DO 808 II=1,12                                                    HYS08770
      DO 808 JJ=1,12                                                    HYS08780
        RTSKR(II,JJ)=0                                                  HYS08790
      DO 808 KK=1,12                                                    HYS08800
  808   RTSKR(II,JJ)=RTSKR(II,JJ)+R(KK,II)*DUMMY(KK,JJ)                  HYS08810
      IF(PRINT) THEN                                                    HYS08820
        WRITE(6,*)'1',' THE SEGMENT RTSKR MATRIX:'                      HYS08830
      CALL PRIN(RTSKR,12,12,12,12)                                      HYS08840
      ENDIF                                                             HYS08850
C                                                                       HYS08860
C ASSEMBLING SEGMENT GLOBAL STIFFNESS TO MEMBER                         HYS08870
C CENTROID STIFFNESS MATRIX                                             HYS08880
C                                                                       HYS08890
      DO 805 J=1,12                                                     HYS08900
      DO 805 JJ=1,12                                                    HYS08910
          ASAT(NP(I,J),NP(I,JJ))=ASAT(NP(I,J),NP(I,JJ))+RTSKR(J,JJ)     HYS08920
  805 CONTINUE                                                          HYS08930
C                                                                       HYS08940
   90 CONTINUE                                                          HYS08950
C                                                                       HYS08960
C CALCULATE MEMBER'S LOAD INCREMENTS BASED ON [K]*DD                    HYS08970
C                                                                       HYS08980
      DO 5021 II=1,IDOF                                                 HYS08990
         DP(II,1)=0.                                                    HYS09000
      DO 5021 K=1,IDOF                                                  HYS09010
         DP(II,1)=DP(II,1)+SEASAT(II,K)*DD(K,1)                         HYS09020
 5021 CONTINUE                                                          HYS09030
C                                                                       HYS09040
C CALCULATE MEMBER'S TOTAL FORCE IN THE GLOBAL DIRECTION                HYS09050
C                                                                       HYS09060
      CALL FORCE(FF1,PRINT, IDOF,IELNO,NDOST,NSEG,                      HYS09070
     &           FGLOB,NP )                                             HYS09080
C                                                                       HYS09090
C     ================================================                  HYS09100
C     CALCULATE STABILITY MEMBER'S STIFFNESS PARAMETER AND              HYS09110
C     DISPLACEMENT NORM                                                 HYS09120
C     ================================================                  HYS09130
C                                                                       HYS09140
      IF ( IAUTO .EQ. 1 ) THEN                                          HYS09150
C                                                                       HYS09160
        DO 977 MM=1,IDOF                                                HYS09170
          DPCOPY(MM,1)=D1(MM,1)                                         HYS09180
        DO 977 NN=1,IDOF                                                HYS09190
  977     DUMMY(MM,NN)=SEASAT(MM,NN)                                    HYS09200
C                                                                       HYS09210
        IF ( ISTART .EQ. 2 ) THEN                                       HYS09220
          CALL SNORM(1,DPCOPY,FF1,DD,DP,DUMMY,DELTP2,S2,                HYS09230
     &               IDOF,IFM,NDOST,1,1,NDOST)                          HYS09240
        ENDIF                                                           HYS09250
C                                                                       HYS09260
        IF ( ISTART .GE. 2 ) THEN                                       HYS09270
          CALL SNORM(1,DPCOPY,FF1,DD,DP,DUMMY,DELTPJ,SJ,IDOF,IFM,      HYS09280
     &               NDOST,1,1,NDOST)                                   HYS09290
          CALL SNORM(2,DPCOPY,FF1,DD,DP,DUMMY,EEE,DMY,IDOF,IFM,        HYS09300
     &               NDOST,1,1,NDOST)                                   HYS09310
          SP=((DELTPJ/DELTP2)**2)*(S2/SJ)                               HYS09320
C                                                                       HYS09330
          IF(SP.LE. 0 .AND. PRINT) THEN                                 HYS09340
           WRITE(6,8944)SP                                              HYS09350
 8944      FORMAT(30X,' *** STABILITY MEMBER ',I2,' BUCKED, SP= ',F10.4)HYS09360
          ENDIF                                                         HYS09370
C         IF( SP .GT. 1.5 ) THEN                                        HYS09380
C          WRITE(6,8943)IELNO                                           HYS09390
C8943      FORMAT(30X,' *** STABILITY MEMBER ',I2,' SP .GT. 1.5 ! ')    HYS09400
C         ENDIF                                                         HYS09410
C                                                                       HYS09420
          RNEEE=ABS(EEE/QRNEE)                                          HYS09430
C                                                                       HYS09440
          IF( PRINT) THEN                                               HYS09450
           WRITE(6,9843)EEE,RNEEE                                       HYS09460
 9843      FORMAT(1X,'*** STABILITY MEMBER ',I2,' DISP. NORM =',        HYS09470
     &            F10.6,'RNEEE=',F10.6)                                 HYS09480
          ENDIF                                                         HYS09490
        ENDIF                                                           HYS09500
      ENDIF                                                             HYS09510
C                                                                       HYS09520
C MEMORIZE THE STABILITY ELEMENT STIFFNESS                              HYS09530
C                                                                       HYS09540
      DO 987 I=1,IDOF                                                   HYS09550
        DO 987 J=1,IDOF                                                 HYS09560
         SEASAT(I,J)=ASAT(I,J)                                          HYS09570
  987 CONTINUE                                                          HYS09580
C                                                                       HYS09590
C CONVERT ASAT MATRIX TO HALF STORAGE AND BANDED MODE                   HYS09600
C                                                                       HYS09610
      CALL SKT(ASAT,STF,NDOST,JSTOR,IDOF,MDIAG,IMD)                     HYS09620
C                                                                       HYS09630
C FIND CONDENSED STIFFNESS MATRIX CORRESPONDING TO TWO-END 12 DOF       HYS09640
C                                                                       HYS09650
      IF ( IDOF .EQ. 12 ) GO TO 4537                                    HYS09660
      L1=1                                                              HYS09670
      L2=IDOF-12                                                        HYS09680
      CALL GAUSS2(1, PRINT,IDOF,IMD,NDOST,1,L1,L2,MDIAG,STF,DPCOPY,     HYS09690
     $            FACTOR,JSTOR)                                         HYS09700
C                                                                       HYS09710
 4537 CONTINUE                                                          HYS09720
C                                                                       HYS09730
C CONDENSE ASAT MATRIX TO GET CONDENSED MATRIX CORRESPONDING TO         HYS09740
C TWO ENDS TWELVE DOF DIRECTIONS                                        HYS09750
C                                                                       HYS09760
      L1=IDOF-12+1                                                      HYS09770
      L2=IDOF                                                           HYS09780
      CALL CONVER(PRINT,STF,SEASAT,NDOST,JSTOR,IDOF,IMD,                HYS09790
     $            12,L1,L2,MDIAG)                                       HYS09800
C                                                                       HYS09810
      IF( IECOP .EQ. 1 ) THEN                                           HYS09820
        CALL TRANK(SEASAT,ECCX,ECCY,TPJT)                               HYS09830
        IF(PRINT) THEN                                                  HYS09840
          WRITE(6,*)'1',' THE FINAL SEASAT MATRIX AFTER CALL TRANK:'    HYS09850
          CALL PRIN( SEASAT,12,12,12, 12)                               HYS09860
        ENDIF                                                           HYS09870
      ENDIF                                                             HYS09880
C                                                                       HYS09890
      DO 575 I=1,NSEG                                                   HYS09900
      DO 575 J=1,12                                                     HYS09910
  575   RNP(I,J)=NP(I,J)                                                HYS09920
      DO 526 I=1,(NSEG+1)*6                                             HYS09930
  526   RIFM(I)=IFM(I)                                                  HYS09940
      DO 527 I=1,(NSEG+1)*6+1                                           HYS09950
  527   RMDIAG(I)=MDIAG(I)                                              HYS09960
      RETURN                                                            HYS09970
      END                                                               HYS09980
C.pa                                                                    HYS09990
C     DEBUG UNIT(6),SUBCHK                                              HYS10000
C     END DEBUG                                                         HYS10010
C                                                                       HYS10020
C-----------------------------------------------------------------------HYS10030
C     SUBROUTINE ROMA                                                   HYS10040
C-----------------------------------------------------------------------HYS10050
C                                                                       HYS10060
      SUBROUTINE ROMA (XA,YA,ZA,ROXA,ROYA,ROZA,                         HYS10070
     *                 DXA,DYA,DZA,DROXA,DROYA,DROZA,                   HYS10080
     *                 XB,YB,ZB,ROXB,ROYB,ROZB,                         HYS10090
     *                 DXB,DYB,DZB,DROXB,DROYB,DROZB,                   HYS10100
     *                 ROWA0,ROWB0,VX0,VY0,VZ0,WX0,WY0,WZ0,ISEG,        HYS10110
     *                 R,PR,BETA,TL,PRINT,NSEG,IELNO)                   HYS10120
      DIMENSION SR(3,3),DA(12,12),DDA(12,12),R(12,12)                   HYS10130
      DIMENSION DB(12,12),DDB(12,12)                                    HYS10140
      DIMENSION PR(NSEG,9),BETA(NSEG),TL(NSEG)                          HYS10150
      LOGICAL PRINT                                                     HYS10160
C                                                                       HYS10170
      TL(ISEG)=SQRT((XB-XA)**2+(YB-YA)**2+(ZB-ZA)**2)                   HYS10180
      WX=(XB-XA)/TL(ISEG)                                               HYS10190
      WY=(YB-YA)/TL(ISEG)                                               HYS10200
      WZ=(ZB-ZA)/TL(ISEG)                                               HYS10210
      ROWBA=BETA(ISEG)+0.5*(ROZA+ROZB)                                  HYS10220
      S=SIN(ROWBA)                                                      HYS10230
      C=COS(ROWBA)                                                      HYS10240
      VBAX=VX0*C-(VY0*WZ0-VZ0*WY0)*S                                    HYS10250
      VBAY=VY0*C-(VZ0*WX0-VX0*WZ0)*S                                    HYS10260
      VBAZ=VZ0*C-(VX0*WY0-VY0*WX0)*S                                    HYS10270
      TM=VBAX*WX+VBAY*WY+VBAZ*WZ                                        HYS10280
      TLV=SQRT((VBAX-TM*WX)**2+(VBAY-TM*WY)**2+(VBAZ-TM*WZ)**2)         HYS10290
      VX=(VBAX-TM*WX)/TLV                                               HYS10300
      VY=(VBAY-TM*WY)/TLV                                               HYS10310
      VZ=(VBAZ-TM*WZ)/TLV                                               HYS10320
      UX=VY*WZ-VZ*WY                                                    HYS10330
      UY=VZ*WX-VX*WZ                                                    HYS10340
      UZ=VX*WY-VY*WX                                                    HYS10350
      SR(1,1)=UX                                                        HYS10360
      SR(1,2)=UY                                                        HYS10370
      SR(1,3)=UZ                                                        HYS10380
      SR(2,1)=VX                                                        HYS10390
      SR(2,2)=VY                                                        HYS10400
      SR(2,3)=VZ                                                        HYS10410
      SR(3,1)=WX                                                        HYS10420
      SR(3,2)=WY                                                        HYS10430
      SR(3,3)=WZ                                                        HYS10440
      IF( PRINT) THEN                                                   HYS10450
      WRITE(6,344)                                                      HYS10460
  344 FORMAT(1X,'THE ROTATIONAL MATRIX:')                               HYS10470
      CALL PRIN(SR,3,3,3,3)                                             HYS10480
      ENDIF                                                             HYS10490
C                                                                       HYS10500
C FIND THE ROTATION MATRIX AND THE SECTION PROPERTIES                   HYS10510
C                                                                       HYS10520
      DO 52 I=1,12                                                      HYS10530
      DO 52 J=1,12                                                      HYS10540
           R(I,J)=0.                                                    HYS10550
   52 CONTINUE                                                          HYS10560
      I=1                                                               HYS10570
   72 R(I,I)=UX                                                         HYS10580
      R(I,I+1)=UY                                                       HYS10590
      R(I,I+2)=UZ                                                       HYS10600
      R(I+1,I)=VX                                                       HYS10610
      R(I+1,I+1)=VY                                                     HYS10620
      R(I+1,I+2)=VZ                                                     HYS10630
      R(I+2,I)=WX                                                       HYS10640
      R(I+2,I+1)=WY                                                     HYS10650
      R(I+2,I+2)=WZ                                                     HYS10660
      I=I+3                                                             HYS10670
      IF (I.EQ.13) GO TO 62                                             HYS10680
      GO TO 72                                                          HYS10690
   62 PR(ISEG,1)=UX                                                     HYS10700
      PR(ISEG,2)=UY                                                     HYS10710
      PR(ISEG,3)=UZ                                                     HYS10720
      PR(ISEG,4)=VX                                                     HYS10730
      PR(ISEG,5)=VY                                                     HYS10740
      PR(ISEG,6)=VZ                                                     HYS10750
      PR(ISEG,7)=WX                                                     HYS10760
      PR(ISEG,8)=WY                                                     HYS10770
      PR(ISEG,9)=WZ                                                     HYS10780
C                                                                       HYS10790
      RETURN                                                            HYS10800
      END                                                               HYS10810
                                                                        HYS10820
C.pa                                                                    HYS10830
C     DEBUG UNIT(6),SUBCHK                                              HYS10840
C     END DEBUG                                                         HYS10850
      SUBROUTINE CURVA( NSEG,ISEG,DEC,DCU,DCV,PRINT,R,EL,PR,            HYS10860
     &                  DXA,DYA,DZA,DROXA,DROYA,DROZA,                  HYS10870
     &                  DXB,DYB,DZB,DROXB,DROYB,DROZB )                 HYS10880
      DIMENSION R(12,12),DA(12,12),DDA(12,12),DB(12,12),DDB(12,12)      HYS10890
      DIMENSION PR(NSEG,9),EL(NSEG)                                     HYS10900
      LOGICAL PRINT                                                     HYS10910
C                                                                       HYS10920
      DO 52 I=1,12                                                      HYS10930
      DO 52 J=1,12                                                      HYS10940
        R(I,J)=0.                                                       HYS10950
   52 CONTINUE                                                          HYS10960
C                                                                       HYS10970
      I=1                                                               HYS10980
   72 R(I,I)=PR(ISEG,1)                                                 HYS10990
      R(I,I+1)=PR(ISEG,2)                                               HYS11000
      R(I,I+2)=PR(ISEG,3)                                               HYS11010
      R(I+1,I)=PR(ISEG,4)                                               HYS11020
      R(I+1,I+1)=PR(ISEG,5)                                             HYS11030
      R(I+1,I+2)=PR(ISEG,6)                                             HYS11040
      R(I+2,I)=PR(ISEG,7)                                               HYS11050
      R(I+2,I+1)=PR(ISEG,8)                                             HYS11060
      R(I+2,I+2)=PR(ISEG,9)                                             HYS11070
      I=I+3                                                             HYS11080
      IF( I .EQ. 13 ) GO TO 62                                          HYS11090
      GO TO 72                                                          HYS11100
   62 DA(1,1)=DXA                                                       HYS11110
      DA(2,1)=DYA                                                       HYS11120
      DA(3,1)=DZA                                                       HYS11130
      DA(4,1)=DROXA                                                     HYS11140
      DA(5,1)=DROYA                                                     HYS11150
      DA(6,1)=DROZA                                                     HYS11160
      DB(1,1)=DXB                                                       HYS11170
      DB(2,1)=DYB                                                       HYS11180
      DB(3,1)=DZB                                                       HYS11190
      DB(4,1)=DROXB                                                     HYS11200
      DB(5,1)=DROYB                                                     HYS11210
      DB(6,1)=DROZB                                                     HYS11220
C                                                                       HYS11230
      DO 710 II=1,12                                                    HYS11240
      DO 710 JJ=1,12                                                    HYS11250
        DDA(II,JJ)=0                                                    HYS11260
      DO 710 KK=1,12                                                    HYS11270
  710   DDA(II,JJ)=DDA(II,JJ)+R(II,KK)*DA(KK,JJ)                        HYS11280
      DO 720 II=1,12                                                    HYS11290
      DO 720 JJ=1,12                                                    HYS11300
        DDB(II,JJ)=0                                                    HYS11310
      DO 720 KK=1,12                                                    HYS11320
  720   DDB(II,JJ)=DDB(II,JJ)+R(II,KK)*DB(KK,JJ)                        HYS11330
C                                                                       HYS11340
C  ============================================                         HYS11350
C  == FIND SEGMENT BASIC DEFORMATION INCREMENTS ==                      HYS11360
C  ============================================                         HYS11370
C     DW=DDB(3,1) - DDA(3,1)                                            HYS11380
C     DV=DDB(2,1) - DDA(2,1)                                            HYS11390
C     DU=DDB(1,1) - DDA(1,1)                                            HYS11400
C     TTL=SQRT((EL(ISEG)+DW)**2 + DV**2 + DU**2 )                       HYS11410
C     DEC=(TTL - EL(ISEG) )/EL(ISEG)                                    HYS11420
C     DDA(4,1)=DDA(4,1) + ( DV/EL(ISEG))                                HYS11430
C     DDB(4,1)=DDB(4,1) + ( DV/EL(ISEG))                                HYS11440
C     DDA(5,1)=DDA(5,1) + ( DU/EL(ISEG))                                HYS11450
C     DDB(5,1)=DDB(5,1) + ( DU/EL(ISEG))                                HYS11460
C     DCU=(DDB(4,1) - DDA(4,1))/EL(ISEG)                                HYS11470
C     DCV=(DDB(5,1) - DDA(5,1))/EL(ISEG)                                HYS11480
```

```
C                                                                       HTS11490
      DEC=(DDB(3,1)-DDA(3,1))/EL(ISEG)                                  HTS11500
      DCU=(DDB(4,1)-DDA(4,1))/EL(ISEG)                                  HTS11510
      DCV=(DDB(5,1)-DDA(5,1))/EL(ISEG)                                  HTS11520
C                                                                       HTS11530
      RETURN                                                            HTS11540
      END                                                               HTS11550
C                                                                       HTS11560
C                                                                       HTS11570
C.pa                                                                    HTS11580
C     DEBUG UNIT(6),SUBCHK                                              HTS11590
C     END DEBUG                                                         HTS11600
      SUBROUTINE PRINC(DEC,DCU,DCV,ISEG,PRINT,ST,TOTA,LIBN,             HTS11610
     &                 EM,ET,G,TS,IOPT,NUMP,NSEG,                       HTS11620
     &                 EA,EIU,EIV,GKT,                                  HTS11630
     &                 IREV1,IREV2,IREV3,IREV4,UC0,VC0,IELNO,IMATER,    HTS11640
     &                 A0,B,BETA,E0,E1,E2,U,V,UC,VC,TE,E,DE,            HTS11650
     &                 INEB,INEH,FJL,                                   HTS11660
     &                 RATIO3,IR,STR,DEP,EOP,STRP,TEP,EREF,SREF,FLP)    HTS11670
C ******************************************                            HTS11680
C SUBROUTINE PRINC FOR CALCULATE PRINCIPAL AXES                         HTS11690
C ******************************************                            HTS11700
C                                                                       HTS11710
C ECCX = X- DISTANCE FROM ECCENTRIC LOAD TO INSTANCE SHAPE-CENTER       HTS11720
C ECCY = Y- DISTANCE FROM ECCENTRIC LOAD TO INSTANCE SHAPE-CENTER       HTS11730
C ECX  = X- DISTANCE FROM ECCENTRIC LOAD TO ORIGINAL SHAPE-CENTER       HTS11740
C ECY  = Y- DISTANCE FROM ECCENTRIC LOAD TO ORIGINAL SHAPE-CENTER       HTS11750
C ***************************************************************      HTS11760
      DIMENSION A0(NUMP),B(NUMP),BETA(NSEG),DE(NSEG,NUMP)               HTS11770
      DIMENSION E(NUMP) ,E0(NSEG,NUMP),E1(NSEG,NUMP),E2(NSEG,NUMP)      HTS11780
      DIMENSION TE(NUMP) ,U(NSEG,NUMP),V(NSEG,NUMP)                     HTS11790
      DIMENSION UC(NUMP),VC(NUMP),STR(NSEG,NUMP),FJL(NUMP)              HTS11800
      DIMENSION DEP(NSEG,NUMP),EOP(NSEG,NUMP),                          HTS11810
     &          STRP(NSEG,NUMP),TEP(NSEG,NUMP),                        HTS11820
     &          EREF(NSEG,NUMP),SREF(NSEG,NUMP)                         HTS11830
      LOGICAL PRINT                                                     HTS11840
      REAL INEB, INEH                                                   HTS11850
      IELAS=1                                                           HTS11860
C                                                                       HTS11870
C CALCULATE STRAIN RATE OF EACH SECTION ELEMENT                         HTS11880
C                                                                       HTS11890
      DO 50 I=1,NUMP                                                    HTS11900
                                                                        HTS11910
      DE(ISEG,I)=DEC*V(ISEG,I)*DCU-U(ISEG,I)*DCV                        HTS11920
                                                                        HTS11930
      E(I)=E0(ISEG,I)+DE(ISEG,I)                                        HTS11940
      E0(ISEG,I)=E(I)                                                   HTS11950
   50 CONTINUE                                                          HTS11960
      IF ( IMATER .EQ. 0 )                                              HTS11970
     &     CALL JUDEL(IELAS,NSEG,NUMP,ISEG,PRINT,IELNO,                 HTS11980
     &                EM,ET,RATIO3,TS,                                  HTS11990
     &                DE,E,E0,E1,E2,TE,STR )                            HTS12000
      IF ( IMATER .EQ. 1 )                                              HTS12010
     &     CALL JUDEL2(IELAS,NSEG,NUMP,ISEG,PRINT,IELNO,                HTS12020
     &                 EM,ET,RATIO3,TS,                                 HTS12030
     &                 DE,E,E0,E1,E2,TE,STR )                           HTS12040
      IF ( IMATER .EQ. 2 )                                              HTS12050
     &     CALL JUDEL3(IELAS,NSEG,NUMP,ISEG,PRINT,IELNO,                HTS12060
     &                 EM,ET,IR,TS,                                     HTS12070
     &                 DE,E,E0,E1,E2,TE,                                HTS12080
     &                 DEP,EOP,STRP,STR,TEP,EREF,SREF,IOPT)             HTS12090
      IF ( IMATER .EQ. 3 )                                              HTS12100
     &     CALL JUDEL4(IELAS,NSEG,NUMP,ISEG,PRINT,IELNO,                HTS12110
     &                 EM,ET,RATIO3,TS,                                 HTS12120
     &                 DE,E,E0,E1,E2,TE,STR )                           HTS12130
C                                                                       HTS12140
C CHECK SECTION REACH TO PLASTIC HINGE OR NOT                           HTS12150
C                                                                       HTS12160
      IPL=1                                                             HTS12170
      DO 566 I=1,NUMP                                                   HTS12180
        IF( TE(I) .EQ. EM) THEN                                         HTS12190
          IPL=0                                                         HTS12200
          GO TO 996                                                     HTS12210
        ENDIF                                                           HTS12220
  566 CONTINUE                                                          HTS12230
  996 IF ( PRINT) THEN                                                  HTS12240
        IF(PRINT .AND. IPL .EQ. 1 ) THEN                                HTS12250
          WRITE(6,567)IELNO,ISEG                                        HTS12260
  567     FORMAT(1X,'*** THE STABILITY MEMBER ',I5,                     HTS12270
     $           ', SEGMENT ',I5,' REACH TO PLASTIC HINGE ***')         HTS12280
        ENDIF                                                           HTS12290
C                                                                       HTS12300
C     JUDGE LOCAL BUCKLING FOR BOX SECTION                              HTS12310
C                                                                       HTS12320
      IF( LIBN .EQ. 1 .AND. IREV1 .NE. 0 ) THEN                         HTS12330
         CALL BOXLOC(LIBN,FJL,STR,NSEG,NUMP,ISEG,INEB,INEH,             HTS12340
     &               TE,ST,EM,UC,VC,FLP )                               HTS12350
      ENDIF                                                             HTS12360
C                                                                       HTS12370
C CALCULATE THE SECTIONAL PROPERTIES ABOUT SECTIONAL REFERENCE          HTS12380
C AXES WITH ORIGIN AT ( 0 , 0 )                                         HTS12390
C                                                                       HTS12400
      EA0=0.                                                            HTS12410
      ESV0=0.                                                           HTS12420
      ESU0=0.                                                           HTS12430
      EIU0=0.                                                           HTS12440
      EIV0=0.                                                           HTS12450
      EIUV0=0.                                                          HTS12460
      DO 60 I=1,NUMP                                                    HTS12470
         IF( FJL(I) .EQ. 0 ) THEN                                       HTS12480
            EA0=EA0+TE(I)*A0(I)                                         HTS12490
            ESV0=ESV0+TE(I)*V0(I)*A0(I)                                 HTS12500
            ESU0=ESU0+TE(I)*U0(I)*A0(I)                                 HTS12510
            EIU0=EIU0+TE(I)*V0(I)*V0(I)*A0(I)                           HTS12520
            EIV0=EIV0+TE(I)*U0(I)*U0(I)*A0(I)                           HTS12530
            EIUV0=EIUV0+TE(I)*U0(I)*V0(I)*A0(I)                         HTS12540
         ELSE IF( FJL(I) .EQ. 1 ) THEN                                  HTS12550
            EA0=EA0+RATIO3*EM*A0(I)                                     HTS12560
            ESV0=ESV0+RATIO3*EM*V0(I)*A0(I)                             HTS12570
            ESU0=ESU0+RATIO3*EM*U0(I)*A0(I)                             HTS12580
            EIU0=EIU0+RATIO3*EM*V0(I)*V0(I)*A0(I)                       HTS12590
            EIV0=EIV0+RATIO3*EM*U0(I)*U0(I)*A0(I)                       HTS12600
            EIUV0=EIUV0+RATIO3*EM*U0(I)*V0(I)*A0(I)                     HTS12610
         ENDIF                                                          HTS12620
   60 CONTINUE                                                          HTS12630
      IF(PRINT) THEN                                                    HTS12640
        WRITE(6,8856)IELNO,ISEG                                         HTS12650
 8856   FORMAT('1','THE STABILITY MEMBER ',I3,' SECTION ',I3/)          HTS12660
      WRITE(6,901)                                                      HTS12670
  901 FORMAT(1X,'EA0,ESV0,ESU0,EIU0,EIV0,EIUV0;')                       HTS12680
        WRITE(6,902)EA0,ESV0,ESU0,EIU0,EIV0,EIUV0                       HTS12690
  902   FORMAT(1X,1P,6G15.5/)                                           HTS12700
      ENDIF                                                             HTS12710
C                                                                       HTS12720
C THE NEW CENTROID LOCATION CORRESPONDING TO SECTION REFERENCE AXES     HTS12730
C                                                                       HTS12740
      UC0=ESU0/EA0                                                      HTS12750
      VC0=ESV0/EA0                                                      HTS12760
C                                                                       HTS12770
C THE SECTIONAL PROPERTIES ABOUT SECTIONAL REFERENCE AXES WITH           HTS12780
C ORIGIN AT ( UC0,VC0 )                                                 HTS12790
C                                                                       HTS12800
      EIU1=EIU0-VC0*VC0*EA0                                             HTS12810
      EIV1=EIV0-UC0*UC0*EA0                                             HTS12820
      EIUV1=EIUV0-UC0*VC0*EA0                                           HTS12830
                                                                        HTS12840
      IF( EIU1 .LE. 0. .OR. EIV1 .LE. 0. ) THEN                         HTS12850
         WRITE(6,*)' ** INEFFICIENT SECTION ELEMENT ARRENGEMENT **'     HTS12860
         WRITE(6,*)'    ====ERROR======ERROR========ERROR====    '      HTS12870
         WRITE(6,*)' **          EIU1 OR EIV1 IS NEGATIVE         **'  HTS12880
      ENDIF                                                             HTS12890
                                                                        HTS12900
      IF(PRINT) THEN                                                    HTS12910
      WRITE(6,109)                                                      HTS12920
  109 FORMAT(1X,'THE UC0 ,VC0,EIU1,EIV1,EIUV1;'/)                       HTS12930
      WRITE(6,867)UC0,VC0,EIU1,EIV1,EIUV1                               HTS12940
  867 FORMAT(1P,5G15.6/)                                                HTS12950
      ENDIF                                                             HTS12960
C                                                                       HTS12970
            IF( ABS((EIV1-EIU1)/EIV1) .LE. 0.00001 .AND.                HTS12980
     &          ABS(EIUV1/EM) .LE. 0.00001 ) THEN                       HTS12990
                BETA(ISEG)=0                                            HTS13000
            ELSE IF( ABS((EIV1-EIU1)/EIV1) .LE. 0.00001 .AND.           HTS13010
     &          ABS(EIUV1/EM) .GT. 0.00001 ) THEN                       HTS13020
                BETA(ISEG)=0.5*ATAN2(2.*EIUV1,0.)                       HTS13030
            ELSE IF( ABS((EIV1-EIU1)/EIV1) .GT. 0.00001 .AND.           HTS13040
     &          ABS(EIUV1/EM) .LE. 0.00001 ) THEN                       HTS13050
                BETA(ISEG)=0                                            HTS13060
            ELSE IF( ABS((EIV1-EIU1)/EIV1) .GT. 0.00001 .AND.           HTS13070
     &          ABS(EIUV1/EM) .GT. 0.00001 ) THEN                       HTS13080
              BETA(ISEG)=0.5*ATAN2(2.*EIUV1,(EIV1-EIU1))                HTS13090
            ENDIF                                                       HTS13100
C                                                                       HTS13110
C           ------------------------------------------------            HTS13120
C           === LIMIT BETA TO (90 DEGREE TO -90 DEGREE) ===             HTS13130
C           ------------------------------------------------            HTS13140
            P180=3.1415926                                              HTS13150
            P90=3.1415926/2.                                            HTS13160
            IF( BETA(ISEG) .GT. P90                                     HTS13170
     &          .AND. BETA(ISEG) .LE. P180  ) THEN                      HTS13180
                BETA(ISEG)=BETA(ISEG)-P180                              HTS13190
            ELSE IF( BETA(ISEG) .GE. -P180                              HTS13200
     &          .AND. BETA(ISEG) .LT. -P90  ) THEN                      HTS13210
                BETA(ISEG)=BETA(ISEG)+P180                              HTS13220
            ENDIF                                                       HTS13230
C           ------------------------------------------------            HTS13240
C           === AVOID UNSTABLE BETA ANGLE OCCURRED ===                  HTS13250
C           === NEAR 90, -90, AND 0 DEGREE         ===                  HTS13260
C           ------------------------------------------------            HTS13270
            IF( ABS(ABS(BETA(ISEG))-P90) .LT. 0.03 ) THEN               HTS13280
                IF(BETA(ISEG) .LT. 0 ) BETA(ISEG)=-P90                  HTS13290
                IF(BETA(ISEG) .GT. 0 ) BETA(ISEG)= P90                  HTS13300
            ELSE IF( ABS(BETA(ISEG)) .LT. 0.03 ) THEN                   HTS13310
                BETA(ISEG)=0                                            HTS13320
            ENDIF                                                       HTS13330
C                                                                       HTS13340
            IF( BETA(ISEG) .EQ. 0 ) THEN                                HTS13350
               EA=EA0                                                   HTS13360
               EIU=EIU1                                                 HTS13370
               EIV=EIV1                                                 HTS13380
            ELSE IF( BETA(ISEG) .NE. 0 ) THEN                           HTS13390
               EA=EA0                                                   HTS13400
               EIU=0.5*(EIU1+EIV1)-(EIUV1/SIN(2.*BETA(ISEG)))           HTS13410
               EIV=0.5*(EIU1+EIV1)+(EIUV1/SIN(2.*BETA(ISEG)))           HTS13420
            ENDIF                                                       HTS13430
                                                                        HTS13440
      IF(PRINT) THEN                                                    HTS13450
      WRITE(6,309)                                                      HTS13460
  309 FORMAT(1X,'THE BETA,EA,EIU,EIV       ;'/)                       HTS13470
      WRITE(6,867)BETA(ISEG),EA,EIU,EIV                                 HTS13480
      ENDIF                                                             HTS13490
C                                                                       HTS13500
C SECTION RIGIDITIES WITH RESPECT TO THE INSTANTANEOUS PRINCIPAL        HTS13510
C AXES U AND V                                                          HTS13520
C                                                                       HTS13530
C                                                                       HTS13540
      CALL SECRIG(LIBN,ST,TOTA,BJ,IREV1,IREV2,IREV3,IREV4,IELNO,        HTS13550
     &            B,NUMP)                                               HTS13560
      GKT=G*BJ                                                          HTS13570
C                                                                       HTS13580
C CALCULATE PRESENT SECTIONAL ELEMENT COORDINATE                        HTS13590
C ACCORDING TO INSTANTANEOUS AXES                                       HTS13600
C                                                                       HTS13610
      DO 52 J=1,NUMP                                                    HTS13620
         U(ISEG,J)=COS(BETA(ISEG))*                                     HTS13630
     *             (U0(J)-UC0)+SIN(BETA(ISEG))*                       HTS13640
     *             (V0(J)-VC0)                                        HTS13650
         V(ISEG,J)=-SIN(BETA(ISEG))*                                    HTS13660
     *             (U0(J)-UC0)+COS(BETA(ISEG))                        HTS13670
     $             *(V0(J)-VC0)                                       HTS13680
   52 CONTINUE                                                          HTS13690
      IF(PRINT) THEN                                                    HTS13700
      WRITE(6,7776)                                                     HTS13710
 7776 FORMAT(1X,'THE U AND V IN PRINC SUBROUTINE'/)                     HTS13720
      WRITE(6,7778)(U(ISEG,J),V(ISEG,J),J=1,NUMP)                       HTS13730
 7778 FORMAT(1X,1P,2G15.6)                                              HTS13740
      ENDIF                                                             HTS13750
      RETURN                                                            HTS13760
      END                                                               HTS13770
```

```
C
C
C.pa
C     DEBUG UNIT(6),SUBCHK
C     END DEBUG
C
      SUBROUTINE SECRIG(LIBN,ST,TOTA,BJ,L1,N1,L2,N2,IELNO,
     &                  B,NUMP)
C     ***************************************************************
C     FIND SECTION RIGIDITY
C
C     NOTE:
C         THE TORSIONAL RIGITY IS ASSUMED ELASTIC CURRENTLY
C     ***************************************************************
      DIMENSION B(NUMP)
C
      BJJ=0.
      BJ=0.
      GO TO (100,100,150,200,200,300,400,400), LIBN
C
  100 CONTINUE
C
C FOR SECT1,2 ELEMENT
C
      DO 48 I=1,NUMP
         BJJ=BJJ+B(I)
   48 CONTINUE
      BJ=4.*TOTA*TOTA*ST/BJJ
      RETURN
C
  150 CONTINUE
C
C FOR SECTION 3 ELEMENT
C
      TEMP1=1-(B(2)**4/(12*B(1)**4))
      BJ=(B(1)*B(2)**3)*(5.3333-(3.36*B(2)/(B(1)))*TEMP1)
      RETURN
C
  200 CONTINUE
C
C FOR SECT 4 ELEMENT
C
         BJ=B(1)
      RETURN
C
  300 CONTINUE
C
C FOR SECT6 ELEMENT
C
      NNUMP=NUMP/2
      DO 58 I=1,NNUMP
         BJJ=BJJ+((B(I)+B(I+NNUMP))/2.)
   58 CONTINUE
         BJ=4.*TOTA*TOTA*ST/BJJ
      RETURN
C
  400 CONTINUE
C
C FOR SECT7 AND SECT8 ELEMENT
C
      ST1=ST/N1
      DO 50 I=1,L1
         SST1=0.
      DO 60 J=1,N1
        K=I+(J-1)*L1
         SST1=SST1+B(I)
   60 CONTINUE
      BJ=BJ+((B(I)*SST1**3)/3.)
   50 CONTINUE
      ST2=ST/N2
      DO 70 I=(L1*N1)+1,(L1*N1)+L2
         SST2=0.
      DO 80 J=1,N2
        K=I+(J-1)*L2
         SST2=SST2+SST2
   80 CONTINUE
      BJ=BJ+((B(I)*SST2**3)/3.)
   70 CONTINUE
      RETURN
C
      END
C
C.pa
C     DEBUG UNIT(6),SUBCHK
C     END DEBUG
      SUBROUTINE JUDEL(IELAS,NSEG,NUMP,ISEG,PRINT,IELNO,
     &                 EM,ET,RATIO3,YS,
     &                 DE,E,E0,E1,E2,TE,STR )
C     ***********************************************
C     ELASTO-PLASTIC STRESS-STRAIN HISTERESIS MODEL
C     ***********************************************
      DIMENSION DE(NSEG,NUMP),E(NUMP),E0(NSEG,NUMP),E1(NSEG,NUMP)
      DIMENSION E2(NSEG,NUMP),TE(NUMP),STR(NSEG,NUMP)
      LOGICAL PRINT
C
C     ----------------------------------
C     --       ELASTIC CONDITION      --
C     ----------------------------------
C
C.....ELASTIC CASE SOLUTION
        IF( IELAS .EQ. 0 ) THEN
          DO 79 I=1,NUMP
            TE(I)=EM
            STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
   79     CONTINUE
          RETURN
        ENDIF
C
      DO 75 I=1,NUMP
      IF( E(I) .LT. E1(ISEG,I) .AND.
     $    E(I) .GT. E2(ISEG,I)) THEN
          TE(I)=EM
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
      ENDIF
C
C     ----------------------------------
C     --    TENSION STRAIN CONTROL    --
C     ----------------------------------
      IF(E(I).GE.E1(ISEG,I) .AND. DE(ISEG,I) .GE. 0.) THEN
          E1(ISEG,I)=E(I)
          E2(ISEG,I)=E(I)-2.*ET
          TE(I)=RATIO3*EM
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
      ENDIF
      IF(E(I).GE.E1(ISEG,I) .AND. DE(ISEG,I) .LT. 0.) THEN
C         E1(ISEG,I)=E(I)
C         E2(ISEG,I)=E(I)-2.*ET
          TE(I)=EM
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
      ENDIF
C
C     ----------------------------------
C     -- COMPRESSION STRAIN CONTROL --
C     ----------------------------------
      IF(E(I).LE.E2(ISEG,I) .AND. DE(ISEG,I) .LE. 0.) THEN
          E2(ISEG,I)=E(I)
          E1(ISEG,I)=E(I)+2.*ET
          TE(I)=RATIO3*EM
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
      ENDIF
      IF(E(I).LE.E2(ISEG,I) .AND. DE(ISEG,I) .GT. 0.) THEN
C         E2(ISEG,I)=E(I)
          E1(ISEG,I)=E(I)+2.*ET
          TE(I)=EM
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
      ENDIF
   75 CONTINUE
C
C OUTPUT SECTIONAL ELEMENT STRAIN RATE, STRAIN,ETC.
C
      IF(PRINT) THEN
      WRITE(6,32)ISEG
   32 FORMAT('1','TENGENT STIFFNESS FOR MEMBER ',I3,'SEGMENT ',I5,':')
      WRITE(6,34)(TE(I),I=1,NUMP)
   34 FORMAT(1X,5G15.6)
      WRITE(6,321)
  321 FORMAT(/4X,'SEG.          E1',
     *          '              E2              DE              E0')
      WRITE(6,44)(I,E1(ISEG,I),E2(ISEG,I),DE(ISEG,I),
     *          E0(ISEG,I),I=1,NUMP)
   44 FORMAT(1X,I5,4G15.6)
      ENDIF
      RETURN
      END
C
C
C.pa
C     DEBUG UNIT(6),SUBCHK
C     END DEBUG
      SUBROUTINE JUDEL2(IELAS,NSEG,NUMP,ISEG,PRINT,IELNO,
     &                  EM,ET,RATIO3,YS,
     &                  DE,E,E0,E1,E2,TE,STR )
C     ***********************************************
C     MODIFIED BILINEAR STESS-STRAIN HYSTERESIS MODEL
C     ***********************************************
      DIMENSION DE(NSEG,NUMP),E(NUMP),E0(NSEG,NUMP),E1(NSEG,NUMP)
      DIMENSION E2(NSEG,NUMP),TE(NUMP),STR(NSEG,NUMP)
      LOGICAL PRINT
C
C     ----------------------------------
C     --       ELASTIC CONDITION      --
C     ----------------------------------
C
C.....ELASTIC CASE SOLUTION
        IF( IELAS .EQ. 0 ) THEN
          DO 79 I=1,NUMP
            TE(I)=EM
   79     CONTINUE
          RETURN
        ENDIF
C
      DO 75 I=1,NUMP
      IF( E(I) .LT. E1(ISEG,I) .AND.
     $    E(I) .GT. E2(ISEG,I)) THEN
          TE(I)=EM
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
      ENDIF
C
C     ----------------------------------
C     --    TENSION STRAIN CONTROL    --
C     ----------------------------------
      IF(E(I).GE.E1(ISEG,I) .AND. DE(ISEG,I) .GE. 0.) THEN
        IF(ABS(STR(ISEG,I)) .GE. YS) THEN
          E1(ISEG,I)=E(I)
          E2(ISEG,I)=E(I)-2.*ET
          TE(I)=RATIO3*EM
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
        ELSE IF(ABS(STR(ISEG,I)) .LT. YS) THEN
          TE(I)=EM
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
        ENDIF
      ELSE IF(E(I).GE.E1(ISEG,I) .AND. DE(ISEG,I) .LT. 0.) THEN
C         E1(ISEG,I)=E(I)
C         E2(ISEG,I)=E(I)-2.*ET
          TE(I)=EM
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
C
C     ----------------------------------
C     -- COMPRESSION STRAIN CONTROL --
C     ----------------------------------
      ELSE IF(E(I).LE.E2(ISEG,I) .AND. DE(ISEG,I) .LE. 0.) THEN
        IF(ABS(STR(ISEG,I)) .GE. YS) THEN
          E2(ISEG,I)=E(I)
          E1(ISEG,I)=E(I)+2.*ET
          TE(I)=RATIO3*EM
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
        ELSE IF(ABS(STR(ISEG,I)) .LT. YS) THEN
          TE(I)=EM
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
        ENDIF
      ELSE IF(E(I).LE.E2(ISEG,I) .AND. DE(ISEG,I) .GT. 0.) THEN
C         E2(ISEG,I)=E(I)
C         E1(ISEG,I)=E(I)+2.*ET
          TE(I)=EM
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)
      ENDIF
   75 CONTINUE
C
C OUTPUT SECTIONAL ELEMENT STRAIN RATE, STRAIN,ETC.
C
      IF(PRINT) THEN
      WRITE(6,32)ISEG
   32 FORMAT('1','TENGENT STIFFNESS FOR MEMBER ',I3,'SEGMENT ',I5,':')
      WRITE(6,34)(TE(I),I=1,NUMP)
   34 FORMAT(1X,5G15.6)
      WRITE(6,321)
  321 FORMAT(/4X,'SEG.          E1',
     *          '              E2              DE              E0')
      WRITE(6,44)(I,E1(ISEG,I),E2(ISEG,I),DE(ISEG,I),
     *          E0(ISEG,I),I=1,NUMP)
   44 FORMAT(1X,I5,4G15.6)
      ENDIF
      RETURN
      END
C
C.pa
C$PROCESS SDUMP GOSTMT XREF
C     DEBUG UNIT(6),SUBCHK
C     END DEBUG
      SUBROUTINE JUDEL3(IELAS,NSEG,NUMP,ISEG,PRINT,IELNO,
     &                  EM,ET,IR,YS,
     &                  DE,E,E0,E1,E2,TE,
     &                  DEP,EOP,STRP,STR,TEP,EREF,SREF,MOPT)
C     ***********************************************
C     RAMBERG-OSGOOD STRESS-STRAIN HYSTERESIS MODEL
C     ***********************************************
      DIMENSION DE(NSEG,NUMP),E(NUMP),E0(NSEG,NUMP),E1(NSEG,NUMP)
      DIMENSION E2(NSEG,NUMP),TE(NUMP),STR(NSEG,NUMP)
      DIMENSION DEP(NSEG,NUMP),EOP(NSEG,NUMP),
     &          STRP(NSEG,NUMP),TEP(NSEG,NUMP),
     &          EREF(NSEG,NUMP),SREF(NSEG,NUMP)
      LOGICAL PRINT
C
C     ----------------------------------
C     --       ELASTIC CONDITION      --
C     ----------------------------------
C
C.....ELASTIC CASE SOLUTION
        IF( IELAS .EQ. 0 ) THEN
          DO 79 I=1,NUMP
            TE(I)=EM
   79     CONTINUE
```

```
        RETURN                                                          HYS16750
      ENDIF                                                             HYS16760
C                                                                       HYS16770
C                                                                       HYS16780
C     ****************************                                      HYS16790
C     ** FOR INITIAL CONDITION **                                       HYS16800
C     ****************************                                      HYS16810
C                                                                       HYS16820
C                                                                       HYS16830
      DO 75 I=1,NUMP                                                    HYS16840
      IF( MOPT .EQ. 2 ) THEN                                            HYS16850
          TE(I)=EM                                                      HYS16860
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      HYS16870
          TEP(ISEG,I)=TE(I)                                             HYS16880
          DEP(ISEG,I)=DE(ISEG,I)                                        HYS16890
          EOP(ISEG,I)=E(I)                                              HYS16900
          STRP(ISEG,I)=STR(ISEG,I)                                      HYS16910
C                                                                       HYS16920
C     ****************************                                      HYS16930
C     ** CHECK REVERSE POINT **                                         HYS16940
C     ****************************                                      HYS16950
      ELSE IF(DEP(ISEG,I)* DE(ISEG,I) .LT. 0 ) THEN                     HYS16960
          EREF(ISEG,I)=EOP(ISEG,I)                                      HYS16970
          SREF(ISEG,I)=STRP(ISEG,I)                                     HYS16980
          TE(I)=EM                                                      HYS16990
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      HYS17000
          TEP(ISEG,I)=TE(I)                                             HYS17010
          DEP(ISEG,I)=DE(ISEG,I)                                        HYS17020
          EOP(ISEG,I)=E(I)                                              HYS17030
          STRP(ISEG,I)=STR(ISEG,I)                                      HYS17040
C                                                                       HYS17050
C     ************************************                              HYS17060
C     ** FOR SKELETON AND BRANCH REGION **                             HYS17070
C     ************************************                              HYS17080
      ELSE IF(DEP(ISEG,I)* DE(ISEG,I) .GE. 0 ) THEN                     HYS17090
          STR(ISEG,I)=STR(ISEG,I)+TEP(ISEG,I)*DE(ISEG,I)                HYS17100
C                                                                       HYS17110
C..... FOR FIRST SKELETON CURVE                                         HYS17120
          IF( SREF(ISEG,I) .EQ. 0 .AND.                                 HYS17130
     &        SREF(ISEG,I) .EQ. 0 ) THEN                                HYS17140
              IF( ABS(STR(ISEG,I)/YS) .LT. 0.2 ) THEN                   HYS17150
                TEMP1=1/EM                                              HYS17160
C               WRITE(13,*)'CASE A'                                     HYS17170
              ELSE IF( STR(ISEG,I) .GE. 0 ) THEN                        HYS17180
C               WRITE(13,*)'CASE A'                                     HYS17190
                TEMP1=(1 + IR*ABS(STR(ISEG,I)/YS)**(IR-1) )/EM          HYS17200
              ELSE IF( STR(ISEG,I) .LT. 0 ) THEN                        HYS17210
C               WRITE(13,*)'CASE B'                                     HYS17220
                TEMP1=(1 + IR*ABS(STR(ISEG,I)/YS)**(IR-1) )/EM          HYS17230
C               TEMP1=(1 + (2-IR)*ABS(STR(ISEG,I)/YS)**(IR-1) )/EM      HYS17240
              ENDIF                                                     HYS17250
              TE(I)=1/TEMP1                                             HYS17260
              TEP(ISEG,I)=TE(I)                                         HYS17270
              DEP(ISEG,I)=DE(ISEG,I)                                    HYS17280
              EOP(ISEG,I)=E(I)                                          HYS17290
              STRP(ISEG,I)=STR(ISEG,I)                                  HYS17300
C                                                                       HYS17310
C..... AFTER FIRST SKELETON CURVE                                       HYS17320
          ELSE IF( SREF(ISEG,I) .NE. 0 .OR.                             HYS17330
     &             SREF(ISEG,I) .NE. 0 ) THEN                           HYS17340
              TEMP2= STR(ISEG,I) - SREF(ISEG,I)                         HYS17350
              IF( ABS(TEMP2/(2*YS)) .LT. 0.2 ) THEN                     HYS17360
C               WRITE(13,*)'CASE B'                                     HYS17370
                TEMP1=1/EM                                              HYS17380
              ELSE IF( TEMP2 .GE. 0 ) THEN                              HYS17390
                TEMP1=(1 + IR*ABS(TEMP2/(2*YS))**(IR-1) )/EM            HYS17400
              ELSE IF( TEMP2 .LT. 0 ) THEN                              HYS17410
                TEMP1=(1 + IR*ABS(TEMP2/(2*YS))**(IR-1) )/EM            HYS17420
C               TEMP1=(1 + (2-IR)*ABS(TEMP2/(2*YS))**(IR-1) )/EM        HYS17430
              ENDIF                                                     HYS17440
              TE(I)=1/TEMP1                                             HYS17450
              TEP(ISEG,I)=TE(I)                                         HYS17460
              DEP(ISEG,I)=DE(ISEG,I)                                    HYS17470
              EOP(ISEG,I)=E(I)                                          HYS17480
              STRP(ISEG,I)=STR(ISEG,I)                                  HYS17490
          ENDIF                                                         HYS17500
      ENDIF                                                             HYS17510
C                                                                       HYS17520
   75 CONTINUE                                                          HYS17530
C                                                                       HYS17540
C OUTPUT SECTIONAL ELEMENT STRAIN RATE, STRAIN,ETC.                     HYS17550
C                                                                       HYS17560
      IF(PRINT) THEN                                                    HYS17570
      WRITE(6,32)IELNO,ISEG                                             HYS17580
   32 FORMAT('1','TENGENT STIFFNESS FOR MEMBER ',I3,'SEGMENT ',I5,'I')  HYS17590
      WRITE(6,34)(TE(I),I=1,NUMP)                                       HYS17600
   34 FORMAT(1X,5G15.6)                                                 HYS17610
      WRITE(6,321)                                                      HYS17620
  321 FORMAT(/4X,'SEG.      STRESS',                                    HYS17630
     *       '          STRAIN          STIFFNESS ')                  HYS17640
      WRITE(6,44)(I,STR(ISEG,I),                                        HYS17650
     *           EO(ISEG,I),TE(I),I=1,NUMP)                             HYS17660
   44 FORMAT(1X,I5,3X,G15.6,3X,2G15.6)                                  HYS17670
      ENDIF                                                             HYS17680
C                                                                       HYS17690
      RETURN                                                            HYS17700
      END                                                               HYS17710
C                                                                       HYS17720
C.pa                                                                    HYS17730
C     DEBUG UNIT(6),SUBCHK                                              HYS17740
C     END DEBUG                                                         HYS17750
      SUBROUTINE JUDEL4(IELAS,NSEG,NUMP,ISEG,PRINT,IELNO,               HYS17760
     &                  EM,EY,RATIO3,YS,                                HYS17770
     &                  DE,E,EO,E1,E2,TE,STR )                          HYS17780
C     ***************************************                           HYS17790
C     THIS IS THE ELASTIC STESS-STRAIN MODEL                            HYS17800
C     ***************************************                           HYS17810
      DIMENSION DE(NSEG,NUMP),E(NUMP),EO(NSEG,NUMP),E1(NSEG,NUMP)       HYS17820
      DIMENSION E2(NSEG,NUMP),TE(NUMP),STR(NSEG,NUMP)                   HYS17830
      LOGICAL PRINT                                                     HYS17840
C                                                                       HYS17850
C.....ELASTIC CASE SOLUTION                                             HYS17860
            DO 79 I=1,NUMP                                              HYS17870
              TE(I)=EM                                                  HYS17880
              STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                  HYS17890
   79       CONTINUE                                                    HYS17900
C                                                                       HYS17910
C OUTPUT SECTIONAL ELEMENT STRAIN RATE, STRAIN,ETC.                     HYS17920
C                                                                       HYS17930
      IF(PRINT) THEN                                                    HYS17940
      WRITE(6,32)ISEG                                                   HYS17950
   32 FORMAT('1','TENGENT STIFFNESS FOR MEMBER ',I3,'SEGMENT ',I5,'I')  HYS17960
      WRITE(6,34)(TE(I),I=1,NUMP)                                       HYS17970
   34 FORMAT(1X,5G15.6)                                                 HYS17980
      WRITE(6,321)                                                      HYS17990
  321 FORMAT(/4X,'SEG.          E1',                                    HYS18000
     *       '          E2              DE              EO')            HYS18010
      WRITE(6,44)(I,E1(ISEG,I),E2(ISEG,I),DE(ISEG,I),                   HYS18020
     *           EO(ISEG,I),I=1,NUMP)                                   HYS18030
   44 FORMAT(1X,I5,4G15.6)                                              HYS18040
      ENDIF                                                             HYS18050
      RETURN                                                            HYS18060
      END                                                               HYS18070
C                                                                       HYS18080
C                                                                       HYS18090
C.pa                                                                    HYS18100
C     DEBUG UNIT(6),SUBCHK                                              HYS18110
C     END DEBUG                                                         HYS18120
      SUBROUTINE ELESTI(NSEG,ISEG,IELNO,PRINT,ISTIF,                    HYS18130
     &                  EA,EIU,EIV,GKT,EL,FLOC,PSK,SK)                  HYS18140
C     ***************************************                           HYS18150
C     CALCULATE STABILITY ELEMENT STIFFNESS                             HYS18160
C     ***************************************                           HYS18170
      DIMENSION EL(NSEG),FLOC(NSEG,12),PSK(NSEG,12,12),SK(12,12)        HYS18180
      LOGICAL PRINT                                                     HYS18190
C                                                                       HYS18200
C INITIALIZE THE SEGMENT STIFFNESS                                      HYS18210
C                                                                       HYS18220
      AUN=1200.                                                         HYS18230
      DO 430 I=1,12                                                     HYS18240
      DO 430 J=1,12                                                     HYS18250
        SK(I,J)=0.                                                      HYS18260
  430 CONTINUE                                                          HYS18270
C                                                                       HYS18280
C FORMULATE THE ELEMENT STIFFNESS MATRIX                                HYS18290
C                                                                       HYS18300
C                                                                       HYS18310
      P=FLOC(ISEG,3)                                                    HYS18320
C     WRITE(17,*)P                                                      HYS18330
      COP=P/(10.*EL(ISEG))                                              HYS18340
      IF(ISTIF .EQ. 1) THEN                                             HYS18350
          SK(1,1)=12.*EIV/(EL(ISEG)**3)-COP*12.                         HYS18360
          SK(2,2)=12.*EIU/(EL(ISEG)**3)-COP*12.                         HYS18370
          SK(3,3)=EA/EL(ISEG)                                           HYS18380
          SK(4,4)=4.*EIU/EL(ISEG)-                                      HYS18390
     &            COP*(4.*EL(ISEG)*EL(ISEG)/3.)                         HYS18400
          SK(5,5)=4.*EIV/EL(ISEG)-                                      HYS18410
     &            COP*(4.*EL(ISEG)*EL(ISEG)/3.)                         HYS18420
          SK(6,6)=GKT/EL(ISEG)                                          HYS18430
          SK(1,5)=6.*EIV/(EL(ISEG)**2)-COP*EL(ISEG)                     HYS18440
          SK(2,4)=-6.*EIU/(EL(ISEG)**2)-COP*(-EL(ISEG))                 HYS18450
          SK(7,7)=SK(1,1)                                               HYS18460
          SK(8,8)=SK(2,2)                                               HYS18470
          SK(9,9)=SK(3,3)                                               HYS18480
          SK(10,10)=SK(4,4)                                             HYS18490
          SK(11,11)=SK(5,5)                                             HYS18500
          SK(12,12)=SK(6,6)                                             HYS18510
          SK(7,11)=-SK(1,5)                                             HYS18520
          SK(8,10)=-SK(2,4)                                             HYS18530
          SK(1,7)=-SK(1,1)                                              HYS18540
          SK(2,8)=-SK(2,2)                                              HYS18550
          SK(3,9)=-SK(3,3)                                              HYS18560
          SK(4,10)=0.5*SK(4,4)-COP*(-EL(ISEG)*EL(ISEG)/3.)              HYS18570
          SK(5,11)=0.5*SK(5,5)-COP*(-EL(ISEG)*EL(ISEG)/3.)              HYS18580
          SK(6,12)=-SK(6,6)                                             HYS18590
          SK(1,11)=SK(1,5)                                              HYS18600
          SK(2,10)=SK(2,4)                                              HYS18610
          SK(4,8)=-SK(2,10)                                             HYS18620
          SK(5,7)=-SK(1,11)                                             HYS18630
      ELSE IF(ISTIF .EQ. 0) THEN                                        HYS18640
      IF( ABS(P) .LT. 1.) THEN                                          HYS18650
          SK(1,1)=12.*EIV/(EL(ISEG)**3)-COP*12.                         HYS18660
          SK(2,2)=12.*EIU/(EL(ISEG)**3)-COP*12.                         HYS18670
          SK(3,3)=EA/EL(ISEG)                                           HYS18680
          SK(4,4)=4.*EIU/EL(ISEG)-                                      HYS18690
     &            COP*(4.*EL(ISEG)*EL(ISEG)/3.)                         HYS18700
          SK(5,5)=4.*EIV/EL(ISEG)-                                      HYS18710
     &            COP*(4.*EL(ISEG)*EL(ISEG)/3.)                         HYS18720
          SK(6,6)=GKT/EL(ISEG)                                          HYS18730
          SK(1,5)=6.*EIV/(EL(ISEG)**2)-COP*EL(ISEG)                     HYS18740
          SK(2,4)=-6.*EIU/(EL(ISEG)**2)-COP*(-EL(ISEG))                 HYS18750
          SK(7,7)=SK(1,1)                                               HYS18760
          SK(8,8)=SK(2,2)                                               HYS18770
          SK(9,9)=SK(3,3)                                               HYS18780
          SK(10,10)=SK(4,4)                                             HYS18790
          SK(11,11)=SK(5,5)                                             HYS18800
          SK(12,12)=SK(6,6)                                             HYS18810
          SK(7,11)=-SK(1,5)                                             HYS18820
          SK(8,10)=-SK(2,4)                                             HYS18830
          SK(1,7)=-SK(1,1)                                              HYS18840
          SK(2,8)=-SK(2,2)                                              HYS18850
          SK(3,9)=-SK(3,3)                                              HYS18860
          SK(4,10)=0.5*SK(4,4)-COP*(-EL(ISEG)*EL(ISEG)/3.)              HYS18870
          SK(5,11)=0.5*SK(5,5)-COP*(-EL(ISEG)*EL(ISEG)/3.)              HYS18880
          SK(6,12)=-SK(6,6)                                             HYS18890
          SK(1,11)=SK(1,5)                                              HYS18900
          SK(2,10)=SK(2,4)                                              HYS18910
          SK(4,8)=-SK(2,10)                                             HYS18920
          SK(5,7)=-SK(1,11)                                             HYS18930
      ENDIF                                                             HYS18940
      IF(P .GT. 1.) THEN                                                HYS18950
          SKU=SQRT(P/EIU)                                               HYS18960
          CU=COS(SKU*EL(ISEG))*AUN                                      HYS18970
          SU=SIN(SKU*EL(ISEG))*AUN                                      HYS18980
          SKV=SQRT(P/EIV)                                               HYS18990
          CV=COS(SKV*EL(ISEG))*AUN                                      HYS19000
          SV=SIN(SKV*EL(ISEG))*AUN                                      HYS19010
          SK(1,1)=EIV*((SKV**3)*SV/(2.*AUN-2.*CV-SKV*EL(ISEG)*SV))      HYS19020
          SK(2,2)=EIU*((SKU**3)*SU/(2.*AUN-2.*CU-SKU*EL(ISEG)*SU))      HYS19030
          SK(3,3)=EA/EL(ISEG)                                           HYS19040
          SK(4,4)=EIU*SKU*((SU-SKU*EL(ISEG)*CU)/                        HYS19050
     *            (2.*AUN-2.*CU-SKU*EL(ISEG)*SU))                       HYS19060
          SK(5,5)=EIV*SKV*((SV-SKV*EL(ISEG)*CV)/                        HYS19070
     *            (2.*AUN-2.*CV-SKV*EL(ISEG)*SV))                       HYS19080
          SK(6,6)=GKT/EL(ISEG)                                          HYS19090
          SK(1,5)=EIV*((SKV**2)*(1.*AUN-CV)/                            HYS19100
     *            (2.*AUN-2.*CV-SKV*EL(ISEG)*SV))                       HYS19110
          SK(2,4)=-EIU*((SKU**2)*(1.*AUN-CU)/                           HYS19120
     *            (2.*AUN-2.*CU-SKU*EL(ISEG)*SU))                       HYS19130
          SK(7,7)=SK(1,1)                                               HYS19140
          SK(8,8)=SK(2,2)                                               HYS19150
          SK(9,9)=SK(3,3)                                               HYS19160
          SK(10,10)=SK(4,4)                                             HYS19170
          SK(11,11)=SK(5,5)                                             HYS19180
          SK(12,12)=SK(6,6)                                             HYS19190
          SK(7,11)=-SK(1,5)                                             HYS19200
          SK(8,10)=-SK(2,4)                                             HYS19210
          SK(1,7)=-SK(1,1)                                              HYS19220
          SK(2,8)=-SK(2,2)                                              HYS19230
          SK(3,9)=-SK(3,3)                                              HYS19240
          SK(4,10)=EIU*SKU*((SKU*EL(ISEG)*AUN-SU)/                      HYS19250
     *            (2.*AUN-2.*CU-SKU*EL(ISEG)*SU))                       HYS19260
          SK(5,11)=EIV*SKV*((SKV*EL(ISEG)*AUN-SV)/                      HYS19270
     *            (2.*AUN-2.*CV-SKV*EL(ISEG)*SV))                       HYS19280
          SK(6,12)=-SK(6,6)                                             HYS19290
          SK(1,11)=SK(1,5)                                              HYS19300
          SK(2,10)=SK(2,4)                                              HYS19310
          SK(4,8)=-SK(2,10)                                             HYS19320
          SK(5,7)=-SK(1,11)                                             HYS19330
      ENDIF                                                             HYS19340
      IF(P .LT. -1.) THEN                                               HYS19350
          SKU=SQRT(-P/EIU)                                              HYS19360
          CU=COSH(SKU*EL(ISEG))*AUN                                     HYS19370
          SU=SINH(SKU*EL(ISEG))*AUN                                     HYS19380
          SKV=SQRT(-P/EIV)                                              HYS19390
          CV=COSH(SKV*EL(ISEG))*AUN                                     HYS19400
          SV=SINH(SKV*EL(ISEG))*AUN                                     HYS19410
          SK(1,1)=EIV*((SKV**3)*SV/(2.*AUN-2.*CV+SKV*EL(ISEG)*SV))      HYS19420
          SK(2,2)=EIU*((SKU**3)*SU/(2.*AUN-2.*CU+SKU*EL(ISEG)*SU))      HYS19430
          SK(3,3)=EA/EL(ISEG)                                           HYS19440
          SK(4,4)=EIU*SKU*((SKU*EL(ISEG)*CU-SU)/                        HYS19450
     *            (2.*AUN-2.*CU+SKU*EL(ISEG)*SU))                       HYS19460
          SK(5,5)=EIV*SKV*((SKV*EL(ISEG)*CV-SV)/                        HYS19470
     *            (2.*AUN-2.*CV+SKV*EL(ISEG)*SV))                       HYS19480
          SK(6,6)=GKT/EL(ISEG)                                          HYS19490
          SK(1,5)=EIV*((SKV**2)*(CV-1.*AUN)/                            HYS19500
     *            (2.*AUN-2.*CV+SKV*EL(ISEG)*SV))                       HYS19510
          SK(2,4)=-EIU*((SKU**2)*(CU-1.*AUN)/                           HYS19520
     *            (2.*AUN-2.*CU+SKU*EL(ISEG)*SU))                       HYS19530
          SK(7,7)=SK(1,1)                                               HYS19540
          SK(8,8)=SK(2,2)                                               HYS19550
          SK(9,9)=SK(3,3)                                               HYS19560
          SK(10,10)=SK(4,4)                                             HYS19570
          SK(11,11)=SK(5,5)                                             HYS19580
```

```
            SK(12,12)=SK(6,6)
            SK(7,11)=-SK(1,5)
            SK(8,10)=-SK(2,4)
            SK(1,7)=-SK(1,1)
            SK(2,8)=-SK(2,2)
            SK(3,9)=-SK(3,3)
            SK(4,10)=EIU*SKU*((SU-SKU*EL(ISEG)*AUN)/
     *                 (2.*AUN-2.*CU+SKU*EL(ISEG)*SU))
            SK(5,11)=EIV*SKV*((SV-SKV*EL(ISEG)*AUN)/
     *                 (2.*AUN-2.*CV+SKV*EL(ISEG)*SV))
            SK(6,12)=-SK(6,6)
            SK(1,11)=SK(1,5)
            SK(2,10)=SK(2,4)
            SK(4,8)=-SK(2,10)
            SK(5,7)=-SK(1,11)
          ENDIF
          ENDIF
          DO 90 I=1,12
            II=I+1
            DO 90 J=II,12
              SK(J,I)=SK(I,J)
   90   CONTINUE
          IF(PRINT) THEN
              WRITE(6,432)IELNO
  432   FORMAT('1','THE ELEMENT',I3,' STIFFNESS')
          CALL PRIN(SK,12,12,12,12)
          ENDIF
C
C MEMORIZE THE ELEMENT STIFFNESS FOR BEING USED IN SUBROUTINE
C CLDAF IN THE NEXT STEP
C
          DO 190 II=1,12
          DO 190 JJ=1,12
            PSK(ISEG,II,JJ)=SK(II,JJ)
  190   CONTINUE
C
C MEMORIZE THE ELEMENT LENGTH FOR BEING USED IN THE NEXT STEP
C
C
          EL(ISEG)=EL(ISEG)
          RETURN
          END
C
C.pa
C     DEBUG UNIT(6),SUBCHK
C     END DEBUG
      SUBROUTINE CLDAF(PRM,PR,I,NSEG,XGLO,PSK,SK,
     &               FGLO,FGLOB,PRINT              )
C*********************************************************************
C     CALCULATE SEGMENT GLOBAL FORCES
C*********************************************************************
      DIMENSION XGLO(NSEG,12),PSK(NSEG,12,12),SK(12,12),PR(NSEG,9)
      DIMENSION FGLOB(NSEG,12),FGLO(12,12)
      DIMENSION DLOCAL(12,12),PRM(12,12)
      DIMENSION XGLOB(12,12),DFGLO(12,12),DFLOC(12,12)
      LOGICAL PRINT
C
CPREVIOUS STEP SEGMENT'S GLOBAL DISPLACEMENT INCREMENT
C
      DO 89 II=1,12
        XGLOB(II,1)=XGLO(I,II)
   89 CONTINUE
C
C PREVIOUS STEP ELEMENT STIFFNESS
C
      DO 98 II=1,12
      DO 98 JJ=1,12
        SK(II,JJ)=PSK(I,II,JJ)
   98 CONTINUE
      DO 32 L=1,12
      DO 32 J=1,12
        PRM(L,J)=0.
   32 CONTINUE
  959 K=1
   42 PRM(K,K)=PR(I,1)
      PRM(K,K+1)=PR(I,2)
      PRM(K,K+2)=PR(I,3)
      PRM(K+1,K)=PR(I,4)
      PRM(K+1,K+1)=PR(I,5)
      PRM(K+1,K+2)=PR(I,6)
      PRM(K+2,K)=PR(I,7)
      PRM(K+2,K+1)=PR(I,8)
      PRM(K+2,K+2)=PR(I,9)
      K=K+3
      IF (K.EQ.13) GO TO 92
      GO TO 42
   92 DO 610 II=1,12
      DO 610 JJ=1,12
        DLOCAL(II,JJ)=0
      DO 610 KK=1,12
  610   DLOCAL(II,JJ)=DLOCAL(II,JJ)+PRM(II,KK)*XGLOB(KK,JJ)
      DO 620 II=1,12
      DO 620 JJ=1,12
        DFLOC(II,JJ)=0
      DO 620 KK=1,12
  620   DFLOC(II,JJ)=DFLOC(II,JJ)+SK(II,KK)*DLOCAL(KK,JJ)
      DO 630 II=1,12
      DO 630 JJ=1,12
        DFGLO(II,JJ)=0
      DO 630 KK=1,12
  630   DFGLO(II,JJ)=DFGLO(II,JJ)+PRM(KK,II)*DFLOC(KK,JJ)
C
        IF ( PRINT) THEN
          WRITE(6,*)'1','SEGMENT GLOBAL DISPLACEMENT INCREMENT:'
          CALL PRIN(XGLOB,12,1,12,12)
          WRITE(6,*)'1','SEGMENT ROTATION MATRIX:'
          CALL PRIN(PRM,12,12,12,12)
          WRITE(6,*)'1','SEGMENT LOCAL DISPLACEMENT INCREMENT:'
          CALL PRIN(DLOCAL,12,1,12,12)
          WRITE(6,*)' SEGMENT LOCAL FORCE INCREMENT:'
          CALL PRIN(DFLOC,12,1,12,12)
          WRITE(6,*)' SEGMENT GLOBAL FORCE INCREMENT:'
          CALL PRIN(DFGLO,12,1,12,12)
        ENDIF
C
      DO 72 II=1,12
            FGLOB(I,II)=DFGLO(II,1)+FGLOB(I,II)
   72 CONTINUE
      DO 52 J=1,12
          FGLO(J,1)=FGLOB(I,J)
   52 CONTINUE
      IF(PRINT) THEN
           WRITE(6,*)'1','THE SEGMENT',I,'INTERNAL GLOBAL FORCE:'
             WRITE(6,81)(FGLO(J,1),J=1,12)
   81        FORMAT(1X,6G15.6)
       ENDIF
       RETURN
       END
C.pa
C     DEBUG UNIT(6),SUBCHK
C     END DEBUG
      SUBROUTINE CSLF(R, NSEG, ISEG, FLOC, FGLO, PRINT)
C*********************************************************************
C SUBROUTINE CSLF FOR CALCULATE SEGMENT LOCAL FORCE
C*********************************************************************
C
      DIMENSION FLOC(NSEG,12), FGLO(12,12)
      DIMENSION FLOCAL(12,12),R(12,12)
      LOGICAL PRINT
C
C CALCULATE THE SEGMENT LOCAL FORCE
C
      DO 710 II=1,12
          DO 710 JJ=1,12
            FLOCAL(II,JJ)=0
          DO 710 KK=1,12
  710     FLOCAL(II,JJ)=FLOCAL(II,JJ)+R(II,KK)*FGLO(KK,JJ)
C
          DO 54 J=1,12
              FLOC(ISEG,J)=FLOCAL(J,1)
   54     CONTINUE
C         IF(PRINT) THEN
C             WRITE(6,80)ISEG
C  80         FORMAT('1','ELEMENT',I3,' LOCAL FORCES:')
C             WRITE(6,72)(FLOC(ISEG,II),II=1,12)
C  72         FORMAT(1X,6G15.6)
C         ENDIF
          RETURN
          END
C
C.pa
C     DEBUG UNIT(6),SUBCHK
C     END DEBUG
C     SUBROUTINE LIB(NNOO,HH,UU,WW,ZZ,INEB,INEH,
     &                ST,IA1,IA2,IA3,IA4,IECOP,IELNO,
     &                U0,V0,A0,B,ECCX0,ECCT0,NUMP)
C---------------------------------------------------------------------
C   SECT1 : BOX SECTION
C   SECT2 : TUBE SECTION WITH ONE LAYER
C   SECT3 : RECTANGULAR CROSS SECTION
C   SECT4 : WIDE-FLANGE SECTION
C   SECT5 : NOT AVAILABLE NOW
C   SECT6 : TUBE SECTION WITH TWO LAYERS
C   SECT7 : ANGLE SECTION WITH EQUAL LEGS
C   SECT8 : ANGLE SECTION WITH EQUAL LEGS( OPPOSIVE SITE )
C---------------------------------------------------------------------
      DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)
      REAL INEB, INEH
C
C THIS SUBROUTINE IS FOR SEARCHING MEMBER'S CROSS-SECTION LIBRARY
C
      IF(NNOO .EQ. 1 ) THEN
          CALL SECT1(HH,UU,WW,ZZ,INEB,INEH,ST,
     &                U0,V0,A0,B,ECCX0,ECCT0,NUMP,
     $                 IA1,IA2,IA3,IA4,IECOP,IELNO)
       ELSE IF(NNOO .EQ. 2 ) THEN
          CALL SECT2(HH,UU,WW,ZZ,INEB,INEH,ST,
     &                U0,V0,A0,B,ECCX0,ECCT0,NUMP,
     $                 IA1,IA2,IA3,IA4,IECOP,IELNO)
       ELSE IF(NNOO .EQ. 3 ) THEN
          CALL SECT3(HH,UU,WW,ZZ,INEB,INEH,ST,
     &                U0,V0,A0,B,ECCX0,ECCT0,NUMP,
     $                 IA1,IA2,IA3,IA4,IECOP,IELNO)
       ELSE IF(NNOO .EQ. 4 ) THEN
          CALL SECT4(HH,UU,WW,ZZ,INEB,INEH,ST,
     &                U0,V0,A0,B,ECCX0,ECCT0,NUMP,
     $                 IA1,IA2,IA3,IA4,IECOP,IELNO)
       ELSE IF(NNOO .EQ. 5 ) THEN
          CALL SECT5(HH,UU,WW,ZZ,INEB,INEH,ST,
     &                U0,V0,A0,B,ECCX0,ECCT0,NUMP,
     $                 IA1,IA2,IA3,IA4,IECOP,IELNO)
       ELSE IF(NNOO .EQ. 6 ) THEN
          CALL SECT6(HH,UU,WW,ZZ,INEB,INEH,ST,
     &                U0,V0,A0,B,ECCX0,ECCT0,NUMP,
     $                 IA1,IA2,IA3,IA4,IECOP,IELNO)
       ELSE IF(NNOO .EQ. 7 ) THEN
          CALL SECT7(HH,UU,WW,ZZ,INEB,INEH,ST,
     &                U0,V0,A0,B,ECCX0,ECCT0,NUMP,
     $                 IA1,IA2,IA3,IA4,IECOP,IELNO)
       ELSE IF(NNOO .EQ. 8 ) THEN
          CALL SECT8(HH,UU,WW,ZZ,INEB,INEH,ST,
     &                U0,V0,A0,B,ECCX0,ECCT0,NUMP,
     $                 IA1,IA2,IA3,IA4,IECOP,IELNO)
       ENDIF
       RETURN
       END
C
C
C.pa
C     DEBUG UNIT(6),SUBCHK
C     END DEBUG
C     SUBROUTINE SECT1(HH,UU,ECCX0,ECCT0,INEB,INEH,ST,
     &                U0,V0,A0,B,ECCX0,ECCT0,NUMP,
     &                 IA1,IA2,IA3,IA4,IECOP,IELNO)
C---------------------------------------------------------------------
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR
C     THE BOX CROSS-SECTION
C        HH   = BOX HIGH
C        UU   = BOX WIDTH
C        INEB = TOTAL NO. OF ELEMENTS FOR HALF WIDTH
C        INEH = TOTAL NO. OF ELEMENTS FOR HALF HIGH
C        ST   = THICKNESS
C---------------------------------------------------------------------
      DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)
      REAL INEB, INEH
C
      UU=UU-2*ST
      NUMP=4*(INEB+INEH)
      ELH=((HH/2.)-ST)/(INEH-1)
      ELB=(UU/2.)/INEB
      U0(1)=(UU/2.)+(ST/2.)
      V0(1)=-((HH/2.)-(ST/2.))
      A0(1)=ST*ST
      B(1)=ST
      DO 52 I=2,INEH
         V0(I)=-((HH/2.)-(ST+ELH/2.)-(I-2)*ELH)
         U0(I)=(UU/2.)+(ST/2.)
         A0(I)=ELH*ST
         B(I)=ELH
   52 CONTINUE
      DO 53 I=1,INEB
         J=I+INEH
         V0(J)=-((HH/2.)-(ST/2.))
         U0(J)=((UU/2.)-(ELB/2.))-(I-1)*ELB
         A0(J)=ELB*ST
         B(J)=ELB
   53 CONTINUE
      NEOQ=INEB+INEH
      DO 54 I=1,NEOQ
         J=NEOQ+I
         U0(J)=U0(I)
         V0(J)=-V0(I)
         A0(J)=A0(I)
         B(J)=B(I)
         JJ=(NEOQ*2)+I
         U0(JJ)=-U0(I)
         V0(JJ)=V0(I)
         A0(JJ)=A0(I)
         B(JJ)=B(I)
         JJJ=(NEOQ*3)+I
         U0(JJJ)=-U0(I)
         V0(JJJ)=-V0(I)
         A0(JJJ)=A0(I)
         B(JJJ)=B(I)
   54 CONTINUE
      IF( IECOP .EQ. 1 ) THEN
         ECCX0=ECCX0
         ECCT0=ECCT0
      ENDIF
      RETURN
      END
C
C.pa
C     DEBUG UNIT(6),SUBCHK
```

```
C     END DEBUG                                                          HTS22430
C                                                                        HTS22440
      SUBROUTINE SECT2(RAD,SPACE1,ECCXO,ECCTO,INEB,ISPACE,ST,            HTS22450
     &                 UO,VO,AO,B,ECCXO,ECCTO,NUMP,                      HTS22460
     $                  IA1,IA2,IA3,IA4,IECOP,IELNO)                     HTS22470
C-----------------------------------------------------------------       HTS22480
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR          HTS22490
C     THE TUBE CROSS-SECTION                                             HTS22500
C       RAD   =  RADIUS OF THE TUBE (BASED ON CENTER LINE                HTS22510
C       INEB  =  TOTAL NO. OF ELEMENTS FOR 1/4 OF THE CIRCLE             HTS22520
C       ST    =  THICKNESS                                               HTS22530
C-----------------------------------------------------------------       HTS22540
      DIMENSION UO(NUMP),VO(NUMP),AO(NUMP),B(NUMP)                       HTS22550
      REAL INEB, INEH                                                    HTS22560
      NUMP=4*INEB                                                        HTS22570
      PI=2.*3.1415926                                                    HTS22580
      SITA=(PI/4.)/(2*INEB)                                              HTS22590
      TEMP=2.*SITA                                                       HTS22600
      DO 25 I=1,INEB                                                     HTS22610
        VO(I)=SQRT(RAD*RAD/(1.+TAN(SITA)*TAN(SITA)))                     HTS22620
        UO(I)=VO(I)*TAN(SITA)                                            HTS22630
   25   SITA=SITA+TEMP                                                   HTS22640
      DO 30 I=INEB+1,2*INEB                                              HTS22650
        VO(I)=VO(I-INEB)                                                 HTS22660
   30   UO(I)=-UO(I-INEB)                                                HTS22670
      DO 35 I=2*INEB+1,3*INEB                                            HTS22680
        VO(I)=-VO(I-2*INEB)                                              HTS22690
   35   UO(I)=UO(I-2*INEB)                                               HTS22700
      DO 40 I=3*INEB+1,4*INEB                                            HTS22710
        VO(I)=-VO(I-3*INEB)                                              HTS22720
   40   UO(I)=-UO(I-3*INEB)                                              HTS22730
      DO 50 I=1,NUMP                                                     HTS22740
        B(I)=(PI*RAD)/NUMP                                               HTS22750
        AO(I)=B(I)*ST                                                    HTS22760
   50 CONTINUE                                                           HTS22770
      IF(IECOP .EQ. 1 ) THEN                                             HTS22780
        ECCXO=ECCXO                                                      HTS22790
        ECCTO=ECCTO                                                      HTS22800
      ENDIF                                                              HTS22810
      RETURN                                                             HTS22820
      END                                                                HTS22830
C                                                                        HTS22840
C                                                                        HTS22850
C                                                                        HTS22860
C.pa                                                                     HTS22870
C     DEBUG UNIT(6),SUBCHK                                               HTS22880
C     END DEBUG                                                          HTS22890
      SUBROUTINE SECT3(HH,UU,ECCXO,ECCTO,INEB,INEH,ST,                   HTS22900
     &                 UO,VO,AO,B,ECCXO,ECCTO,NUMP,                      HTS22910
     &                  IA1,IA2,IA3,IA4,IECOP,IELNO)                     HTS22920
C-----------------------------------------------------------------       HTS22930
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR          HTS22940
C     THE RECTANGULAR CROSS-SECTION                                      HTS22950
C        HH    = SECTIONAL HIGH                                          HTS22960
C        UU    = SECTIONAL WIDTH                                         HTS22970
C        INEB  = TOTAL NO. OF ELEMENTS FOR HALF WIDTH                    HTS22980
C        INEH  = TOTAL NO. OF ELEMENTS FOR HALF HIGH                     HTS22990
C        ST    = DUMMY VARIABLES                                         HTS23000
C        B(1) = HH/2 (LARGER LENGTH)                                     HTS23010
C        B(2) = UU/2 (SMALLER LENGTH)                                    HTS23020
C        NOTE : B(1) AND B(2) ARE FOR CALCULATING TORSIONAL RIGIDITY     HTS23030
C-----------------------------------------------------------------       HTS23040
      DIMENSION UO(NUMP),VO(NUMP),AO(NUMP),B(NUMP)                       HTS23050
      REAL INEB, INEH                                                    HTS23060
C                                                                        HTS23070
      NUMP=4*(INEB*INEH)                                                 HTS23080
      ELH=HH/(2.*INEH)                                                   HTS23090
      ELB=UU/(2.*INEB)                                                   HTS23100
C                                                                        HTS23110
      DO 50 M=1,INEB                                                     HTS23120
       II=(M-1)*INEH + 1                                                 HTS23130
       UO(II)=(UU/2)-(ELB/2)-(M-1)*ELB                                   HTS23140
       VO(II)=(HH/2)-(ELH/2)                                             HTS23150
       AO(II)=ELH*ELB                                                    HTS23160
       J=1                                                               HTS23170
       DO 60 I=(M-1)*INEH+2, M*INEH                                      HTS23180
          J=J+1                                                          HTS23190
          UO(I)=UO(II)                                                   HTS23200
          VO(I)=(HH/2)-(J-1)*ELH-(ELH/2)                                 HTS23210
          AO(I)=ELH*ELB                                                  HTS23220
   60  CONTINUE                                                          HTS23230
   50 CONTINUE                                                           HTS23240
C                                                                        HTS23250
      IQRT=INEB*INEH                                                     HTS23260
      DO 70 I=IQRT+1,2*IQRT                                              HTS23270
         UO(I)=UO(I-IQRT)                                                HTS23280
         VO(I)=-VO(I-IQRT)                                               HTS23290
         AO(I)=ELH*ELB                                                   HTS23300
   70 CONTINUE                                                           HTS23310
      DO 80 I=2*IQRT+1,3*IQRT                                            HTS23320
         UO(I)=-UO(I-2*IQRT)                                             HTS23330
         VO(I)=VO(I-2*IQRT)                                              HTS23340
         AO(I)=ELH*ELB                                                   HTS23350
   80 CONTINUE                                                           HTS23360
      DO 90 I=3*IQRT+1,4*IQRT                                            HTS23370
         UO(I)=-UO(I-3*IQRT)                                             HTS23380
         VO(I)=-VO(I-3*IQRT)                                             HTS23390
         AO(I)=ELH*ELB                                                   HTS23400
   90 CONTINUE                                                           HTS23410
C                                                                        HTS23420
      IF( HH .GE. UU ) THEN                                              HTS23430
         B(1)=HH/2                                                       HTS23440
         B(2)=UU/2                                                       HTS23450
      ELSE IF( HH .LT. UU ) THEN                                         HTS23460
         B(1)=UU/2                                                       HTS23470
         B(2)=HH/2                                                       HTS23480
      ENDIF                                                              HTS23490
C                                                                        HTS23500
      IF( IECOP .EQ. 1 ) THEN                                            HTS23510
         ECCXO=ECCXO                                                     HTS23520
         ECCTO=ECCTO                                                     HTS23530
      ENDIF                                                              HTS23540
      RETURN                                                             HTS23550
      END                                                                HTS23560
C                                                                        HTS23570
C.pa                                                                     HTS23580
C     DEBUG UNIT(6),SUBCHK                                               HTS23590
C     END DEBUG                                                          HTS23600
C                                                                        HTS23610
@PROCESS SDUMP GOSTMT XREF                                               HTS23620
      SUBROUTINE SECT4(HH,UU,ECCXO,ECCTO,INEB,INEH,ST,                   HTS23630
     &                 UO,VO,AO,B,ECCXO,ECCTO,NUMP,                      HTS23640
     &                  IREV1,IREV2,IREV3,IREV4,IECOP,IELNO)             HTS23650
C-----------------------------------------------------------------       HTS23660
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR          HTS23670
C     THE WIDE-FLANGE SECTION                                            HTS23680
C          ST    = FLANGE THICKNESS                                      HTS23690
C          INEB  = WEB THICKNESS                                         HTS23700
C          HH    = SECTION HIGH                                          HTS23710
C          UU    = SECTION WIDTH                                         HTS23720
C          IREV1= NO. OF FLANGE COLUMN IN THE REFERENCE AXIS U DIRECTIONHTS23730
C          IREV2= NO. OF FLANGE ROW    IN THE REFERENCE AXIS U DIRECTIONHTS23740
C          IREV3= NO. OF WEB COLUMN    IN THE REFERENCE AXIS V DIRECTIONHTS23750
C          IREV4= NO. OF WEB ROW       IN THE REFERENCE AXIS V DIRECTIONHTS23760
C-----------------------------------------------------------------       HTS23770
      DIMENSION UO(NUMP),VO(NUMP),AO(NUMP),B(NUMP)                       HTS23780
      REAL INEB, INEH                                                    HTS23790
C                                                                        HTS23800
      NUMP=IREV1*IREV2*2 + IREV3*IREV4                                   HTS23810
      WEBHIGH = HH- 2*ST                                                 HTS23820
      SS = UU/IREV2                                                      HTS23830
      TT = ST/IREV1                                                      HTS23840
      SSS = INEB/IREV3                                                   HTS23850
      TTT = WEBHIGH/IREV4                                                HTS23860
      B(1)= (2*UU*ST**3 + WEBHIGH*INEB**3)/3                             HTS23870
C                                                                        HTS23880
C....CALCULATE TOP FLANGE ELEMENT COORDINATES                            HTS23890
      K=0                                                                HTS23900
      DO 50 I=1, IREV1                                                   HTS23910
        DO 50 J=1,IREV2                                                  HTS23920
          K=K+1                                                          HTS23930
          UO(K)=(-UU/2) + 0.5*SS + (J-1)*SS                              HTS23940
          VO(K)=( HH/2) - 0.5*TT - (I-1)*TT                              HTS23950
          AO(K)=SS*TT                                                    HTS23960
   50 CONTINUE                                                           HTS23970
C                                                                        HTS23980
C....CALCULATE BOTTOM FLANGE ELEMENT COORDINATES                         HTS23990
      DO 60 I=1, IREV1                                                   HTS24000
        DO 60 J=1,IREV2                                                  HTS24010
          K=K+1                                                          HTS24020
          UO(K)=(-UU/2) + 0.5*SS + (J-1)*SS                              HTS24030
          VO(K)=(-HH/2) + 0.5*TT + (I-1)*TT                              HTS24040
          AO(K)=SS*TT                                                    HTS24050
   60 CONTINUE                                                           HTS24060
C                                                                        HTS24070
C....CALCULATE WEB ELEMENT COORDINATES                                   HTS24080
      DO 70 I=1, IREV4                                                   HTS24090
        DO 70 J=1,IREV3                                                  HTS24100
          K=K+1                                                          HTS24110
          UO(K)=(-INEB/2) + 0.5*SSS + (J-1)*SSS                          HTS24120
          VO(K)=(WEBHIGH/2) - 0.5*TTT - (I-1)*TTT                        HTS24130
          AO(K)=SSS*TTT                                                  HTS24140
   70 CONTINUE                                                           HTS24150
C                                                                        HTS24160
      IF( IECOP .EQ. 1 ) THEN                                            HTS24170
        ECCXO=ECCXO                                                      HTS24180
        ECCTO=ECCTO                                                      HTS24190
      ENDIF                                                              HTS24200
      RETURN                                                             HTS24210
      END                                                                HTS24220
C.pa                                                                     HTS24230
C     DEBUG UNIT(6),SUBCHK                                               HTS24240
C     END DEBUG                                                          HTS24250
C                                                                        HTS24260
      SUBROUTINE SECT5(BB,T,D,SPACE1,NCS,NPS,T1,                         HTS24270
     &                 UO,VO,AO,B,ECCXO,ECCTO,NUMP,                      HTS24280
     $                  IA1,IA2,IA3,IA4,IECOP,IELNO)                     HTS24290
C-----------------------------------------------------------------       HTS24300
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR          HTS24310
C     THE OPEN WEB JOIST PLATE-CHORD CROSS-SECTION(BOTTOM)               HTS24320
C          BB    = PLATE WIDTH                                           HTS24330
C          T     = PLATE THICKNESS                                       HTS24340
C          T1    = CHORD THICKNESS                                       HTS24350
C          NPS   = TOTAL NO. OF ELEMENTS FOR HALF PLATE WIDTH            HTS24360
C          NCS   = TOTAL NO. OF ELEMENTS FOR EACH ANGLE CHORD'S TOE      HTS24370
C          D     = THE ANGLE'S LENGTH                                    HTS24380
C-----------------------------------------------------------------       HTS24390
      DIMENSION UO(NUMP),VO(NUMP),AO(NUMP),B(NUMP)                       HTS24400
      REAL INEB, INEH                                                    HTS24410
C                                                                        HTS24420
      IF(IECOP .EQ. 1 ) THEN                                             HTS24430
       WRITE(6,*)'1','SECT5 LIBRARY FOR ECCENTRIC LOAD IS NOT',          HTS24440
     $          'AVAILABLE.............1'                                HTS24450
       IECOP=0                                                           HTS24460
      ENDIF                                                              HTS24470
      NUMP=4*NCS+2*NPS+2                                                 HTS24480
      S=-((-.5*BB*T**2)-(4.*T1*T*D+D*T1**2 +T1*D**2-2.*T*T1**2-T1**3))   HTS24490
     $  /(BB*T+4.*T1*D-2.*T1**2)                                         HTS24500
      DS=(D-T1)/(2.*NCS)                                                 HTS24510
      UO(1)=(BB/2.)-(T1/2.)                                              HTS24520
      VO(1)=-(S-(T+0.5*T1))                                              HTS24530
      AO(1)=T1**2                                                        HTS24540
      B(1)=T1                                                            HTS24550
      DO 52 I=2,NCS+1                                                    HTS24560
        UO(I)=(0.5*BB-D-DS)+2.*DS*(I-1)                                  HTS24570
        VO(I)=-(S-(T+0.5*T1))                                            HTS24580
        AO(I)=2.*DS*T1                                                   HTS24590
        B(I)=2.*DS                                                       HTS24600
   52 CONTINUE                                                           HTS24610
       DO 54 I=NCS+2, 2*NCS+1                                            HTS24620
        UO(I)=0.5*BB-0.5*T1                                              HTS24630
        VO(I)=(D+T-S)+DS-2.*DS*(I-NCS-1)                                 HTS24640
        AO(I)=2.*DS*T1                                                   HTS24650
        B(I)=2.*DS                                                       HTS24660
   54 CONTINUE                                                           HTS24670
      UO(2*NCS+2)=-UO(1)                                                 HTS24680
      VO(2*NCS+2)=VO(1)                                                  HTS24690
      AO(2*NCS+2)=AO(1)                                                  HTS24700
      B(2*NCS+2)=B(1)                                                    HTS24710
      DO 56 I=2*NCS+3, 3*NCS+2                                           HTS24720
        UO(I)=-(0.5*BB-D-DS)-2.*DS*(I-2*NCS-2)                           HTS24730
        VO(I)=-(S-(T+0.5*T1))                                            HTS24740
        AO(I)=2.*DS*T1                                                   HTS24750
        B(I)=2.*DS                                                       HTS24760
   56 CONTINUE                                                           HTS24770
      DO 58 I=3*NCS+3, 4*NCS+2                                           HTS24780
        UO(I)=-0.5*BB+0.5*T1                                             HTS24790
        VO(I)=(D+T-S)+DS-2.*DS*(I-3*NCS-2)                               HTS24800
        AO(I)=2.*DS*T1                                                   HTS24810
        B(I)=2.*DS                                                       HTS24820
   58 CONTINUE                                                           HTS24830
      DBB=0.5*BB/(2.*NPS)                                                HTS24840
      DO 60 I=4*NCS+3, 4*NCS+3+NPS-1                                     HTS24850
        UO(I)=0.5*BB+DBB-2.*DBB*(I-4*NCS-2)                              HTS24860
        VO(I)=-(S-0.5*T)                                                 HTS24870
        AO(I)=2.*DBB*T                                                   HTS24880
        B(I)=2.*DBB                                                      HTS24890
   60 CONTINUE                                                           HTS24900
      DO 64 I=4*NCS+NPS+3, 4*NCS+2*NPS+2                                 HTS24910
        UO(I)=-0.5*BB-DBB+2.*DBB*(I-4*NCS-NPS-2)                         HTS24920
        VO(I)=-(S-0.5*T)                                                 HTS24930
        AO(I)=2.*DBB*T                                                   HTS24940
        B(I)=2.*DBB                                                      HTS24950
   64 CONTINUE                                                           HTS24960
      RETURN                                                             HTS24970
      END                                                                HTS24980
C                                                                        HTS24990
C.pa                                                                     HTS25000
C     DEBUG UNIT(6),SUBCHK                                               HTS25010
C     END DEBUG                                                          HTS25020
C                                                                        HTS25030
      SUBROUTINE SECT6(RAD,SPACE1,ECCXO,ECCTO,INEB,ISPACE,ST,            HTS25040
     &                 UO,VO,AO,B,ECCXO,ECCTO,NUMP,                      HTS25050
     $                  IA1,IA2,IA3,IA4,IECOP,IELNO)                     HTS25060
C-----------------------------------------------------------------       HTS25070
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR          HTS25080
C     THE TUBE CROSS-SECTIONWHICH HAS TWO LAYERS                         HTS25090
C       RAD   =  RADIUS OF THE TUBE (BASED ON CENTER LINE                HTS25100
C       INEB  =  TOTAL NO. OF ELEMENTS FOR 1/4 OF THE CIRCLE FOR         HTS25110
C                EACH LAYER                                              HTS25120
C       ST    =  THICKNESS                                               HTS25130
C-----------------------------------------------------------------       HTS25140
      DIMENSION UO(NUMP),VO(NUMP),AO(NUMP),B(NUMP)                       HTS25150
      REAL INEB, INEH                                                    HTS25160
      NUMP=8*INEB                                                        HTS25170
      NNUMP=NUMP/2                                                       HTS25180
      RAD1=RAD+(ST/4.)                                                   HTS25190
      RAD2=RAD-(ST/4.)                                                   HTS25200
      PI=2.*3.1415926                                                    HTS25210
      SITA=(PI/4.)/(2*INEB)                                              HTS25220
      TEMP=2.*SITA                                                       HTS25230
      DO 25 I=1,INEB                                                     HTS25240
        VO(I)=SQRT(RAD1*RAD1/(1.+TAN(SITA)*TAN(SITA)))                   HTS25250
        UO(I)=VO(I)*TAN(SITA)                                            HTS25260
```

```
        V0(I+NNUMP)=SQRT(RAD2*RAD2/(1.+TAN(SITA)*TAN(SITA)))
        U0(I+NNUMP)=V0(I+NNUMP)*TAN(SITA)
   25   SITA=SITA+TEMP
        DO 30 I=INEB+1,2*INEB
          V0(I)=V0(I-INEB)
          U0(I)=-U0(I-INEB)
          V0(I+NNUMP)=V0(I-INEB+NNUMP)
   30   U0(I+NNUMP)=-U0(I-INEB+NNUMP)
        DO 35 I=2*INEB+1,3*INEB
          V0(I)=-V0(I-2*INEB)
          U0(I)=U0(I-2*INEB)
          V0(I+NNUMP)=-V0(I-2*INEB+NNUMP)
   35   U0(I+NNUMP)=U0(I-2*INEB+NNUMP)
        DO 40 I=3*INEB+1,4*INEB
          V0(I)=-V0(I-3*INEB)
          U0(I)=-U0(I-3*INEB)
          V0(I+NNUMP)=-V0(I-3*INEB+NNUMP)
   40   U0(I+NNUMP)=-U0(I-3*INEB+NNUMP)
        DO 50 I=1,NNUMP
          B(I)=(PI*RAD1)/NNUMP
          A0(I)=B(I)*(ST/2.)
          B(I+NNUMP)=(PI*RAD2)/NNUMP
          A0(I+NNUMP)=B(I+NNUMP)*(ST/2.)
   50   CONTINUE
        IF(IECOP .EQ. 1 ) THEN
          ECCX0=ECCX0
          ECCY0=ECCY0
        ENDIF
        RETURN
        END
C
C.pa
C       DEBUG UNIT(6),SUBCHK
C       END DEBUG
C
        SUBROUTINE SECT7(D,SPACE1,ECCX0,ECCY0,NCS,ISPACE,T,
     &                   U0,V0,A0,B,ECCX0,ECCY0,NUMP,
     &                    L1,N1,L2,N2,IECOP,IELNO)
C-----------------------------------------------------------------
C       THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR
C       THE ANGLE CROSS-SECTION
C            D    = ANGLE SIDE TOTAL LENGTH
C            T    = THICKNESS
C            NCS  = DUMMY VARIABLE
C            ECCX0= THE X-DISTANCE FROM ECCENTRIC LOAD TO SHAPE CENTER
C            ECCY0= THE Y-DISTANCE FROM ECCENTRIC LOAD TO SHAPE CENTER
C            L1   = NO. OF COLUMN IN X DIRECTION
C            N1   = NO OF ROW IN X DIRECTION
C            L2   = NO. OF ROW IN Y DIRECTION
C            N2   = NO OF COLUMN IN Y DIRECTION
C            IECOP= ECCENTRIC LOAD OPTION
C                   1: ECCENTRIC LOAD APPLIED
C                   0: WITHOUT ECCENTRIC LOAD
C-----------------------------------------------------------------
        DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)
        REAL INEB, INEH
        BB=(D*T+D**2-T**2)/(4.*D-2.*T)
        ST1=(T/N1)*0.5
        SX1=(D/L1)*0.5
        DO 101 J=1,N1
           II=J-1
        DO 101 I=1,L1
           M=II*L1+I
           U0(M)=BB-(D-(2*I-1)*SX1)
           V0(M)=BB-(2*J-1)*ST1
           A0(M)=ST1*SX1*4.
           B(M)=2.*SX1
  101 CONTINUE
C
        SX2=(T/N2)*0.5
        ST2=((D-T)/L2)*0.5
        DO 201 J=1,N2
           II=J-1
        DO 201 I=1,L2
           M1=II*L2+I+(N1*L1)
           U0(M1)=BB-(2*J-1)*SX2
           V0(M1)=BB-(D-(2*I-1)*ST2)
           A0(M1)=ST2*SX2*4.
           B(M1)=2.*ST2
  201 CONTINUE
        NUMP=L1*N1+L2*N2
        IF(IECOP .EQ. 1 ) THEN
          ECCX0=ECCX0
          ECCY0=ECCY0
        ENDIF
        RETURN
        END
C
C
C.pa
C       DEBUG UNIT(6),SUBCHK
C       END DEBUG
        SUBROUTINE SECT8(D,SPACE1,ECCX0,ECCY0,NCS,ISPACE,T,
     &                   U0,V0,A0,B,ECCX0,ECCY0,NUMP,
     $                    L1,N1,L2,N2,IECOP,NOST)
C-----------------------------------------------------------------
C       THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR
C       THE ANGLE CROSS-SECTION (BOTTOM CHORD)
C            D    = ANGLE SIDE TOTAL LENGTH
C            T    = THICKNESS
C            NCS  = TOTAL NO. OF ELEMENTS FOR EACH TOE
C            ECCX0= THE X-DISTANCE FROM ECCENTRIC LOAD TO SHAPE CENTER
C            ECCY0= THE Y-DISTANCE FROM ECCENTRIC LOAD TO SHAPE CENTER
C            L1   = NO. OF COLUMN IN X DIRECTION
C            N1   = NO OF ROW IN X DIRECTION
C            L2   = NO. OF ROW IN Y DIRECTION
C            N2   = NO OF COLUMN IN Y DIRECTION
C            IECOP= ECCENTRIC LOAD OPTION
C                   1: ECCENTRIC LOAD APPLIED
C                   0: WITHOUT ECCENTRIC LOAD
C-----------------------------------------------------------------
        DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)
        REAL INEB, INEH
        BB=(D*T+D**2-T**2)/(4.*D-2.*T)
        ST1=(T/N1)*0.5
        SX1=(D/L1)*0.5
        DO 101 J=1,N1
           II=J-1
        DO 101 I=1,L1
           M=II*L1+I
           U0(M)=BB-(D-(2*I-1)*SX1)
           V0(M)=-(BB-(2*J-1)*ST1)
           A0(M)=ST1*SX1*4.
           B(M)=2.*SX1
  101 CONTINUE
C
        SX2=(T/N2)*0.5
        ST2=((D-T)/L2)*0.5
        DO 201 J=1,N2
           II=J-1
        DO 201 I=1,L2
           M1=II*L2+I+(N1*L1)
           U0(M1)=BB-(2*J-1)*SX2
           V0(M1)=-(BB-(D-(2*I-1)*ST2))
           A0(M1)=ST2*SX2*4.
           B(M1)=2.*ST2
  201 CONTINUE
        NUMP=L1*N1+L2*N2
        ECCX0=ECCX0
        ECCY0=ECCY0
        RETURN
        END
C
C
C.pa
C       DEBUG UNIT(6),SUBCHK
C       END DEBUG
C
        SUBROUTINE SECT9(D,SPACE1,ECCX0,ECCY0,NCS,ISPACE,T,
     &                   U0,V0,A0,B,ECCX0,ECCY0,NUMP,
     &                    L1,N1,L2,N2,IECOP,IELNO)
C-----------------------------------------------------------------
C       THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR
C       THE ANGLE CROSS-SECTION
C            D    = ANGLE SIDE TOTAL LENGTH
C            T    = THICKNESS
C            NCS  = DUMMY VARIABLE
C            ECCX0= THE X-DISTANCE FROM ECCENTRIC LOAD TO SHAPE CENTER
C            ECCY0= THE Y-DISTANCE FROM ECCENTRIC LOAD TO SHAPE CENTER
C            L1   = NO. OF COLUMN IN X DIRECTION
C            N1   = NO OF ROW IN X DIRECTION
C            L2   = NO. OF ROW IN Y DIRECTION
C            N2   = NO OF COLUMN IN Y DIRECTION
C            IECOP= ECCENTRIC LOAD OPTION
C                   1: ECCENTRIC LOAD APPLIED
C                   0: WITHOUT ECCENTRIC LOAD
C-----------------------------------------------------------------
        DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)
        REAL INEB, INEH
        BB=(D*T+D**2-T**2)/(4.*D-2.*T)
        ST1=(T/N1)*0.5
        SX1=(D/L1)*0.5
        DO 101 J=1,N1
           II=J-1
        DO 101 I=1,L1
           M=II*L1+I
           U0(M)=BB-(D-(2*I-1)*SX1)
           V0(M)=BB-(2*J-1)*ST1
           A0(M)=ST1*SX1*4.
           B(M)=2.*SX1
  101 CONTINUE
C
        SX2=(T/N2)*0.5
        ST2=((D-T)/L2)*0.5
        DO 201 J=1,N2
           II=J-1
        DO 201 I=1,L2
           M1=II*L2+I+(N1*L1)
           U0(M1)=BB-(2*J-1)*SX2
           V0(M1)=-(BB-(D-(2*I-1)*ST2))
           A0(M1)=ST2*SX2*4.
           B(M1)=2.*ST2
  201 CONTINUE
        NUMP=L1*N1+L2*N2
        ECCX0=ECCX0
        ECCY0=ECCY0
        RETURN
        END
C
C
C.pa
C       DEBUG UNIT(6),SUBCHK
C       END DEBUG
        SUBROUTINE PRIN(A,N,M,NN,MM)
C
C SUBROUTINE PRIN
C
        DIMENSION A(NN,MM)
        DO 1 I=1,N
        WRITE(6,2)(A(I,J),J=1,M)
      2 FORMAT(1X,6E15.5)
      1 CONTINUE
        WRITE(6,789)
    789 FORMAT(1X,'-------------------------------------------------------')
        RETURN
        END
C.pa
C       DEBUG UNIT(6),SUBCHK
C       END DEBUG
        SUBROUTINE FORCE(FF1,PRINT,IDOF,IELNO,NDOST,NSEG,
     &                   FGLOB ,NP)
C       ================================================================
C       TRANSFERING SEGMENT INTERNAL FORCE TO GLOBAL FORCE DIRECTION
C       ================================================================
C
        LOGICAL PRINT
        DIMENSION FF1(NDOST,1),FGLOB(NSEG,12)
        DIMENSION NP(32   ,12)
        DO 804 I=1,IDOF
            FF1(I,1)=0.
  804 CONTINUE
        DO 805 J=1,NSEG
           DO 805 K=1,12
              FF1(NP(J,K),1)=FF1(NP(J,K),1)+FGLOB(J,K)
  805 CONTINUE
        RETURN
        END
C
C
C.pa
C       DEBUG UNIT(6),SUBCHK
C       END DEBUG
        SUBROUTINE ECCENT(NSEG,UC0,VC0,ISEG,ROZA,ROZB,
     &                    ECX ,ECY,ECCX,ECCY,PRINT)
C       ================================
C       CALCULATE MEMBER-END ECCENTRICITIES
C       ================================
        DIMENSION ECX(2),ECY(2),ECCX(2),ECCY(2)
        LOGICAL PRINT
C
C ECCENTRICITIES ABOUT SECTION REFERENCE COORDINATES
C
        IF( ISEG .EQ. 1) THEN
          EXX=UC0-ECX(1)
          EYY=VC0-ECY(1)
        ELSE IF( ISEG .EQ. NSEG) THEN
          EXX=UC0-ECX(2)
          EYY=VC0-ECY(2)
        ENDIF
C NOTE: EXX AND EYY ARE VECTORS FROM LOAD APPLIED POINT TO CENTROID
C       IN SECTION REFERENCE COORDINATES DIRECTION
C       ECX IS THE DISTANCE FROM REFERENCE COORDINATE ORIGIN TO
C       LOAD APPLIED LOCATION
C
C ECCENTRICITIES ABOUT MEMBER COORDINATES
C
        IF( ISEG .EQ. 1 ) THEN
          ECCX(1)=COS(ROZA)*EXX - SIN(ROZA)*EYY
          ECCY(1)=SIN(ROZA)*EXX + COS(ROZA)*EYY
        ELSE IF( ISEG .EQ. NSEG ) THEN
          ECCX(2)=COS(ROZB)*EXX - SIN(ROZB)*EYY
          ECCY(2)=SIN(ROZB)*EXX + COS(ROZB)*EYY
        ENDIF
C
        IF( PRINT) THEN
          WRITE(6,*)' ROZA,EXX,EYY,ECCX(1),ECCY(1) FROM ECCENT SUBROUTINE'
          WRITE(6,78)ROZA, EXX, EYY,ECCX(1),ECCY(1)
   78     FORMAT(2X, 5G15.4)
        ENDIF
C NOTE: ECCX AND ECCY ARE VECTORS FROM LOAD APPLIED POINT TO CENTROID
C       IN MEMBER COORDINATES DIRECTION
C
        RETURN
        END
C.pa
C       DEBUG UNIT(6),SUBCHK
C       END DEBUG
        SUBROUTINE TRANK(FFASAT,ECCX,ECCY,TPJT)
C       ================================================
C       TRANSFER MEMBER STIFFNESS MATRIX INTO ECCENTRICITY LOAD
C       APPLY DIRECTION
C       ================================================
        DIMENSION ECCX(2),ECCY(2),TPJT(12,12)
        DIMENSION FFASAT(12,12),TPJ(12,12),ASATPJ(12,12)
        DO 250 I=1,12
        DO 250 J=1,12
          TPJ(I,J)=0.
          IF( I .EQ. J ) TPJ(I,J)=1.
  250 CONTINUE
        TPJ(4,2)=0.
        TPJ(4,3)=ECCY(1)
        TPJ(5,1)=0.
        TPJ(5,3)=-ECCX(1)
        TPJ(6,1)=-ECCY(1)
        TPJ(6,2)=ECCX(1)
        TPJ(10,8)=0.
        TPJ(10,9)=ECCY(2)
        TPJ(11,7)=0.
        TPJ(11,9)=-ECCX(2)
        TPJ(12,7)=-ECCY(2)
        TPJ(12,8)=ECCX(2)
C
C TRANSPOSE OF TPJ
C
        DO 255 I=1,12
        DO 255 J=1,12
          TPJT(I,J)=TPJ(J,I)
  255 CONTINUE
C
C |EEASAT|*|TPJT|
C
        DO 260 I=1,12
          DO 260 J=1,12
            ASATPJ(I,J)=0.
            DO 260 K=1,12
              ASATPJ(I,J)=ASATPJ(I,J)+FFASAT(I,K)*TPJT(K,J)
  260 CONTINUE
C
C|TPJ|*|EEASAT|*|TPJT|
C
        DO 270 I=1,12
          DO 270 J=1,12
            FFASAT(I,J)=0.
            DO 270 K=1,12
              FFASAT(I,J)=FFASAT(I,J)+TPJ(I,K)*ASATPJ(K,J)
  270 CONTINUE
        RETURN
        END
C
```

```
      SUBROUTINE CONVER(PRINT,A,B,IDF,ISTOR,NOP,IMD,IS,L1,L2,MDIAG)     HYS20110
C*****************************************************************     HYS20120
C***     CONVERT STIFFNESS MATRIX TO FULL STORAGE MODE          ***     HYS20130
C*****************************************************************     HYS20140
C NOTE : CONVERT ONLY TAKES PLACE FROM L1 TO L2                         HYS20150
C  A     : INPUT HALF STORAGE BANDED MATRIX                             HYS20160
C  B     : CONVERTED MATRIX                                             HYS20170
C ISTOR : SIZE OF A MATRIX                                             HYS20180
C NOP   : DOF OF A MATRIX                                              HYS20190
C MDIAG : A MATRIX MAIN DIAGONAL TERM ADDRESS                          HYS20200
C IS    : B MSATRIX SIZE                                               HYS20210
C                                                                      HYS20220
      LOGICAL PRINT                                                    HYS20230
      DIMENSION A(ISTOR),B(IS,IS),MDIAG(  199)                         HYS20240
                                                                       HYS20250
      DO 200 J=L1,L2,+1                                                HYS20260
        DO 200 I=J,L1,-1                                               HYS20270
          IJ=MDIAG(J)+J-I                                              HYS20280
          IF(IJ .LT. MDIAG(J+1) .OR. IJ .EQ. MDIAG(J) ) THEN           HYS20290
              B(I-L1+1,J-L1+1)=A(IJ)                                   HYS20300
              B(J-L1+1,I-L1+1)=B(I-L1+1,J-L1+1)                        HYS20310
C           ELSE                                                       HYS20320
C               B(I-L1+1,J-L1+1)=0                                     HYS20330
C               B(J-L1+1,I-L1+1)=0                                     HYS20340
            ENDIF                                                      HYS20350
  200 CONTINUE                                                         HYS20360
C                                                                      HYS20370
      IF (PRINT) THEN                                                  HYS20380
          WRITE(6,*)' CONVERT HALF STORAGE BANDED MODE TO'             HYS20390
     $          //' FULL STORAGE MODE:'                               HYS20400
          WRITE(6,*)'THE INPUT MATRIX:'                                HYS20410
          WRITE(6,67)(A(I),I=1,IMD)                                    HYS20420
   67     FORMAT(1X,F15.6)                                             HYS20430
          WRITE(6,*)'CONVERT ONLY TAKE PLACE FROM ',L1,' TO ',L2       HYS20440
          DO 77 I=1,L2-L1+1                                            HYS20450
             WRITE(6,78)(B(I,J),J=1,L2-L1+1)                           HYS20460
   78        FORMAT(1X,6F15.6)                                         HYS20470
   77     CONTINUE                                                     HYS20480
      ENDIF                                                            HYS20490
C                                                                      HYS20500
      RETURN                                                           HYS20510
      END                                                              HYS20520
C                                                                      HYS20530
C.PA                                                                   HYS20540
CPROCESS SOUMP OPT(3) GOSTMT XREF                                      HYS20550
C*****************************************************************     HYS20560
C          DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    HYS20570
C          END DEBUG                                                   HYS20580
C*****************************************************************     HYS20590
C  MATRIX SOLUTION OF EQUATION AX=P, WHERE THE UPPER TRIANGULAR MATRIX  HYS20600
C    OF THE SYMMETRIC MATRIX A IS STORED BY SKYLINES. P IS STORED AS    HYS20610
C    A FULL MATRIX.  THE MATRIX P IS REPLACED BY X AFTER BACK           HYS20620
C    SUBSTITUTION IS COMPLEATE.  BOTH A AND P ARE ALTERED.....         HYS20630
C  GUYAN OPTION: IF GUYAN IS .TRUE. MASS MATRIX IS ALSO CONDENSED      HYS20640
C*****************************************************************     HYS20650
      SUBROUTINE GAUSS2(IOPT,PRINT,N,IMD,LROW,NLC,L1,L2,MD,A,P,FACTOR,   HYS20660
     $                  ISTOR)                                         HYS20670
      DIMENSION A(ISTOR),FACTOR(LROW),P(LROW,NLC),MD( 199  )           HYS20680
      LOGICAL BTEST,ABORT,PRINT                                        HYS20690
C                                                                      HYS20700
      IJI(I,J)=MD(J)+J-I                                               HYS20710
C----------------------------------------------------------------------HYS20720
C   MATRIX STORAGE SCHEME:                                             HYS20730
C  -THE MATRIX A IS BANDED AND SYMMETRIC. THUS ONLY THE SKYLINE ABOVE  HYS20740
C   AND THE MAIN DIAGONAL NEEDS TO BE STORED.                          HYS20750
C   THE MATRIX A IS STORED IN A LINEAR ARRAY BY COLUMNS.  THE VALUE ON HYS20760
C   THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST.             HYS20770
C   ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE MAIN DIAGONAL HYS20780
C   TERMS ARE STORED IN MD.                                            HYS20790
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW...            HYS20800
C        | 1 | 3 |   |   | 12 |                                       HYS20810
C        |   | 2 | 5 |   | 11 |   NOTE: B(1,3),B(1,4) AND B(2,4) ARE ZERO. HYS20820
C   B=   |   |   | 4 | 7 | 10 |                                       HYS20830
C        |   | SYM.  | 6 | 9 |   MD= 1, 2, 4, 6, 8, 13                HYS20840
C        |   |   |   |   | 8 |                                        HYS20850
C   THUS, B(3,3)=A( MD(3) )= A(4)                                      HYS20860
C         B(I,J)=A( MD(J)+J-I ) IF J>I AND (MD(J)+J-I)<MD(J+1)         HYS20870
C               =0              IF J>I AND (MD(J)+J-I)>MD(J+1)         HYS20880
C  -MATRIX P IS NOT SYMMETRIC, AND IS STORED AS A FULL MATRIX          HYS20890
C----------------------------------------------------------------------HYS20900
C   VARIABLE TABLE:                                                    HYS20910
C   IOPT  = OPTION NUMBER                                              HYS20920
C   N     = NUMBER OF DEGREES OF FREEDOM OF A MATRIX                   HYS20930
C   NLC   = NUMBER OF LOAD CASES FOR THE P MATRIX                      HYS20940
C   L1    = FIRST ROW TO BE WORKED WITH                                HYS20950
C   L2    = LAST ROW TO BE WORKED WITH                                 HYS20960
C   MD    = ADDRESS OF THE MAIN DIAGONAL TERMS OF A                    HYS20970
C   P     = LAOD MATRIX ON INPUT, SOLN (X) OF AX=P ON COMPLEATION OF   HYS20980
C           IOPT=4                                                     HYS20990
C   FACTOR= TEMPORARY STORAGE MATRIX                                   HYS21000
C   I     = ROW NUMBER FOR ELIMINATION                                 HYS21010
C   I     = ROW NUMBER                                                 HYS21020
C   J     = COLUMN NUMBER                                              HYS21030
C----------------------------------------------------------------------HYS21040
C                                                                      HYS21050
      L1=MAX(L1,1)                                                     HYS21060
      L2=MIN(L2,N)                                                     HYS21070
C                                                                      HYS21080
      GO TO (10,110,210,310,410),IOPT                                  HYS21090
C                                                                      HYS21100
   10 CONTINUE                                                         HYS21110
C---------- IOPT=1, REDUCE A AND P ------------------------------------HYS21120
C           GAUSSIAN ELIMINIATION IS USED TO REDUCE THE A AND P MATRICIESHYS21130
C           FROM ROW L1 TO ROW L2.  IF L1 IS NOT 1, IT IS ASSUMED THAT HYS21140
C           A AND P ARE ALLREADY REDUCED FROM 1 TO L1...               HYS21150
C----------------------------------------------------------------------HYS21160
      IF (PRINT) THEN                                                  HYS21170
        WRITE (6,500) 'IOPT=1, REDUCE STIFFNESS AND LOADS',L1,L2,N     HYS21180
        CALL LMATRX(A,MD,N,IMD ,'INPUT STIFFNESS')                     HYS21190
        CALL WMATRX(P,N ,NLC,'INPUT LOAD')                             HYS21200
      ENDIF                                                            HYS21210
C                                                                      HYS21220
      DO 100 L=L1,L2                                                   HYS21230
        IF (L.EQ.N) GO TO 100                                          HYS21240
C                                                                      HYS21250
C------- TEST FOR ZERO PIVOT.................                          HYS21260
        IF (A(MD(L)).EQ.0) THEN                                        HYS21270
          WRITE (6,2000) L,MD(L),A(MD(L))                              HYS21280
C         CALL ERRTRA                                                  HYS21290
C       RETURN '*** ABNORMAL END IN SUBROUTINE GAUSS **'               HYS21300
          ABORT=.TRUE.                                                 HYS21310
          RETURN                                                       HYS21320
        ENDIF                                                          HYS21330
C                                                                      HYS21340
C------- CALCULATE THE FACTORS EACH ROW IS MULTIPLIED BY               HYS21350
        DO 30 I=L+1,N                                                  HYS21360
          LI=IJI(L,I)                                                  HYS21370
          IF ( LI.LT.MD(I+1) ) THEN                                    HYS21380
               FACTOR(I)=A(LI) /A( MD(L) )                             HYS21390
          ELSE                                                         HYS21400
               FACTOR(I)=0                                             HYS21410
          ENDIF                                                        HYS21420
   30     CONTINUE                                                     HYS21430
C                                                                      HYS21440
C------- REDUCE A BY COLUMNS (J) EACH ROW (I)...                       HYS21450
        DO 40 J=(L+1),N                                                HYS21460
          LJ=IJI(L,J)                                                  HYS21470
          IF (LJ.GE.MD(J+1)) GO TO 40                                  HYS21480
          DO 35 I=(L+1),J                                              HYS21490
            IJ=IJI(I,J)                                                HYS21500
            IF (IJ.LT.MD(J+1)) A(IJ)=A(IJ)-FACTOR(I)*A(LJ)             HYS21510
C           A(IJ)=A(IJ)-FACTOR(I)*A(LJ)                                HYS21520
   35       CONTINUE                                                   HYS21530
   40   CONTINUE                                                       HYS21540
C                                                                      HYS21550
C------- REDUCE P BY COLUMNS (J) EACH ROW (I)...                       HYS21560
        DO 50 J=1,NLC                                                  HYS21570
          IF (P(L,J).EQ.0) GO TO 50                                    HYS21580
          DO 45 I=(L+1),N                                              HYS21590
   45     P(I,J)=P(I,J)-FACTOR(I)*P(L,J)                               HYS21600
   50   CONTINUE                                                       HYS21610
C                                                                      HYS21620
C                                                                      HYS21630
  100 CONTINUE                                                         HYS21640
      IF (PRINT) THEN                                                  HYS21650
        CALL LMATRX(A,MD,N,IMD,'REDUCED STIFFNESS')                    HYS21660
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')                           HYS21670
      ENDIF                                                            HYS21680
      RETURN                                                           HYS21690
C                                                                      HYS21700
  110 CONTINUE                                                         HYS21710
C---------- IOPT=2, REDUCE P ONLY -------------------------------------HYS21720
C           GAUSSIAN ELIMINIATION IS USED TO REDUCE THE P MATRIX       HYS21730
C           FROM ROW L1 TO ROW L2.  IF L1 IS NOT 1, IT IS ASSUMED THAT HYS21740
C           A AND P ARE ALLREADY REDUCED FROM 1 TO L1...               HYS21750
C----------------------------------------------------------------------HYS21760
      IF (PRINT) THEN                                                  HYS21770
        WRITE (6,500) 'IOPT=2, REDUCE LOADS ONLY',L1,L2,N              HYS21780
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                HYS21790
        CALL WMATRX(P,N ,NLC,'NEW LOAD MATRIX ')                       HYS21800
      ENDIF                                                            HYS21810
C                                                                      HYS21820
C------- LOOP FOR EACH ROW (L) TO REDUCE....                           HYS21830
      DO 160 L=L1,L2                                                   HYS21840
C                                                                      HYS21850
        IF (A(MD(L)).EQ.0) THEN                                        HYS21860
          WRITE (6,2000) L,MD(L),A(MD(L))                              HYS21870
C         CALL ERRTRA                                                  HYS21880
          ABORT=.TRUE.                                                 HYS21890
          RETURN                                                       HYS21900
        ENDIF                                                          HYS21910
C                                                                      HYS21920
C       LOOP FOR EACH COLUMN (J)                                       HYS21930
        DO 150 J=1,NLC                                                 HYS21940
          IF (P(L,J).EQ.0) GO TO 150                                   HYS21950
          DO 120 I=(L+1),N                                             HYS21960
            LI=IJI(L,I)                                                HYS21970
            IF (LI.LT.MD(I+1)) P(I,J)=P(I,J)-A(LI)*P(L,J)/A(MD(L))     HYS21980
  120       CONTINUE                                                   HYS21990
  150   CONTINUE                                                       HYS22000
  160   CONTINUE                                                       HYS22010
C     CALL WMATRX(P,N ,NLC,'REDUCED LOAD')                             HYS22020
      RETURN                                                           HYS22030
C                                                                      HYS22040
  210 CONTINUE                                                         HYS22050
C---------- IOPT=3, BACK SUBSTUTION -----------------------------------HYS22060
C           BACK SUBSTUTION IS USED TO SOLVE FOR X IN SX=V, WHERE S IS HYS22070
C           THE REDUCE UPPER TRIANGULAR FACTOR OF THE A MATRIX, AND V  HYS22080
C           IS THE REDUCED FORRM OF THE P MATRIX.  THE RESULTS (X) ARE HYS22090
C           STORED IN P.                                               HYS22100
C           BACK SUBSTUTION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.   HYS22110
C----------------------------------------------------------------------HYS22120
      IF (PRINT) THEN                                                  HYS22130
        WRITE (6,500) 'IOPT=3, BACK SUBSTUTION',L1,L2,N                HYS22140
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                HYS22150
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')                           HYS22160
      ENDIF                                                            HYS22170
C                                                                      HYS22180
C------ LOOP FOR EACH ROW L                                           HYS22190
C     DO 250 L=L2,L1,-1                          /*ORIGINAL          HYS22200
C                                                /*ORIGINAL          HYS22210
C       IF (A(MD(L)).LE.0) THEN                  /*ORIGINAL          HYS22220
C         WRITE (6,2000) L,MD(L),A(MD(L))        /*ORIGINAL          HYS22230
C         CALL ERRTRA                            /*ORIGINAL          HYS22240
C         ABORT=.TRUE.                           /*ORIGINAL          HYS22250
C         RETURN                                 /*ORIGINAL          HYS22260
C       ENDIF                                    /*ORIGINAL          HYS22270
C                                                                      HYS22280
C     CALCULAGE X FOR EACH LOAD CASE, STORE IN P                       HYS22290
      DO 250 J=L2,L1,-1                                                HYS22300
      IF (A(MD(J)).EQ.0) THEN                                          HYS22310
        WRITE (6,2000) J,MD(J),A(MD(J))                                HYS22320
        CALL ERRTRA                                                    HYS22330
        ABORT=.TRUE.                                                   HYS22340
        RETURN                                                         HYS22350
      ENDIF                                                            HYS22360
      DO 215 LC=1,NLC                                                  HYS22370
  215   P(J,LC)=P(J,LC)/A(MD(J))                                       HYS22380
      IF (J.GT.1) THEN                                                 HYS22390
       IJ=MD(J+1)                                                      HYS22400
       NI=J-(MD(J+1)-MD(J)) +1                                         HYS22410
C        WRITE (6,*) 'NI,IJ',NI,IJ                                     HYS22420
       DO 220 I=NI,J-1,+1                                              HYS22430
         IJ=IJ-1                                                       HYS22440
         DO 220 LC=1,NLC                                               HYS22450
C          WRITE (6,*) 'I:',I,'J:',J,'LC:',LC,'IJ:',IJ                 HYS22460
C          WRITE (6,*) 'P(I,LC):',P(I,LC)                              HYS22470
C          WRITE (6,*) 'A(IJ)   ',A(IJ)                                HYS22480
           P(I,LC)=P(I,LC)-P(J,LC)*A(IJ)                               HYS22490
C          WRITE (6,*) 'Q(I,LC):',P(I,LC)                              HYS22500
  220  CONTINUE                                                        HYS22510
       ENDIF                                                           HYS22520
  250  CONTINUE                                                        HYS22530
C                                                                      HYS22540
C       DO 215 J=1,NLC                   /*                           HYS22550
C 215   P(L,J)=P(L,J)/A(MD(L))           /*                           HYS22560
C                                        /*                           HYS22570
C       MODIFY THE REMAINDER OF V        /*                           HYS22580
C       DO 220 I=1,(L-1)                 /*                           HYS22590
C       DO 220 J=1,NLC                   /*                           HYS22600
C         IL=IJI(I,L)                    /*                           HYS22610
C         IF (IL.LT.MD(L+1)) P(I,J)=P(I,J)-A(IL)*P(L,J)     /*         HYS22620
C 220   CONTINUE                         /*                           HYS22630
C 250   CONTINUE                         /*                           HYS22640
      IF (PRINT) CALL WMATRX(P,N ,NLC,'DISPLACEMENT')                  HYS22650
C                                                                      HYS22660
      RETURN                                                           HYS22670
C                                                                      HYS22680
  310 CONTINUE                                                         HYS22690
C---------- IOPT=4, REDUCED MULTIPLICATION ----------------------------HYS22700
C           SOLVE FOR V IN S*D=P WHERE S HAS BEEN REDUCED BY GAUSSIAN   HYS22710
C           ELIMINATION.                                               HYS22720
C           THE RESULTING LOAD IS STORED IN P.                         HYS22730
C           THE DISPLACEMENTS ARE INPUT IN D.                          HYS22740
C           MULTIPLACATION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.    HYS22750
C----------------------------------------------------------------------HYS22760
      IF (PRINT) THEN                                                  HYS22770
        WRITE (6,500) 'IOPT=4, REDUCED MULTIPLACATION',L1,L2,N         HYS22780
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                HYS22790
        CALL WMATRX(FACTOR,N ,1,'DISPLACEMENT')                        HYS22800
      ENDIF                                                            HYS22810
C                                                                      HYS22820
C------ ONLY ONE LOAD CASE....                                        HYS22830
      LC=1                                                             HYS22840
C       ----- ZERO P ----                                             HYS22850
      DO 320 J=L1,L2                                                   HYS22860
        IF (A(MD(J)).EQ.0) THEN                                        HYS22870
          WRITE (6,2000) J,MD(J),A(MD(J))                              HYS22880
C         CALL ERRTRA                                                  HYS22890
          ABORT=.TRUE.                                                 HYS22900
          RETURN                                                       HYS22910
        ENDIF                                                          HYS22920
  320   P(J,LC)=0                                                      HYS22930
```

```
C                                                                       HYS30950
C------------ MULTIPLY S_RED * DISP = P                                 HYS30960
      DO 340 I=L1,L2,+1                                                 HYS30970
      DO 340 J=L1,L2,+1                                                 HYS30980
        IF( I .LE. J ) IJ=MD(J)+J-I                                     HYS30990
        IF( I .GT. J ) IJ=MD(I)+I-J                                     HYS31000
  340   P(I,LC)=P(I,LC)+A(IJ)*FACTOR(J)                                 HYS31010
      IF (PRINT) CALL WMATRX(P,N ,1,'REDUCED LOAD')                     HYS31020
C                                                                       HYS31030
  400   CONTINUE                                                        HYS31040
C                                                                       HYS31050
      IF (PRINT) CALL WMATRX(P,N ,1,'LOAD')                             HYS31060
C                                                                       HYS31070
      RETURN                                                            HYS31080
  410 CONTINUE                                                          HYS31090
C--------- IOPT=5, BACK SUBSTUTION ---------------------------------------HYS31100
C           SOLVE FOR D'IN S''D'=P'                                     HYS31110
C                                                                       HYS31120
C    L1--> |        |     |  |   |  |P'|                                HYS31130
C          |   S'   |     |D'|   |  |  |                                HYS31140
C    L2--> |--------|  *  |--|   =  |--|                                HYS31150
C          |        |  S  |D |   |  |P |                                HYS31160
C    L3--> |        |     |  |   |  |  |                                HYS31170
C          WHERE S IS THE CONDENSED MATRIX BY GAUSS ELEMINATION,         HYS31180
C          AND IT IS A NON_POSITIVE DEFINITIVE MATRIX                   HYS31190
C          D AND P ARE THE REDUCED DISPLACEMENT AND LOAD MATRIX         HYS31200
C          P' ARE INPUT IN ARRAY P                                      HYS31210
C          D ARE INPUT IN ARRARY FACTOR                                 HYS31220
C-------------------------------------------------------------------------HYS31230
      IF (PRINT) THEN                                                   HYS31240
         WRITE (6,500) 'IOPT=5, BACK SUBSTUTION',L1,L2,N                HYS31250
         CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                HYS31260
         CALL WMATRX(FACTOR,N ,1,'INPUT DISPLACEMENT')                  HYS31270
         CALL WMATRX(P,N ,NLC,'INPUT LOAD')                             HYS31280
      ENDIF                                                             HYS31290
C                                                                       HYS31300
C------ ONLY ONE LOAD CASE....                                          HYS31310
      LC=1                                                              HYS31320
C                                                                       HYS31330
      DO 720 J=L1,L2                                                    HYS31340
        IF (A(MD(J)).EQ.0) THEN                                         HYS31350
          WRITE (6,2000) J,MD(J),A(MD(J))                               HYS31360
          CALL ERRTRA                                                   HYS31370
          ABORT=.TRUE.                                                  HYS31380
          RETURN                                                        HYS31390
        ENDIF                                                           HYS31400
  720   CONTINUE                                                        HYS31410
C                                                                       HYS31420
C-------CALCULATE DISPLACEMENT MATRIX , BACK SUBSTUTION ONLY           HYS31430
C       TAKES PLACE BETWEEN L1 AND L2                                   HYS31440
      L3=N                                                              HYS31450
      DO 640 I=L1,L2                                                    HYS31460
  640   FACTOR(I)=0.                                                    HYS31470
      DO 350 I=L2,L1,-1                                                 HYS31480
        DO 350 J=I,L3,+1                                                HYS31490
          IJ=MD(J)+J-I                                                  HYS31500
          IF( IJ .EQ. MD(I) ) THEN                                      HYS31510
            FACTOR(I)=FACTOR(I)+P(I,LC)/A(MD(I))                        HYS31520
          ELSE IF( IJ .LT. MD(J+1) ) THEN                               HYS31530
            FACTOR(I)=FACTOR(I)- A(IJ)*FACTOR(J)/A(MD(I))               HYS31540
          ENDIF                                                         HYS31550
  350   CONTINUE                                                        HYS31560
        IF(PRINT) THEN                                                  HYS31570
                  CALL WMATRX(P,N,LC,'OUTPUT LOAD')                     HYS31580
                  WRITE(6,*)'PRINTING MATRIX: OUTPUT DISPLACEMENT'      HYS31590
                  WRITE(6,79)(J,FACTOR(J),J=1,N)                        HYS31600
   79             FORMAT(1X,'ROW',I5,3X,F15.6)                          HYS31610
        ENDIF                                                           HYS31620
      RETURN                                                            HYS31630
  500 FORMAT (/2X,A,2X,'L1=',I5,'   L2=',I5,'   N=',I5)                 HYS31640
 2000     FORMAT (///1X,50('*')/                                        HYS31650
     &            ' *  ER','ROR IN SUBROUTINE GAUSS',          T51,'*'/ HYS31660
     &            ' *  PIVOT IS LE 0 -------- ROW=',I5,        T51,'*'/ HYS31670
     &            ' *                STORAGE LOC=',I5,         T51,'*'/ HYS31680
     &            ' *                      PIVOT=',1P,G15.6,T51,'*'/    HYS31690
     &            ' *  SOLUTION IS ABORTED ',                   T51,'*'/ HYS31700
     &            1X,50('*')/)                                          HYS31710
      END                                                               HYS31720
C                                                                       HYS31730
C.PA                                                                    HYS31740
      SUBROUTINE SKY ( ASAT,STIFF,IDF,ISTOR,NOP,MDIAG,IMD  )            HYS31750
C=======================================================================HYS31760
C==   CONVERT STIFFNESS MATRIX TO HALF STORAGE AND BANDED MODE      ==  HYS31770
C=======================================================================HYS31780
C     DIMENSION ASAT(IDF,IDF),STIFF(ISTOR),MDIAG(IDF+1)                 HYS31790
      DIMENSION ASAT(IDF,IDF),STIFF(ISTOR),MDIAG( 199 )                 HYS31800
C                                                                       HYS31810
C CONVERT STIFFNESS MATRIX TO HALF STORAGE BANDED MODE - STIFF          HYS31820
C THE MAIN DIAGONAL ADDRESS ARE STORED IN MDIAG ARRAY  - MDIAG          HYS31830
C                                                                       HYS31840
      L=0                                                               HYS31850
      DO 70 J=1,NOP                                                     HYS31860
      DO 80 I=1,J                                                       HYS31870
C       WRITE(6,4)'I=',I,'J=',J                                         HYS31880
        IF( ASAT(I,J) .EQ. 0 .AND. I .NE. J )GO TO 80                   HYS31890
        IF( ASAT(I,J) .EQ. 0 .AND. I .EQ. J ) THEN                      HYS31900
          WRITE(6,*)'THE STIFFNESS ',I,'TH DIAGONAL TERM ARE ZERO'      HYS31910
     &              //' - SOLUTION ABORTED'                            HYS31920
          RETURN                                                        HYS31930
        ENDIF                                                           HYS31940
        M=I                                                             HYS31950
        GO TO 100                                                       HYS31960
   80 CONTINUE                                                          HYS31970
                                                                        HYS31980
  100 DO 70 I=J,M,-1                                                    HYS31990
            L=L+1                                                       HYS32000
            IF( I.EQ. J) MDIAG(J)=L                                     HYS32010
            IF( ASAT(I,J) .EQ. 0 .AND. I .EQ. J ) THEN                  HYS32020
              WRITE(6,*)'THE STIFFNESS ',I,'TH DIAGONAL TERM ARE ZERO'  HYS32030
     &              //' - SOLUTION ABORTED'                            HYS32040
              RETURN                                                    HYS32050
            ENDIF                                                       HYS32060
            STIFF(L)=ASAT(I,J)                                          HYS32070
            IPT=L                                                       HYS32080
   70 CONTINUE                                                          HYS32090
      MDIAG(NOP+1)=IPT+1                                                HYS32100
      IMD=MDIAG(NOP+1)-1                                                HYS32110
C                                                                       HYS32120
      RETURN                                                            HYS32130
      END                                                               HYS32140
C                                                                       HYS32150
C.pa                                                                    HYS32160
C     DEBUG UNIT(6),SUBCHK                                              HYS32170
C     END DEBUG                                                         HYS32180
      SUBROUTINE SNORM(IOPTT,SR,BR,DSR,DBR,TK,DELTP,DEN,NOP,IFM,        HYS32190
     &                 MM,LL,JJ,NN)                                     HYS32200
C=======================================================================HYS32210
C CALCULATE NORM OF LOAD INCREMENT AND NORM OF DISPLACEMENT INCREMENT   HYS32220
C                                                                       HYS32230
C IOPT : 1 - CALCULATE DELTP FOR NORM OF LOAD AND DEN                   HYS32240
C IOPT : 2 - CALCULATE DELTP FOR NORM OF DISPLACEMENT                   HYS32250
C                                                                       HYS32260
C SR   : DISPLACEMENT VECTOR                                           HYS32270
C BR   : LOAD VECTOR                                                    HYS32280
C DSR  : DISPLACEMENT INCREMENT VECTOR                                  HYS32290
C DBR  : LOAD VECTOR  INCREMENT VECTOR                                  HYS32300
C TK   : STIFFNESS MATRIX                                               HYS32310
C DELTP: NORM OF LOAD OR DISPLACEMENT                                   HYS32320
C DEN  : DEN = TRANSPOSE OF |DSR|*|TK|*|DSR|                           HYS32330
C IFM  : TRANVERSE AND ROTATION DOF IDENTIFICATION INDEX                HYS32340
C        0 FOR TRANVERSE                                                HYS32350
C        1 FOR ROTATION                                                 HYS32360
C                                                                       HYS32370
C NOTE: SR(MM,JJ), BR(MM,JJ), DSR(MM,LL), DBR(MM,LL), TK(MM,MM)        HYS32380
C       IFM(NN)                                                         HYS32390
C       DSRT AND DSRTK ARRAY ARE BASED ON 32 SEGMENTS. IF SEGMENT'S     HYS32400
C       NUMBER IS LARGER THAN 32, THE DIMENSION OF DSRT AND DSRTK       HYS32410
C       NEED TO MODIFIED                                                HYS32420
C=======================================================================HYS32430
C                                                                       HYS32440
      DIMENSION DSR(MM,LL),DBR(MM,LL),SR(MM,JJ),BR(MM,JJ),TK(MM,MM),    HYS32450
     &          DSRT(1,198),DSRTK(1,198),IFM(NN)                        HYS32460
C                                                                       HYS32470
C SKIM STIFFNESS TERMS                                                  HYS32480
C                                                                       HYS32490
      DO 80 I=1,NOP                                                     HYS32500
      DO 80 J=1,NOP                                                     HYS32510
        IF( ABS(TK(I,J)) .LT. 1.E-7 ) TK(I,J)=0                         HYS32520
   80 CONTINUE                                                          HYS32530
C                                                                       HYS32540
      GO TO (100,200),IOPTT                                             HYS32550
C----------------  IOPTT=1  ----------------                            HYS32560
C                                                                       HYS32570
C SEARCH MAXIMUM FORCE OR DISPLACEMENT INCREMENT                        HYS32580
C                                                                       HYS32590
  100 CONTINUE                                                          HYS32600
      FMAX=0.                                                           HYS32610
      TMMAX=0.                                                          HYS32620
      DO 25 I=1,NOP                                                     HYS32630
         IF(IFM(I) .EQ. 0 ) THEN                                        HYS32640
            IF(ABS(BR(I,1)) .GT. ABS(FMAX) ) THEN                       HYS32650
               FMAX=BR(I,1)                                             HYS32660
            ENDIF                                                       HYS32670
         ELSE IF(IFM(I) .NE. 0 ) THEN                                   HYS32680
            IF(ABS(BR(I,1)) .GT. ABS(TMMAX) ) THEN                      HYS32690
               TMMAX=BR(I,1)                                            HYS32700
            ENDIF                                                       HYS32710
         ENDIF                                                          HYS32720
   25 CONTINUE                                                          HYS32730
C                                                                       HYS32740
C FIND NORM OF DBR                                                      HYS32750
C                                                                       HYS32760
      DELT=0.                                                           HYS32770
      DO 32 I=1,NOP                                                     HYS32780
       IF(IFM(I) .EQ. 0 ) THEN                                          HYS32790
         IF(FMAX .EQ. 0. ) GO TO 32                                     HYS32800
         DELT=DELT+(ABS(DBR(I,1)/FMAX))**2                              HYS32810
       ELSE IF(IFM(I) .NE. 0 ) THEN                                     HYS32820
         IF(TMMAX .EQ. 0. ) GO TO 32                                    HYS32830
         DELT=DELT+(ABS(DBR(I,1)/TMMAX))**2                             HYS32840
       ENDIF                                                            HYS32850
   32 CONTINUE                                                          HYS32860
      DELTP=SQRT(DELT/NOP)                                              HYS32870
C                                                                       HYS32880
C FIND |DSR|**T*|TK|*|DSR|                                              HYS32890
C                                                                       HYS32900
      DO 67 I=1,NOP                                                     HYS32910
         DSRT(1,I)=DSR(I,1)                                             HYS32920
   67 CONTINUE                                                          HYS32930
C                                                                       HYS32940
C |DSRT|**(T)*|TK|=|DSRTK|                                              HYS32950
C                                                                       HYS32960
      DO 68 J=1,NOP                                                     HYS32970
         DSRTK(1,J)=0.                                                  HYS32980
         DO 68 K=1,NOP                                                  HYS32990
            DSRTK(1,J)=DSRTK(1,J)+DSRT(1,K)*TK(K,J)                     HYS33000
   68 CONTINUE                                                          HYS33010
C                                                                       HYS33020
C |DSRTK|*|DSR|=|DSRT|                                                  HYS33030
C                                                                       HYS33040
         DSRT(1,1)=0.                                                   HYS33050
      DO 78 K=1,NOP                                                     HYS33060
         DSRT(1,1)=DSRT(1,1)+DSRTK(1,K)*DSR(K,1)                        HYS33070
   78 CONTINUE                                                          HYS33080
      DEN=DSRT(1,1)                                                     HYS33090
C                                                                       HYS33100
      RETURN                                                            HYS33110
C                                                                       HYS33120
C----------------  IOPTT=2  ----------------                            HYS33130
  200 CONTINUE                                                          HYS33140
      FMAX=0.                                                           HYS33150
      TMMAX=0.                                                          HYS33160
      DO 45 I=1,NOP                                                     HYS33170
         IF(IFM(I) .EQ. 0 ) THEN                                        HYS33180
            IF(ABS(SR(I,1)) .GT. ABS(FMAX) ) THEN                       HYS33190
               FMAX=SR(I,1)                                             HYS33200
            ENDIF                                                       HYS33210
         ELSE IF(IFM(I) .NE. 0 ) THEN                                   HYS33220
            IF(ABS(SR(I,1)) .GT. ABS(TMMAX) ) THEN                      HYS33230
               TMMAX=SR(I,1)                                            HYS33240
            ENDIF                                                       HYS33250
         ENDIF                                                          HYS33260
   45 CONTINUE                                                          HYS33270
C                                                                       HYS33280
C FIND NORM OF DSR                                                      HYS33290
C                                                                       HYS33300
      DELT=0.                                                           HYS33310
      DO 42 I=1,NOP                                                     HYS33320
       IF(IFM(I) .EQ. 0 ) THEN                                          HYS33330
         IF(FMAX .EQ. 0. ) GO TO 42                                     HYS33340
         DELT=DELT+(ABS(DSR(I,1)/FMAX))**2                              HYS33350
       ELSE IF(IFM(I) .NE. 0 ) THEN                                     HYS33360
         IF(TMMAX .EQ. 0. ) GO TO 42                                    HYS33370
         DELT=DELT+(ABS(DSR(I,1)/TMMAX))**2                             HYS33380
       ENDIF                                                            HYS33390
   42 CONTINUE                                                          HYS33400
      DELTP=SQRT(DELT/NOP)                                              HYS33410
      RETURN                                                            HYS33420
      END                                                               HYS33430
C                                                                       HYS33440
C.PA                                                                    HYS33450
C@PROCESS SDUMP GOSTMT XREF MAP                                         HYS33460
C=======================================================================HYS33470
      SUBROUTINE BOXLOC(LIBN,FJL,STR,NSEG,NUMP,ISEG,IINEB,IINEH,       HYS33480
     &                  TE,ST,EM,UO,VO,FLP )                            HYS33490
C     ------------------------------------------                        HYS33500
C     JUDGE LOCAL BUCKLING STRESS FOR BOX SECTION                       HYS33510
C     ------------------------------------------                        HYS33520
C     NOTE: FLP=0    : NO LOCAL BUCKLING OCCURRED                       HYS33530
C           FLP=1-5  : LOCAL BUCKLING OCCURRED                         HYS33540
C                                                                       HYS33550
      DIMENSION STR(NSEG,NUMP),UO(NUMP),VO(NUMP)                        HYS33560
      DIMENSION FJL(NUMP),TE(NUMP)                                      HYS33570
      REAL IINEB, IINEH                                                 HYS33580
      INEB=IINEB                                                        HYS33590
      INEH=IINEH                                                        HYS33600
C                                                                       HYS33610
      IF( LIBN .NE. 1 ) RETURN                                          HYS33620
C                                                                       HYS33630
      DO 60 I=1,NUMP                                                    HYS33640
   60   FJL(I)=0                                                        HYS33650
        IPT1=1                                                          HYS33660
        INEQ=INEB+INEH                                                  HYS33670
        IPT2=INEQ+1                                                     HYS33680
        IPT3=2*INEQ+1                                                   HYS33690
        IPT4=3*INEQ+1                                                   HYS33700
C                                                                       HYS33710
C     CALCULATE SECTION ELEMENT LENGTH                                  HYS33720
C                                                                       HYS33730
        ELH=ABS(VO(2)-VO(3))                                            HYS33740
                                                                        HYS33750
                                                                        HYS33760
                                                                        HYS33770
                                                                        HYS33780
```

```
      ELB=ABS(U0(INEH+1)-U0(INEH+2))
C     ***********************************
C     ** JUDGE LOCAL BUCKLING INDEX FJL **
C     ***********************************
C
C     AVOID SECTION REACH TO FULLY YIELD COMPRESSION STRESS
C
      IF( STR(ISEG,IPT1) .LT. 0 .AND. STR(ISEG,IPT2) .LT. 0 .AND.
     &    STR(ISEG,IPT3) .LT. 0 .AND. STR(ISEG,IPT4) .LT. 0 ) THEN
            IF( TE(IPT1) .NE. EM .AND. TE(IPT2) .NE. EM .AND.
     &          TE(IPT3) .NE. EM .AND. TE(IPT4) .NE. EM       ) THEN
              DO 260 I=1,NUMP
  260           FJL(I)=1
                FLP=5
               RETURN
            ENDIF
      ENDIF
C
      IF( STR(ISEG,IPT1) .LT. 0 .AND. STR(ISEG,IPT3) .LT. 0 ) THEN
        IF(TE(IPT1) .NE. EM .AND. TE(IPT3) .NE. EM) THEN
          DO 70 I=INEH+1,INEQ
   70       FJL(I)=1
          DO 72 I=2*INEQ+INEH+1, 3*INEQ
   72       FJL(I)=1
          W=0
          TOTST=0
          J=0
          DO 74 I=2,INEH
              IF(STR(ISEG,I) .LT. 0 ) THEN
                 J=J+1
                 W=W+ELH
                 TOTST=TOTST+STR(ISEG,I)
              ENDIF
   74     CONTINUE
          DO 76 I=INEQ+2,INEQ+INEH
              IF(STR(ISEG,I) .LT. 0 ) THEN
                 J=J+1
                 W=W+ELH
                 TOTST=TOTST+STR(ISEG,I)
              ENDIF
   76     CONTINUE
C
C     FIND AVERAGE STRESS AND CRITICAL STRESS SCR
C
          IF( J .EQ. 0 ) THEN
              SAVE=0
              SCR=999
          ELSE IF( J .NE. 0 ) THEN
              SAVE=ABS(TOTST/J)
              SCR=0.425*(3.14159*3.14159)*EM/(12*0.91*(W/ST)**2)
          ENDIF
C
          IF( SAVE .GT. SCR) THEN
            DO 78 I=2,INEH
   78         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1
            DO 80 I=2,INEH
   80         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1
            FLP=1
          ENDIF
C
          W=0
          TOTST=0
          J=0
          DO 84 I=2*INEQ+2,2*INEQ+INEH
            IF( STR(ISEG,I) .LT. 0 ) THEN
               J=J+1
               W=W+ELH
               TOTST=TOTST+STR(ISEG,I)
            ENDIF
   84     CONTINUE
          DO 86 I=3*INEQ+2,3*INEQ+INEH
            IF( STR(ISEG,I) .LT. 0 ) THEN
               J=J+1
               W=W+ELH
               TOTST=TOTST+STR(ISEG,I)
            ENDIF
   86     CONTINUE
C
C     FIND AVERAGE STRESS AND CRITICAL STRESS SCR
C
          IF( J .EQ. 0 ) THEN
              SAVE=0
              SCR=999
          ELSE IF( J .NE. 0 ) THEN
              SAVE=ABS(TOTST/J)
              SCR=0.425*(3.14159*3.14159)*EM/(12*0.91*(W/ST)**2)
          ENDIF
C
          IF( SAVE .GT. SCR) THEN
            DO 88 I=2*INEQ+2,2*INEQ+INEH
   88         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1
            DO 90 I=3*INEQ+2,3*INEQ+INEH
   90         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1
            FLP=1
          ENDIF
        ENDIF
      ENDIF
C
      IF( STR(ISEG,IPT2) .LT. 0 .AND. STR(ISEG,IPT4) .LT. 0 ) THEN
        IF(TE(IPT2) .NE. EM .AND. TE(IPT4) .NE. EM) THEN
          DO 40 I=INEQ+INEH+1,2*INEQ
   40       FJL(I)=1
          DO 42 I=3*INEQ+INEH+1, 4*INEQ
   42       FJL(I)=1
          W=0
          TOTST=0
          J=0
          DO 44 I=2,INEH
              IF(STR(ISEG,I) .LT. 0 ) THEN
                 J=J+1
                 W=W+ELH
                 TOTST=TOTST+STR(ISEG,I)
              ENDIF
   44     CONTINUE
          DO 46 I=INEQ+2,INEQ+INEH
              IF(STR(ISEG,I) .LT. 0 ) THEN
                 J=J+1
                 W=W+ELH
                 TOTST=TOTST+STR(ISEG,I)
              ENDIF
   46     CONTINUE
C
C     FIND AVERAGE STRESS AND CRITICAL STRESS SCR
C
          IF( J .EQ. 0 ) THEN
              SAVE=0
              SCR=999
          ELSE IF( J .NE. 0 ) THEN
              SAVE=ABS(TOTST/J)
              SCR=0.425*(3.14159*3.14159)*EM/(12*0.91*(W/ST)**2)
          ENDIF
C
          IF( SAVE .GT. SCR) THEN
            DO 48 I=2,INEH
   48         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1
            DO 50 I=INEQ+2,INEQ+INEH
   50         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1
            FLP=2
          ENDIF
C
          W=0
          TOTST=0
          J=0
          DO 54 I=2*INEQ+2,2*INEQ+INEH
            IF( STR(ISEG,I) .LT. 0 ) THEN
               J=J+1
               W=W+ELH
               TOTST=TOTST+STR(ISEG,I)
            ENDIF
   54     CONTINUE
          DO 56 I=3*INEQ+2,3*INEQ+INEH
            IF( STR(ISEG,I) .LT. 0 ) THEN
               J=J+1
               W=W+ELH
               TOTST=TOTST+STR(ISEG,I)
            ENDIF
   56     CONTINUE
C
C     FIND AVERAGE STRESS AND CRITICAL STRESS SCR
C
          IF( J .EQ. 0 ) THEN
              SAVE=0
              SCR=999
          ELSE IF( J .NE. 0 ) THEN
              SAVE=ABS(TOTST/J)
              SCR=0.425*(3.14159*3.14159)*EM/(12*0.91*(W/ST)**2)
          ENDIF
C
          IF( SAVE .GT. SCR) THEN
            DO 58 I=2*INEQ+2,2*INEQ+INEH
   58         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1
            DO 61 I=3*INEQ+2,3*INEQ+INEH
   61         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1
            FLP=2
          ENDIF
        ENDIF
      ENDIF
C
      IF( STR(ISEG,IPT1) .LT. 0 .AND. STR(ISEG,IPT2) .LT. 0 ) THEN
        IF(TE(IPT1) .NE. EM .AND. TE(IPT2) .NE. EM) THEN
          DO 20 I=1,INEH
   20       FJL(I)=1
          DO 22 I=INEQ+1, INEQ+INEH
   22       FJL(I)=1
          W=0
          TOTST=0
          J=0
          DO 24 I=INEQ+INEH+1,2*INEQ
              IF(STR(ISEG,I) .LT. 0 ) THEN
                 J=J+1
                 W=W+ELB
                 TOTST=TOTST+STR(ISEG,I)
              ENDIF
   24     CONTINUE
          DO 26 I=3*INEQ+INEH+1,4*INEQ
              IF(STR(ISEG,I) .LT. 0 ) THEN
                 J=J+1
                 W=W+ELB
                 TOTST=TOTST+STR(ISEG,I)
              ENDIF
   26     CONTINUE
C
C     FIND AVERAGE STRESS AND CRITICAL STRESS SCR
C
          IF( J .EQ. 0 ) THEN
              SAVE=0
              SCR=999
          ELSE IF( J .NE. 0 ) THEN
              SAVE=ABS(TOTST/J)
              SCR=0.425*(3.14159*3.14159)*EM/(12*0.91*(W/ST)**2)
          ENDIF
C
          IF( SAVE .GT. SCR) THEN
            DO 28 I=INEQ+INEH+1,2*INEQ
   28         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1
            DO 120 I=3*INEQ+INEH+1,4*INEQ
  120         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1
            FLP=3
          ENDIF
C
          W=0
          TOTST=0
          J=0
          DO 154 I=INEH+1,INEQ
            IF( STR(ISEG,I) .LT. 0 ) THEN
               J=J+1
               W=W+ELB
               TOTST=TOTST+STR(ISEG,I)
            ENDIF
  154     CONTINUE
          DO 156 I=2*INEQ+INEH+1,3*INEQ
            IF( STR(ISEG,I) .LT. 0 ) THEN
               J=J+1
               W=W+ELB
               TOTST=TOTST+STR(ISEG,I)
            ENDIF
  156     CONTINUE
C
C     FIND AVERAGE STRESS AND CRITICAL STRESS SCR
C
          IF( J .EQ. 0 ) THEN
              SAVE=0
              SCR=999
          ELSE IF( J .NE. 0 ) THEN
              SAVE=ABS(TOTST/J)
              SCR=0.425*(3.14159*3.14159)*EM/(12*0.91*(W/ST)**2)
          ENDIF
C
          IF( SAVE .GT. SCR) THEN
            DO 158 I=INEH+1,INEQ
  158         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1
            DO 161 I=2*INEQ+INEH+1,3*INEQ
  161         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1
            FLP=3
          ENDIF
        ENDIF
      ENDIF
C
      IF( STR(ISEG,IPT3) .LT. 0 .AND. STR(ISEG,IPT4) .LT. 0 ) THEN
        IF(TE(IPT3) .NE. EM .AND. TE(IPT4) .NE. EM) THEN
          DO 30 I=2*INEQ+1,2*INEQ+INEH
   30       FJL(I)=1
          DO 32 I=3*INEQ+1, 3*INEQ+INEH
   32       FJL(I)=1
          W=0
          TOTST=0
          J=0
          DO 34 I=2*INEQ+INEH+1,3*INEQ
              IF(STR(ISEG,I) .LT. 0 ) THEN
                 J=J+1
                 W=W+ELB
                 TOTST=TOTST+STR(ISEG,I)
              ENDIF
   34     CONTINUE
          DO 36 I=INEH+1,INEQ
              IF(STR(ISEG,I) .LT. 0 ) THEN
                 J=J+1
                 W=W+ELB
                 TOTST=TOTST+STR(ISEG,I)
              ENDIF
   36     CONTINUE
C
C     FIND AVERAGE STRESS AND CRITICAL STRESS SCR
C
```

```
          IF( J .EQ. 0 ) THEN                                                   HYS36630
              SAVE=0                                                            HYS36640
              SCR=999                                                           HYS36650
          ELSE IF( J .NE. 0 ) THEN                                              HYS36660
              SAVE=ABS(TOTST/J)                                                 HYS36670
              SCR=0.425*(3.14159+3.14159)*EM/(12*0.91*(W/ST)**2)                HYS36680
          ENDIF                                                                 HYS36690
C                                                                               HYS36700
          IF( SAVE .GT. SCR) THEN                                               HYS36710
            DO 30 I=INEH+1,INEQ                                                 HYS36720
   30         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1            HYS36730
            DO 130 I=2*INEQ+INEH+1,3*INEQ                                       HYS36740
  130         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1            HYS36750
            FLP=4                                                               HYS36760
          ENDIF                                                                 HYS36770
C                                                                               HYS36780
          W=0                                                                   HYS36790
          TOTST=0                                                               HYS36800
          J=0                                                                   HYS36810
          DO 14 I=3*INEQ+INEH+1,4*INEQ                                          HYS36820
            IF( STR(ISEG,I) .LT. 0 ) THEN                                       HYS36830
              J=J+1                                                             HYS36840
              W=W+ELB                                                           HYS36850
              TOTST=TOTST+STR(ISEG,I)                                           HYS36860
            ENDIF                                                               HYS36870
   14     CONTINUE                                                             HYS36880
          DO 16 I=INEQ+INEH+1,2*INEQ                                            HYS36890
            IF( STR(ISEG,I) .LT. 0 ) THEN                                       HYS36900
              J=J+1                                                             HYS36910
              W=W+ELB                                                           HYS36920
              TOTST=TOTST+STR(ISEG,I)                                           HYS36930
            ENDIF                                                               HYS36940
   16     CONTINUE                                                             HYS36950
C                                                                               HYS36960
C     FIND AVERAGE STRESS AND CRITICAL STRESS SCR                               HYS36970
C                                                                               HYS36980
          IF( J .EQ. 0 ) THEN                                                   HYS36990
              SAVE=0                                                            HYS37000
              SCR=999                                                           HYS37010
          ELSE IF( J .NE. 0 ) THEN                                              HYS37020
              SAVE=ABS(TOTST/J)                                                 HYS37030
              SCR=0.425*(3.14159+3.14159)*EM/(12*0.91*(W/ST)**2)                HYS37040
          ENDIF                                                                 HYS37050
C                                                                               HYS37060
          IF( SAVE .GT. SCR) THEN                                               HYS37070
            DO 18 I=INEQ+INEH+1,2*INEQ                                          HYS37080
   18         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1            HYS37090
            DO 11 I=3*INEQ+INEH+1,4*INEQ                                        HYS37100
   11         IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1            HYS37110
            FLP=4                                                               HYS37120
          ENDIF                                                                 HYS37130
        ENDIF                                                                   HYS37140
      ENDIF                                                                     HYS37150
C                                                                               HYS37160
      RETURN                                                                    HYS37170
      END                                                                       HYS37180
C@PROCESS SDUMP GOSTMT XREF                                                     HYS00010
C********** HYST13 ************************************************************HYS00020
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                            HYS00030
C           END DEBUG                                                           HYS00040
C                                                                               HYS00050
C     GOEL TRUSS HYSTERESIS MODEL FOR SIMULATED LOCAL BUCKLING                  HYS00060
C     OF BOX SECTION                                                            HYS00070
C     NOTE: IT IS ASSUMED THAT OVERALL BUCKLING WILL NOT OCCURRED               HYS00080
C                                                                               HYS00090
C                                                                               HYS00100
      SUBROUTINE HYST13(P,PLA,DE,DELA,V,VLA,EL,SLK,                             HYS00110
     &                  EM,AREA,RADG,YS,PT,                                     HYS00120
     &                  CC,CYCLE,RULE,STIF,IREV,IDUC,                           HYS00130
     &                  HA,VA,HC,VC,RSN,DEMAX,REGIN,PREGIN,                     HYS00140
     &                  PBOT,DBOT,PTOP,DTOP,PRINT,                              HYS00150
     &                  EKI, DUC, EXCR, DELT, ESE, PSE, PSEOLD, CSE)            HYS00160
      DIMENSION DUC(3), EXCR(6), ESE(3)                                         HYS00170
      LOGICAL REVRSC,PRINT                                                      HYS00180
      REAL*4 IREV, IDUC                                                         HYS00190
      IF( IREV .EQ. 0.) REVRSC=.FALSE.                                          HYS00200
      IF( IREV .EQ. 1.) REVRSC=.TRUE.                                           HYS00210
      PI=3.1415926                                                              HYS00220
      ES=AREA*EM/EL                                                             HYS00230
      DELT=PT/ES                                                                HYS00240
      DEL=DE/DELT                                                               HYS00250
      PL=P/PT                                                                   HYS00260
      ESR=SLK*EL/RADG                                                           HYS00270
      SLOP7=STIF/ES                                                             HYS00280
      DA=ABS(DE)                                                                HYS00290
C                                                                               HYS00300
      DP=P-PLA                                                                  HYS00310
      DDE=DE-DELA                                                               HYS00320
      DDEL=DDE/DELT                                                             HYS00330
C ==== CALCULATE EXCURSION RATE AT EVERY ZERO CROSSING ========================HYS00340
C                                                                               HYS00350
      IF( P*PLA .LE. 0 ) THEN                                                   HYS00360
        CALL DNE(0, DUC,EXCR,DELT, DA,ESE,PSE,PSEOLD,CSE)                       HYS00370
      ENDIF                                                                     HYS00380
C                                                                               HYS00390
C*****************************************************************            HYS00400
C     DUC     ---- ELEMENT DUCTILITY RATE                                       HYS00410
C     EXCR    ---- ELEMENT EXCURSION RATE                                       HYS00420
C     DELT    ---- ELEMENT YIELDING DISPLACEMENT                                HYS00430
C     ESE     ---- ELEMENT ELASTIC STRAIN ENERGY                                HYS00440
C     PSE     ---- ELEMENT PLASTIC STRAIN ENERGY                                HYS00450
C     PSEOLD  ---- ELEMENT PLASTIC STRAIN ENERGY AT LAST ZERO CROSSING          HYS00460
C     CSE     ---- ELEMENT CONSTANT STRAIN ENERGY                               HYS00470
C                                                                               HYS00480
C     DP      ---- ELEMENT AXIAL LOAD INCREMENT (KIPS)                          HYS00490
C     ESR     ---- EFFECTIVE SLENDER RATIO  (KL/R)                              HYS00500
C     YS      ---- YIELDING STRESS                                              HYS00510
C     AREA    ---- CROSS SECTION AREA (A)                                       HYS00520
C     EM      ---- ELASTIC MODULUS  (E)                                         HYS00530
C     DP      ---- ELEMENT AXIAL LOAD INCREMENT (KIPS)                          HYS00540
C     P       ---- CURRENT AXIAL LOAD (KIPS)                                    HYS00550
C     PL      ---- P - PY RATIO (P/PY)                                          HYS00560
C     PLA     ---- LAST AXIAL LOAD                                              HYS00570
C     DDEL    ---- DISPLACEMENT INCREMENT  (IN)                                 HYS00580
C     DE      ---- CURRENT AXIAL DISPLACEMENT  (IN)                             HYS00590
C     DEMAX   ---- MAXIMUM AXIAL DISPLACEMENT  (IN)                             HYS00600
C     DA      ---- CURRENT ABSOLUTE AXIAL DISPLACEMENT  (IN)                    HYS00610
C     DELA    ---- LAST AXIAL DISPLACEMENT                                      HYS00620
C     DDE     ---- ELEMENT DISPLACEMENT INCREMENT                               HYS00630
C     DEL     ---- DEL=DE/DELT                                                  HYS00640
C     DELT    ---- YIELDING DISPLACEMENT                                        HYS00650
C     V       ---- CURRENT VELOCITY                                             HYS00660
C     VLA     ---- LAST VELOCITY                                                HYS00670
C     ES      ---- ELEMENT ELASTIC STIFFNESS                                    HYS00680
C     EKI     ---- UNLOADING EQUIVALENT STIFFNESS FOR CALCULATING EESE          HYS00690
C     PT      ---- TENSION YIELDING LOAD                                        HYS00700
C     EL      ---- ELEMENT LENGTH  (IN)                                         HYS00710
C     CC      ---- C SUB. C CONSTANT                                            HYS00720
C     PMAX    ---- BUCKING LOAD (KIPS)                                          HYS00730
C     DELMAX  ---- BUCKING DISPLACEMENT (IN)                                    HYS00740
C     HA,HB,..,HC1,HD1,..,HF ---- P/PY - DE/DELY PLOT'S DE/DELY COORDINATE      HYS00750
C     VA,VB,..,VC1,VD1,..,VF ---- P/PY - DE/DELY PLOT'S P/PY COORDINATE         HYS00760
C     RSN     ---- RESIDUAL STRAIN                                              HYS00770
C     RULE  ---- RULE NUMBER                                                    HYS00780
C     CYCLE   ---- NO. OF CYCLE                                                 HYS00790
C     STIF    ---- ELEMENT STIFFNESS                                            HYS00800
C     REVRSC ---- LOGICAL INDEX FOR REVERSING CURVE OCCURRED                    HYS00810
C     IDUC    ---- 1: HC LESS THAN 12,  0:HC=12                                 HYS00820
C     REGIN  ---- REGIN 1: DE >  HA                                             HYS00830
C                      2: DE = (HH, HA)                                         HYS00840
C                      3: DE = (HB, HH)                                         HYS00850
C                      4: DE = (HC, HB)                                         HYS00860
C   PREGIN ---- PREVIOUS STEP REGIN ID NUMBER                                   HYS00870
C*****************************************************************            HYS00880
C                                                                               HYS00890
C INITIAL BUCKLING LOAD                                                         HYS00900
C                                                                               HYS00910
      IF( ESR .LT. CC .AND. CYCLE .EQ. 1 ) THEN                                 HYS00920
          PMAX=-YS*AREA                                                         HYS00930
      ELSE IF( ESR .GE. CC .AND. CYCLE .EQ. 1 ) THEN                            HYS00940
          PMAX=-PI*PI*AREA*EM/(ESR**2)                                          HYS00950
      ENDIF                                                                     HYS00960
C                                                                               HYS00970
C RESIDUAL ELOGATION FACTOR RFAC                                                HYS00980
C                                                                               HYS00990
C                     /** FOR BOX AND ANGLE SECTION **/                        HYS01000
C     RFAC=1.75                                                                 HYS01010
C                     /** FOR I SHAPE SECTION       **/                        HYS01020
      RFAC=1.4                                                                  HYS01030
C                                                                               HYS01040
C GENERATE CONTROL POINTS COORDINATE                                            HYS01050
C                                                                               HYS01060
C  (1)  POINT A                                                                 HYS01070
C                                                                               HYS01080
      IF( CYCLE .EQ. 1 ) THEN                                                   HYS01090
          VA=PMAX/PT                                                            HYS01100
          HA=PMAX/(ES*DELT)                                                     HYS01110
      ELSE IF( CYCLE .EQ. 2) THEN                                               HYS01120
        IF( ESR .LE. 40 ) THEN                                                  HYS01130
          PMAX=-0.2*PT                                                          HYS01140
        ELSE IF( ESR .GT. 40 ) THEN                                             HYS01150
          PMAX=-0.2*PT                                                          HYS01160
        ENDIF                                                                   HYS01170
      ELSE IF( CYCLE .GT. 2) THEN                                               HYS01180
        IF( ESR .LE. 40 ) THEN                                                  HYS01190
          PMAX=-0.2*PT                                                          HYS01200
        ELSE IF( ESR .GT. 40 ) THEN                                             HYS01210
          PMAX=-0.2*PT                                                          HYS01220
        ENDIF                                                                   HYS01230
      ENDIF                                                                     HYS01240
C                                                                               HYS01250
C  (2)  POINT B                                                                 HYS01260
C                                                                               HYS01270
      HB=-5.                                                                    HYS01280
      IF( ESR .LE. 40 ) THEN                                                    HYS01290
        VB=-0.15                                                                HYS01300
      ELSE IF( ESR .GT. 40 ) THEN                                               HYS01310
C                     /** FOR BOX AND ANGLE SECTION **/                        HYS01320
C       VB=-0.15                                                                HYS01330
C                     /** FOR I SHAPE SECTION       **/                        HYS01340
        VB=-0.15                                                                HYS01350
      ENDIF                                                                     HYS01360
C                                                                               HYS01370
C  (3)  POINT C                                                                 HYS01380
C                                                                               HYS01390
      IF( IDUC .EQ. 0.) THEN                                                    HYS01400
        HC=-12.                                                                 HYS01410
        IF( ESR .LE. 40 ) THEN                                                  HYS01420
          VC=-0.1                                                               HYS01430
        ELSE IF( ESR .GT. 40 ) THEN                                             HYS01440
C                     /** FOR BOX AND ANGLE SECTION **/                        HYS01450
C         VC=-0.1                                                               HYS01460
C                     /** FOR I SHAPE SECTION       **/                        HYS01470
          VC=-0.1                                                               HYS01480
        ENDIF                                                                   HYS01490
      ENDIF                                                                     HYS01500
C                                                                               HYS01510
C  (4)  POINT E                                                                 HYS01520
C                                                                               HYS01530
          HE=1.+(RSN*EL/DELT)                                                   HYS01540
          VE=1.                                                                 HYS01550
C                                                                               HYS01560
C  (5)  POINT D1                                                                HYS01570
C                                                                               HYS01580
      SLOPE=(VE-VC)/(HE-HC)                                                     HYS01590
      HD1=(SLOPE*HC-VC)/(SLOPE+0.267949)                                        HYS01600
      VD1=-0.267949*HD1                                                         HYS01610
C                                                                               HYS01620
C  (6)  POINT D                                                                 HYS01630
C                                                                               HYS01640
      IF( ESR .LE. 40 ) THEN                                                    HYS01650
        HD=HD1*31./40                                                           HYS01660
        VD=VD1*31./40                                                           HYS01670
      ELSE IF( ESR .GT. 40 ) THEN                                               HYS01680
C                     /** FOR BOX AND ANGLE SECTION **/                        HYS01690
C       HD=HD1*60./40                                                           HYS01700
C       VD=VD1*60./40                                                           HYS01710
C                     /** FOR I SHAPE SECTION       **/                        HYS01720
        HD=HD1*20./40                                                           HYS01730
        VD=VD1*20./40                                                           HYS01740
      ENDIF                                                                     HYS01750
C                                                                               HYS01760
C DEFINE ENVELOPE SLOPE                                                         HYS01770
C                                                                               HYS01780
      SLOP1=1.                                                                  HYS01790
      SLOP2=(VB-VA)/(HB-HA)                                                     HYS01800
      SLOP3=(VC-VB)/(HC-HB)                                                     HYS01810
      SLOP4=(VD-VC)/(HD-HC)                                                     HYS01820
      SLOP5=(VE-VD)/(HE-HD)                                                     HYS01830
      SLOP6=0.0001                                                              HYS01840
C                                                                               HYS01850
C DEFINE ENVELOPE INTERMIDIATE POINTS                                           HYS01860
C                                                                               HYS01870
C  (1)  POINT H                                                                 HYS01880
C                                                                               HYS01890
      VH=(VB+SLOP2*(-VD+HD-HB))/(1.-SLOP2)                                      HYS01900
      HH=VH-VD+HD                                                               HYS01910
C                                                                               HYS01920
C  (2)  POINT C1                                                                HYS01930
C                                                                               HYS01940
      VC1=0.                                                                    HYS01950
      HC1=HC-(VC/SLOP4)                                                         HYS01960
C                                                                               HYS01970
C  (3)  POINT (PREF,DREF) : CALCULATE IN RULE NO. 4                             HYS01980
C                                                                               HYS01990
C                                                                               HYS02000
C  (4)  POINT (HREF,VREF) : CALCULATE IN RULE NO. 5                             HYS02010
C                                                                               HYS02020
C--- DEFINE CURRENT REGIN ZONE NUMBER                                           HYS02030
      PREGIN=REGIN                                                              HYS02040
      IF( DEL .GE. HA) REGIN=1                                                  HYS02050
      IF( DEL .GE. HH .AND. DEL .LT. HA ) REGIN=2                               HYS02060
      IF( DEL .GE. HB .AND. DEL .LT. HH ) REGIN=3                               HYS02070
      IF( DEL .GE. HC .AND. DEL .LT. HB ) REGIN=4                               HYS02080
C                                                                               HYS02090
      IF (PRINT) THEN                                                           HYS02100
      WRITE(6,567)CYCLE,RULE,HA,VA,HB,VB,HC,VC,HC1,VC1,                         HYS02110
     &            HD,VD,HD1,VD1,HH,VH,HE,VE                                     HYS02120
  567 FORMAT(5X,'GOEL HYSTERESIS ENVELOPE CONTROL POINTS....'/                  HYS02130
     &       5X,'=========================================='/                   HYS02140
     &       5X,'CYCLE    ---------=',F10.4,/                                  HYS02150
     &       5X,'RULE     ---------=',F10.4,/                                  HYS02160
     &       5X,'(HA, VA) ---------=',1X,'( ',F10.4,',',F10.4,' )',/           HYS02170
     &       5X,'(HB, VB) ---------=',1X,'( ',F10.4,',',F10.4,' )',/           HYS02180
     &       5X,'(HC, VC) ---------=',1X,'( ',F10.4,',',F10.4,' )',/           HYS02190
     &       5X,'(HC1, VC1) -------=',1X,'( ',F10.4,',',F10.4,' )',/           HYS02200
     &       5X,'(HD, VD) ---------=',1X,'( ',F10.4,',',F10.4,' )',/           HYS02210
     &       5X,'(HD1, VD1) -------=',1X,'( ',F10.4,',',F10.4,' )',/           HYS02220
     &       5X,'(HH, VH) ---------=',1X,'( ',F10.4,',',F10.4,' )',/           HYS02230
     &       5X,'(HE, VE) ---------=',1X,'( ',F10.4,',',F10.4,' )',/)          HYS02240
      ENDIF                                                                     HYS02250
C                                                                               HYS02260
C     GOEL'S HYSTORESIS LOOP RULES                                              HYS02270
```

```
C                                                                       HYS02290
C  (1) RULE 1                                                           HYS02300
C                                                                       HYS02310
      IF( RULE .EQ. 1.) THEN                                            HYS02320
            IF( P. GE. PMAX .AND. P .LT. PY ) THEN                      HYS02330
                  STIF=SLOP1*ES                                         HYS02340
                  EKI=SLOP1*ES                                          HYS02350
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN           HYS02360
                    RSN=((0.55*ABS(DE)/40)+0.0002*ABS(DE)**2)*RFAC      HYS02370
                  ENDIF                                                 HYS02380
            ELSE IF( P. GE. PY ) THEN                                   HYS02390
                  STIF=SLOP6                                            HYS02400
                  RULE=6                                                HYS02410
C                 RSN=0                                                 HYS02420
                  EKI=SLOP1*ES                                          HYS02430
            ELSE IF( P. LT. PMAX) THEN                                  HYS02440
                  HA=DEL                                                HYS02450
                  VA=PL                                                 HYS02460
                  SLOP2=(VB-VA)/(HB-HA)                                 HYS02470
                  STIF=SLOP2*ES                                         HYS02480
                  RULE=2                                                HYS02490
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN           HYS02500
                    RSN=((0.55*ABS(DE)/40)+0.0002*ABS(DE)**2)*RFAC      HYS02510
                  ENDIF                                                 HYS02520
                  EKI=SLOP1*ES                                          HYS02530
            ENDIF                                                       HYS02540
            CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)          HYS02550
            CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )    HYS02560
      RETURN                                                            HYS02570
      ENDIF                                                             HYS02580
C                                                                       HYS02590
C  (2) RULE 2                                                           HYS02600
C                                                                       HYS02610
      IF( RULE .EQ. 2.) THEN                                            HYS02620
          IF( DEL .GE. HH) THEN                                         HYS02630
                IF( DDE .LE. 0. ) THEN                                  HYS02640
                      IF(PREGIN .NE. REGIN ) REVRSC= .FALSE.            HYS02650
                      STIF=SLOP2*ES                                     HYS02660
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN           HYS02670
                    RSN=((0.55*ABS(DE)/40)+0.0002*ABS(DE)**2)*RFAC      HYS02680
                  ELSE IF( DEMAX .GT. 0 ) THEN                          HYS02690
                    RSN=((0.55*ABS(DE)/40)+0.0002*ABS(DE)**2)*RFAC      HYS02700
                  ENDIF                                                 HYS02710
C                   ------------------------------------------          HYS02720
                      IF(.NOT. REVRSC )THEN                             HYS02730
                         EKI=SLOP1*ES                                   HYS02740
                      ELSE IF( REVRSC )THEN                             HYS02750
                         EKI=ES*((PTOP-PL)/(DTOP-DEL))                  HYS02760
                      ENDIF                                             HYS02770
C                   ------------------------------------------          HYS02780
                ELSE IF( DDE .GT. 0. )THEN                              HYS02790
                   IF(.NOT. REVRSC )THEN                                HYS02800
                      RULE=7                                            HYS02810
                      STIF=SLOP1*ES                                     HYS02820
C                                                                       HYS02830
                      PLA=PL-SLOP2*DDEL                                 HYS02840
                      DLA=DEL-DDEL                                      HYS02850
                      DBOT=DLA                                          HYS02860
                      PBOT=PLA                                          HYS02870
                      PL=PLA+SLOP1*DDEL                                 HYS02880
                      P=PL*PY                                           HYS02890
C                                                                       HYS02900
                      PTOP=VB+(SLOP5*(PBOT-VB-SLOP1*DBOT                HYS02910
     *                      +SLOP1*HB)/(SLOP5-SLOP1))                   HYS02920
                      DTOP=(PBOT-VB+SLOP5*HB-SLOP1*DBOT)                HYS02930
     *                     /(SLOP5-SLOP1)                             HYS02940
                      EKI=ES*SLOP1                                      HYS02950
                   ELSE IF( REVRSC ) THEN                               HYS02960
                      RULE=12                                           HYS02970
                      DBOT=DEL                                          HYS02980
                      PBOT=P/PY                                         HYS02990
                      SLOP7=(PTOP-PBOT)/(DTOP-DBOT)                     HYS03000
                      STIF=SLOP7*ES                                     HYS03010
                      EKI=ES*SLOP7                                      HYS03020
                   ENDIF                                                HYS03030
                ENDIF                                                   HYS03040
          ELSE IF(DEL .LT. HH .AND. DEL .GE. HB ) THEN                  HYS03050
                IF( DDE .LE. 0. ) THEN                                  HYS03060
C                     IF(PREGIN .NE. REGIN ) REVRSC= .FALSE.            HYS03070
                      STIF=SLOP2*ES                                     HYS03080
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN           HYS03090
                    RSN=((0.55*ABS(DE)/40)+0.0002*ABS(DE)**2)*RFAC      HYS03100
                  ELSE IF( DEMAX .GT. 0 ) THEN                          HYS03110
                    RSN=((0.55*ABS(DE)/40)+0.0002*ABS(DE)**2)*RFAC      HYS03120
                  ENDIF                                                 HYS03130
C                   ------------------------------------------          HYS03140
                      IF(.NOT. REVRSC )THEN                             HYS03150
                         EKI=SLOP1*ES                                   HYS03160
                      ELSE IF( REVRSC )THEN                             HYS03170
                         EKI=ES*((PTOP-PL)/(DTOP-DEL))                  HYS03180
                      ENDIF                                             HYS03190
C                   ------------------------------------------          HYS03200
                ELSE IF( DDE .GT. 0. )THEN                              HYS03210
                   IF( .NOT. REVRSC ) THEN                              HYS03220
                      RULE=8                                            HYS03230
                      STIF=SLOP1*ES                                     HYS03240
C                                                                       HYS03250
                      PLA=PL-SLOP2*DDEL                                 HYS03260
                      DLA=DEL-DDEL                                      HYS03270
                      DBOT=DLA                                          HYS03280
                      PBOT=PLA                                          HYS03290
                      PL=PLA+SLOP1*DDEL                                 HYS03300
                      P=PL*PY                                           HYS03310
C                                                                       HYS03320
                      PTOP=VD+(SLOP4*(PBOT-VD-SLOP1*DBOT                HYS03330
     *                      +SLOP1*HD)/(SLOP4-SLOP1))                   HYS03340
                      DTOP=(PBOT-VD+SLOP4*HD-SLOP1*DBOT)                HYS03350
     *                     /(SLOP4-SLOP1)                             HYS03360
                      EKI=ES*SLOP1                                      HYS03370
                   ELSE IF( REVRSC ) THEN                               HYS03380
                      RULE=11                                           HYS03390
                      DBOT=DEL                                          HYS03400
                      PBOT=P/PY                                         HYS03410
                      PTOP=VD                                           HYS03420
                      DTOP=HD                                           HYS03430
                      SLOP7=(PTOP-PBOT)/(DTOP-DBOT)                     HYS03440
                      STIF=SLOP7*ES                                     HYS03450
                      EKI=ES*SLOP7                                      HYS03460
                   ENDIF                                                HYS03470
                ENDIF                                                   HYS03480
          ELSE IF( DEL .LT. HB) THEN                                    HYS03490
                IF(PREGIN .NE. REGIN ) REVRSC= .FALSE.                  HYS03500
                RULE=3                                                  HYS03510
                EKI=ES*SLOP1                                            HYS03520
          ENDIF                                                         HYS03530
      IF( REVRSC ) IRV=1.                                               HYS03540
      IF ( .NOT. REVRSC) IRV=0.                                         HYS03550
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)                HYS03560
            IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                          HYS03570
              CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )  HYS03580
              DEMAX=DE                                                  HYS03590
            ENDIF                                                       HYS03600
      RETURN                                                            HYS03610
      ENDIF                                                             HYS03620
C                                                                       HYS03630
C  (3) RULE 3                                                           HYS03640
C                                                                       HYS03650
      IF( RULE .EQ. 3.) THEN                                            HYS03660
          IF( DEL .GT. HC) THEN                                         HYS03670
                IF( DDE .LE. 0. ) THEN                                  HYS03680
                      IF(PREGIN .NE. REGIN ) REVRSC= .FALSE.            HYS03690
                      STIF=SLOP3*ES                                     HYS03700
                      IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN       HYS03710
                        RSN=((0.55*ABS(DE)/40)+0.0002*ABS(DE)**2)*RFAC  HYS03720
                      ELSE IF( DEMAX .GT. 0 ) THEN                      HYS03730
                        RSN=((0.55*ABS(DE)/40)+0.0002*ABS(DE)**2)*RFAC  HYS03740
                      ENDIF                                             HYS03750
C                       ------------------------------------------      HYS03760
                        IF(.NOT. REVRSC )THEN                           HYS03770
                           EKI=SLOP1*ES                                 HYS03780
                        ELSE IF( REVRSC )THEN                           HYS03790
                           EKI=ES*((PTOP-PL)/(DTOP-DEL))                HYS03800
                        ENDIF                                           HYS03810
C                       ------------------------------------------      HYS03820
                    ELSE IF( DDE .GT. 0. )THEN                          HYS03830
                      IF( DEL .GT. -12. ) THEN                          HYS03840
                        IF( .NOT. REVRSC )THEN                          HYS03850
                           RULE=9                                       HYS03860
                           STIF=SLOP1*ES                                HYS03870
C                                                                       HYS03880
                           PLA=PL-SLOP2*DDEL                            HYS03890
                           DLA=DEL-DDEL                                 HYS03900
                           DBOT=DLA                                     HYS03910
                           PBOT=PLA                                     HYS03920
                           PL=PLA+SLOP1*DDEL                            HYS03930
                           P=PL*PY                                      HYS03940
C                                                                       HYS03950
                           PTOP=VD+(SLOP4*(PBOT-VD-SLOP1*DBOT           HYS03960
     *                           +SLOP1*HD)/(SLOP4-SLOP1))              HYS03970
                           DTOP=(PBOT-VD+SLOP4*HD-SLOP1*DBOT)           HYS03980
     *                          /(SLOP4-SLOP1)                        HYS03990
                           EKI=ES*SLOP1                                 HYS04000
                        ELSE IF( REVRSC ) THEN                          HYS04010
                           RULE=10                                      HYS04020
                           DBOT=DEL                                     HYS04030
                           PBOT=P/PY                                    HYS04040
                           SLOP7=(PTOP-PBOT)/(DTOP-DBOT)                HYS04050
                           STIF=SLOP7*ES                                HYS04060
                           EKI=ES*SLOP7                                 HYS04070
                         ENDIF                                          HYS04080
                      ELSE IF ( DEL .LE. -12. ) THEN                    HYS04090
                         REVRSC= .FALSE.                                HYS04100
                         RULE=4                                         HYS04110
                         STIF=SLOP4*ES                                  HYS04120
                         EKI=ES*SLOP4                                   HYS04130
                      ENDIF                                             HYS04140
                    ENDIF                                               HYS04150
          ELSE IF( DEL .LE. HC ) THEN                                   HYS04160
            IF( DDE .LT. 0 ) THEN                                       HYS04170
              RULE=3                                                    HYS04180
              HC=DEL                                                    HYS04190
              VC=PL                                                     HYS04200
              IDUC=1.                                                   HYS04210
              STIF=-SLOP6*ES                                            HYS04220
              EKI=ES*((VD-VC)/(HD-HC))                                  HYS04230
                    IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN         HYS04240
                      RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC   HYS04250
                    ELSE IF( DEMAX .GT. 0 ) THEN                        HYS04260
                      RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC   HYS04270
                    ENDIF                                               HYS04280
             ELSE IF( DDE .GT. 0 ) THEN                                 HYS04290
              RULE=4                                                    HYS04300
              STIF=SLOP4*ES                                             HYS04310
              EKI=ES*SLOP4                                              HYS04320
             ENDIF                                                      HYS04330
            ENDIF                                                       HYS04340
      IF( REVRSC ) IRV=1.                                               HYS04350
      IF ( .NOT. REVRSC) IRV=0.                                         HYS04360
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)                HYS04370
            IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                          HYS04380
              CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )  HYS04390
              DEMAX=DE                                                  HYS04400
            ENDIF                                                       HYS04410
      RETURN                                                            HYS04420
      ENDIF                                                             HYS04430
C                                                                       HYS04440
C  (4) RULE 7                                                           HYS04450
C                                                                       HYS04460
      IF( RULE .EQ. 7.) THEN                                            HYS04470
          IF( DEL .GT. DBOT .AND. DEL .LT. DTOP .AND. P .GT. PMAX) THEN HYS04480
                STIF=SLOP1*ES                                           HYS04490
                EKI=ES*SLOP1                                            HYS04500
          ELSE IF(DEL .LT. DBOT .OR. P .LE. PMAX) THEN                  HYS04510
                IF( P .LE. PMAX ) THEN                                  HYS04520
                   REVRSC= .FALSE.                                      HYS04530
                   PTOP=0                                               HYS04540
                   DTOP=0                                               HYS04550
                   PBOT=PL                                              HYS04560
                   DBOT=DEL                                             HYS04570
                   HA=DEL                                               HYS04580
                   VA=PL                                                HYS04590
                   SLOPE2=(VB-VA)/(HB-HA)                               HYS04600
                ENDIF                                                   HYS04610
                STIF=SLOP2*ES                                           HYS04620
                RULE=2                                                  HYS04630
C                                                                       HYS04640
                PL=PBOT-(DBOT-DEL)*SLOP2                                HYS04650
                P=PL*PY                                                 HYS04660
C                                                                       HYS04670
                EKI=ES*((PTOP-PL)/(DTOP-DEL))                           HYS04680
          ELSE IF(DEL .GT. DTOP .AND. PL .LT. VE) THEN                  HYS04690
                STIF=SLOP5*ES                                           HYS04700
                RULE=5                                                  HYS04710
                EKI=ES*SLOP1                                            HYS04720
          ELSE IF(DEL .GT. DTOP .AND. PL .GE. VE) THEN                  HYS04730
                STIF=SLOP6*ES                                           HYS04740
                RULE=6                                                  HYS04750
                EKI=ES*SLOP1                                            HYS04760
          ENDIF                                                         HYS04770
      IF( REVRSC ) IRV=1.                                               HYS04780
      IF ( .NOT. REVRSC) IRV=0.                                         HYS04790
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)                HYS04800
C     CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )          HYS04810
      RETURN                                                            HYS04820
      ENDIF                                                             HYS04830
C                                                                       HYS04840
C  (5) RULE 8                                                           HYS04850
C                                                                       HYS04860
      IF( RULE .EQ. 8.) THEN                                            HYS04870
          IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                  HYS04880
                STIF=SLOP1*ES                                           HYS04890
                EKI=ES*SLOP1                                            HYS04900
          ELSE IF(DEL .LT. DBOT ) THEN                                  HYS04910
                STIF=SLOP2*ES                                           HYS04920
                RULE=2                                                  HYS04930
C                                                                       HYS04940
                PL=PBOT-(DBOT-DEL)*SLOP2                                HYS04950
                P=PL*PY                                                 HYS04960
C                                                                       HYS04970
                EKI=ES*((PTOP-PL)/(DTOP-DEL))                           HYS04980
          ELSE IF(DEL .GT. DTOP ) THEN                                  HYS04990
                STIF=SLOP4*ES                                           HYS05000
                RULE=4                                                  HYS05010
                EKI=ES*SLOP1                                            HYS05020
          ENDIF                                                         HYS05030
      IF( REVRSC ) IRV=1.                                               HYS05040
      IF ( .NOT. REVRSC) IRV=0.                                         HYS05050
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)                HYS05060
C     CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )          HYS05070
      RETURN                                                            HYS05080
      ENDIF                                                             HYS05090
C                                                                       HYS05100
C  (6) RULE 9                                                           HYS05110
C                                                                       HYS05120
```

```
      IF( RULE .EQ. 9.) THEN                                                    HYS05130
          IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                          HYS05140
                STIF=SLOP1*ES                                                   HYS05150
                EKI=ES*SLOP1                                                    HYS05160
          ELSE IF(DEL .LT. DBOT ) THEN                                          HYS05170
                STIF=SLOP3*ES                                                   HYS05180
                RULE=3                                                          HYS05190
C                                                                               HYS05200
                PL=PBOT-(DBOT-DEL)*SLOP3                                        HYS05210
                P=PL*PY                                                         HYS05220
C                                                                               HYS05230
                EKI=ES*((PTOP-PL)/(DTOP-DEL))                                   HYS05240
          ELSE IF(DEL .GT. DTOP ) THEN                                          HYS05250
                STIF=SLOP4*ES                                                   HYS05260
                RULE=4                                                          HYS05270
                EKI=ES*SLOP1                                                    HYS05280
          ENDIF                                                                 HYS05290
      IF( REVRSC ) IRV=1.                                                       HYS05300
      IF ( .NOT. REVRSC) IRV=0.                                                 HYS05310
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)                        HYS05320
C     CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )                  HYS05330
      RETURN                                                                    HYS05340
      ENDIF                                                                     HYS05350
C                                                                               HYS05360
C  (7) RULE 4                                                                   HYS05370
C                                                                               HYS05380
      IF( RULE .EQ. 4.) THEN                                                    HYS05390
          IF (DEL .GE. HC1 .AND. DEL .LT. HD ) THEN                             HYS05400
              IF(DDE .GE. 0. ) THEN                                             HYS05410
                  STIF=SLOP4*ES                                                 HYS05420
                  EKI=ES*SLOP1                                                  HYS05430
              ELSE IF( DDE .LT. 0. ) THEN                                       HYS05440
                  PREF=VD+(SLOP4*(VB-VD-SLOP1*HB                                HYS05450
     *                 +SLOP1*HD)/(SLOP4-SLOP1))                                HYS05460
                  DREF=(VB-VD+SLOP4*HD-SLOP1*HB)/(SLOP4-SLOP1)                  HYS05470
                  IF( DEL .LT. DREF) THEN                                       HYS05480
                      PTOP=P/PY                                                 HYS05490
                      DTOP=DEL                                                  HYS05500
                      DBOT=(SLOP3*HB-VB+PTOP-SLOP1*DEL)/(SLOP3-SLOP1)           HYS05510
                      PBOT=VB+SLOP3*(DBOT-HB)                                   HYS05520
                      STIF=SLOP1*ES                                             HYS05530
                      RULE=9                                                    HYS05540
                      REVRSC= .TRUE.                                            HYS05550
                      EKI=ES*SLOP1                                              HYS05560
                  ELSE IF( DEL .GE. DREF) THEN                                  HYS05570
                      PTOP=P/PY                                                 HYS05580
                      DTOP=DEL                                                  HYS05590
                      DBOT=(SLOP2*HA-VA+PTOP-SLOP1*DEL)/(SLOP2-SLOP1)           HYS05600
                      PBOT=VA+SLOP2*(DBOT-HA)                                   HYS05610
                      STIF=SLOP1*ES                                             HYS05620
                      RULE=8                                                    HYS05630
                      REVRSC= .TRUE.                                            HYS05640
                      EKI=ES*SLOP1                                              HYS05650
                  ENDIF                                                         HYS05660
              ENDIF                                                             HYS05670
          ELSE IF( DEL .GT. HD .AND. DDE .GT. 0. ) THEN                         HYS05680
                STIF=SLOP5*ES                                                   HYS05690
                RULE=5                                                          HYS05700
                EKI=ES*SLOP1                                                    HYS05710
          ELSE IF( DEL .LT. HC1 .AND. DEL .GE. HC ) THEN                        HYS05720
                REVRSC= .FALSE.                                                 HYS05730
                RULE=4                                                          HYS05740
                STIF=SLOP4*ES                                                   HYS05750
                EKI=ES*SLOP4                                                    HYS05760
          ELSE IF( DEL .LT. HC ) THEN                                           HYS05770
              REVRSC= .FALSE.                                                   HYS05780
              RULE=3                                                            HYS05790
              HC=DEL                                                            HYS05800
              VC=PL                                                             HYS05810
              IDUC=1.                                                           HYS05820
              STIF=-SLOP6*ES                                                    HYS05830
              EKI=ES*((VD-VC)/(HD-HC))                                          HYS05840
          ENDIF                                                                 HYS05850
      IF( REVRSC ) IRV=1.                                                       HYS05860
      IF ( .NOT. REVRSC) IRV=0.                                                 HYS05870
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)                        HYS05880
          IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                                    HYS05890
            CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )            HYS05900
            DEMAX=DE                                                            HYS05910
          ENDIF                                                                 HYS05920
      RETURN                                                                    HYS05930
      ENDIF                                                                     HYS05940
C                                                                               HYS05950
C  (8) RULE 5                                                                   HYS05960
C                                                                               HYS05970
      IF( RULE .EQ. 5.) THEN                                                    HYS05980
          IF (DEL .LT. HE ) THEN                                                HYS05990
              IF(DDE .GE. 0. ) THEN                                             HYS06000
                  STIF=SLOP5*ES                                                 HYS06010
                  EKI=ES*SLOP1                                                  HYS06020
              ELSE IF( DDE .LT. 0. ) THEN                                       HYS06030
                    BREF=VA-SLOP1*HA                                            HYS06040
                    HREF=(VD-SLOP5*HD-BREF)/(SLOP1-SLOP5)                       HYS06050
                    IF( DEL .GE. HREF ) CYCLE=CYCLE+1                           HYS06060
                    PTOP=P/PY                                                   HYS06070
                    DTOP=DEL                                                    HYS06080
                    DBOT=(SLOP2*HA-VA+PTOP-SLOP1*DEL)/(SLOP2-SLOP1)             HYS06090
                    PBOT=VA+SLOP2*(DBOT-HA)                                     HYS06100
                    STIF=SLOP1*ES                                               HYS06110
                    RULE=7                                                      HYS06120
                    REVRSC= .TRUE.                                              HYS06130
                    EKI=ES*SLOP1                                                HYS06140
              ENDIF                                                             HYS06150
          ELSE IF( DEL .GE. HE .AND. DDE .GT. 0. ) THEN                         HYS06160
                STIF=SLOP6                                                      HYS06170
                RULE=6                                                          HYS06180
                EKI=ES*SLOP1                                                    HYS06190
          ENDIF                                                                 HYS06200
      IF( REVRSC ) IRV=1.                                                       HYS06210
      IF ( .NOT. REVRSC) IRV=0.                                                 HYS06220
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)                        HYS06230
          IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                                    HYS06240
            CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )            HYS06250
            DEMAX=DE                                                            HYS06260
          ENDIF                                                                 HYS06270
      RETURN                                                                    HYS06280
      ENDIF                                                                     HYS06290
C                                                                               HYS06300
C  (9) RULE 6                                                                   HYS06310
C                                                                               HYS06320
      IF( RULE .EQ. 6.) THEN                                                    HYS06330
          IF(DDE .GE. 0.) THEN                                                  HYS06340
                STIF=SLOP6                                                      HYS06350
                EKI=ES*SLOP1                                                    HYS06360
          ELSE IF(DDE .LT. 0.) THEN                                             HYS06370
                STIF=SLOP1*ES                                                   HYS06380
                RULE=1                                                          HYS06390
                CYCLE=CYCLE+1.                                                  HYS06400
                REVRSC= .FALSE.                                                 HYS06410
                EKI=ES*SLOP1                                                    HYS06420
          ENDIF                                                                 HYS06430
      IF( REVRSC ) IRV=1.                                                       HYS06440
      IF ( .NOT. REVRSC) IRV=0.                                                 HYS06450
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)                        HYS06460
          IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                                    HYS06470
            CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )            HYS06480
            DEMAX=DE                                                            HYS06490
          ENDIF                                                                 HYS06500
      RETURN                                                                    HYS06510
      ENDIF                                                                     HYS06520
C                                                                               HYS06530
C  (10) RULE 10                                                                 HYS06540
C                                                                               HYS06550
      IF( RULE .EQ. 10.) THEN                                                   HYS06560
          IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                          HYS06570
                STIF=SLOP7*ES                                                   HYS06580
                EKI=ES*SLOP7                                                    HYS06590
          ELSE IF(DEL .LT. DBOT ) THEN                                          HYS06600
                STIF=SLOP3*ES                                                   HYS06610
                RULE=3                                                          HYS06620
                REVRSC= .FALSE.                                                 HYS06630
                PTOP=0                                                          HYS06640
                DTOP=0                                                          HYS06650
                PBOT=0                                                          HYS06660
                DBOT=0                                                          HYS06670
                EKI=ES*((PTOP-PL)/(DTOP-DEL))                                   HYS06680
          ELSE IF(DEL .GT. DTOP ) THEN                                          HYS06690
                STIF=SLOP4*ES                                                   HYS06700
                RULE=4                                                          HYS06710
                EKI=ES*SLOP1                                                    HYS06720
          ENDIF                                                                 HYS06730
      IF( REVRSC ) IRV=1.                                                       HYS06740
      IF ( .NOT. REVRSC) IRV=0.                                                 HYS06750
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)                        HYS06760
C     CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )                  HYS06770
      RETURN                                                                    HYS06780
      ENDIF                                                                     HYS06790
C                                                                               HYS06800
C  (11) RULE 11                                                                 HYS06810
C                                                                               HYS06820
      IF( RULE .EQ. 11.) THEN                                                   HYS06830
          IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                          HYS06840
                STIF=SLOP7*ES                                                   HYS06850
                EKI=ES*SLOP7                                                    HYS06860
          ELSE IF(DEL .LT. DBOT ) THEN                                          HYS06870
                STIF=SLOP2*ES                                                   HYS06880
                RULE=2                                                          HYS06890
                REVRSC= .FALSE.                                                 HYS06900
                PTOP=0                                                          HYS06910
                DTOP=0                                                          HYS06920
                PBOT=0                                                          HYS06930
                DBOT=0                                                          HYS06940
                EKI=ES*((VD-PL)/(HD-DEL))                                       HYS06950
          ELSE IF(DEL .GT. DTOP ) THEN                                          HYS06960
                STIF=SLOP4*ES                                                   HYS06970
                RULE=4                                                          HYS06980
                EKI=ES*SLOP1                                                    HYS06990
          ENDIF                                                                 HYS07000
      IF( REVRSC ) IRV=1.                                                       HYS07010
      IF ( .NOT. REVRSC) IRV=0.                                                 HYS07020
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)                        HYS07030
C     CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )                  HYS07040
      RETURN                                                                    HYS07050
      ENDIF                                                                     HYS07060
C                                                                               HYS07070
C  (12) RULE 12                                                                 HYS07080
C                                                                               HYS07090
      IF( RULE .EQ. 12.) THEN                                                   HYS07100
          IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                          HYS07110
                STIF=SLOP7*ES                                                   HYS07120
                EKI=ES*SLOP7                                                    HYS07130
          ELSE IF(DEL .LT. DBOT ) THEN                                          HYS07140
                STIF=SLOP2*ES                                                   HYS07150
                RULE=2                                                          HYS07160
                REVRSC= .FALSE.                                                 HYS07170
                PTOP=0                                                          HYS07180
                DTOP=0                                                          HYS07190
                PBOT=0                                                          HYS07200
                DBOT=0                                                          HYS07210
                EKI=ES*((PTOP-PL)/(DTOP-DEL))                                   HYS07220
          ELSE IF(DEL .GT. DTOP ) THEN                                          HYS07230
                STIF=SLOP5*ES                                                   HYS07240
                RULE=5                                                          HYS07250
                EKI=ES*SLOP1                                                    HYS07260
          ENDIF                                                                 HYS07270
      ENDIF                                                                     HYS07280
      IF( REVRSC ) IRV=1.                                                       HYS07290
      IF ( .NOT. REVRSC) IRV=0.                                                 HYS07300
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,STIF,EKI)                        HYS07310
C     CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )                  HYS07320
      RETURN                                                                    HYS07330
      END                                                                       HYS07340
C-----------------------------------------------------------------------       HYS00010
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                          HYS00020
C           END DEBUG                                                         HYS00030
C-----------------------------------------------------------------------       HYS00040
C** THIS HYSTERESIS SUBROUTINE FOR PERFORATED SHEAR WALL....HYST14.....       HYS00050
C-----------------------------------------------------------------------       HYS00060
C                                                                               HYS00070
      SUBROUTINE HYST14(FP     ,DP     ,FPP    ,DPP    ,H      ,              HYS00080
     &            OFC    ,ODC    ,OFY    ,ODY    ,OFU    ,ODU    ,            HYS00090
     &            OFF    ,ODF    ,BDAM   ,HO     ,HS     ,LWP    ,            HYS00100
     &            AD     ,AV     ,MQD    ,OSOC   ,OSCY   ,OSOY   ,            HYS00110
     &            OSYU   ,OSUF   ,OSCU   ,OSOB   ,OSOE   ,                    HYS00120
     &            K      ,LRULE  ,KKL    ,DSINC  ,DMAX   ,DM     ,            HYS00130
     &            FM     ,KP     ,SOC    ,SCY    ,SOY    ,STU    ,            HYS00140
     &            SUF    ,SCU    ,SOB    ,SOE    ,FUO    ,DUO    ,            HYS00150
     &            KEYDS2 ,SECNO  ,SZY    ,DU17   ,FU17   ,AREA1  ,            HYS00160
     &            CYCLEN ,CSE    ,FU00   ,DU00   ,DKP    ,TNPTFG,             HYS00170
     &            POSTFG ,CO1    ,RAT    ,RAT1   ,SQ6    ,SR     ,            HYS00180
     &            DP20   ,SRFLAG ,USE    ,IOPT2  ,IOPT   ,                    HYS00190
     &            PSQ6   ,KLAST  ,DPTT   ,DPT    ,ESE    ,PSE    ,            HYS00200
     &            PSEOLD ,EXCR   ,FC     ,FY     ,FU     ,FF     ,            HYS00210
     &            DC     ,DY     ,DU     ,DF     ,DSINCL ,UPOS   ,            HYS00220
     &            UNEG   )                                                     HYS00230
C                                                                               HYS00240
      INTEGER CYCLEN,LRULE,LRULST,IOPT                                          HYS00250
      REAL K,KP,DUO(4),FUO(4),KLAST,KEYDS1,KEYDS2,MQD,LWP                       HYS00260
      REAL UPOS(3),UNEG(3),EXCR(6),ESE(3)                                       HYS00270
      LOGICAL SRFLAG,TNPTFG,POSTFG                                              HYS00280
C                                                                               HYS00290
C                                                                               HYS00300
C-----------------------------------------------------------------------       HYS00310
C           DEFINITION FOR VARIABLE :                                         HYS00320
C-----------------------------------------------------------------------       HYS00330
C HS ----- WALL SHEAR SPAN,FROM LOADING POINT TO THE BOTTOM OF WALL            HYS00340
C WID----- WALL WIDTH                                                          HYS00350
C HO ----- THE HEIGHT OF THE OPENING                                           HYS00360
C H  ----- THE HEIGHT OF THE WALL                                              HYS00370
C AD ----- THE NET AREA OF DIAGONAL STEEL BARS                                 HYS00380
C AV ----- THE NET AREA OF VERTICAL STEEL BARS                                 HYS00390
C HO,H ABOVE JUST USED FOR CALCULATION OF FAILURE DISPL.(STIFF SUBRTN)         HYS00400
C ODC ----INITIAL (OR ORIGINAL) CRACKING DISPL.                                HYS00410
C ODY ----INITIAL YIELDING DISPL.                                              HYS00420
C ODU ----INITIAL ULTIMATE DISPL.                                              HYS00430
C ODF ----INITIAL FAILURE DISPL.                                               HYS00440
C OFC ----INITIAL CRACKING LOAD                                                HYS00450
C OFY ----INITIAL YIELDING LOAD                                                HYS00460
C OFU ----INITIAL ULTIMATE LOAD                                                HYS00470
C OFF ----INITIAL FAILURE LOAD                                                 HYS00480
C DC ----CRACKING DISPL.                                                       HYS00490
C DY ----YIELDING DISPL.                                                       HYS00500
C DU ----ULTIMATE DISPL.                                                       HYS00510
C DF ----FAILURE DISPL.                                                        HYS00520
C FC ----CRACKING LOAD                                                         HYS00530
C FY ----YIELDING LOAD                                                         HYS00540
C FU ----ULTIMATE LOAD                                                         HYS00550
C FF ----FAILURE LOAD                                                          HYS00560
C DMAX --MAX. DISPL. FOR THE LOADING SENSE IN THE CURRENT CYCLE                HYS00570
C DCP ----CRACKING DISPL. IN THE NEGATIVE LOADING SENSE                        HYS00580
C DYP ----YIELDING DISPL. IN THE NEGATIVE LOADING SENSE                        HYS00590
C DKP---SEE DFORCE SUBRTN; WHICH EQUAL TO CURRENT DP AND ASSIGNED TO           HYS00600
C       DPP JUST BEFORE CHANGING DP TO A NEW VALUE                             HYS00610
C FCP ----CRACKING LOAD IN THE NEGATIVE LOADING SENSE                          HYS00620
```

```
C FYP ----YIELDING LOAD IN THE NEGATIVE LOADING SENSE                    HYS00630
C DP ,FP-----CURRENT DISPL. ,CURRENT LOAD                                HYS00640
C FF -----FAILURE LOAD=0.5*(FC+FY)                                       HYS00650
C FM -----MAX. LOAD BEFORE THE CURRENT CYCLE                             HYS00660
C FU0 ,DU0-----LOAD AND DISPL. FOR UNLOADING POINT U0                    HYS00670
C FY08----0.8*FY                                                         HYS00680
C KKL=1 --- FOR POSITIVE LOADING                                         HYS00690
C KKL=2 --- FOR POSITIVE UNLOADING                                       HYS00700
C KKL=3 --- FOR NEGATIVE LOADING                                         HYS00710
C KKL=4 --- FOR NEGATIVE UNLOADING                                       HYS00720
C KKL=5 --- FOR REVERSAL LOADING FROM POSITIVE TO NEGATIVE               HYS00730
C KKL=6 --- FOR REVERSAL LOADING FROM NEGATIVE TO POSITIVE               HYS00740
C KKL=7 --- FOR RELOADING AFTER UNLOADING IN POSITIVE DIRECTION          HYS00750
C KKL=8 --- FOR RELOADING AFTER UNLOADING IN NEGATIVE DIRECTION          HYS00760
C SRFLAG --- SEE NEWSTI SUBRTN;TO CONTROL REVERS. LOAD IN YIELD STATE    HYS00770
C MQD --- M/(Q*D) ,SHEAR-SPAN-LENGTH RATIO                               HYS00780
C LWP-----L'/W' FOR OPENING SHEAR WALL CASE                              HYS00790
C KLAST----LAST STIFFNESS                                                HYS00800
C KEYDS1=KEYDS(SECNO-2)  ,KEYDS2-KEYDS(SECNO-1)                          HYS00810
C FU0(1)=FU0(2*CYCLEN); FU0(2)=FU0(2*CYCLEN-1)                           HYS00820
C FU0(3)=FU0(2*CYCLEN-2); FU0(4)=FU0(2*CYCLEN-3)                         HYS00830
C DU0(1)=DU0(2*CYCLEN); DU0(2)=DU0(2*CYCLEN-1)                           HYS00840
C USE= EQUIVALENT UNLOADING STIFFNESS (SAME AS SE IN THE HYST03)         HYS00850
C                                                                        HYS00860
C                                                                        HYS00870
      LRULST=LRULE                                                       HYS00880
      KLAST=K                                                            HYS00890
      KKLLST=KKL                                                         HYS00900
      DSINC=DP-DPP                                                       HYS00910
C ---- IF WALL FAIL(K=0) AT FORCE=0 THEN STILL K=0 AND RETURN            HYS00920
      IF(K.EQ.0) THEN                                                    HYS00930
        K=0.                                                             HYS00940
        FP=0.                                                            HYS00950
        FPP=0.                                                           HYS00960
        DP=DP                                                            HYS00970
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DP,FP,AREA1,            HYS00980
     &              K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FU00,LRULE)    HYS00990
        DSINCL=DSINC                                                     HYS01000
        STOP                                                             HYS01010
      END IF                                                             HYS01020
C                                                                        HYS01030
C--- WHEN DISPLACEMENT INCREMENT CHANGE SIGN, REMEMBER TWO PREVIOUS       HYS01040
C    MAX. DISP. OF TWO HALF CYCLES                                       HYS01050
C                                                                        HYS01060
      IF((DSINCL*DSINC).LT.0) THEN                                       HYS01070
      KEYDS1=KEYDS2                                                      HYS01080
      KEYDS2=DPP                                                         HYS01090
C                                                                        HYS01100
C TO MEMORISE THE POINTS AT WHICH DSINC*DSINCL < 0.                      HYS01110
C                                                                        HYS01120
        IF(POSTFG) THEN                                                  HYS01130
          IF(KKLLST.EQ.2 .AND. KKL.EQ.5) THEN                            HYS01140
            FU0(2)=0.                                                    HYS01150
            DU0(2)=0.                                                    HYS01160
          END IF                                                         HYS01170
          IF(KKLLST.EQ.4 .AND. KKL.EQ.6) THEN                            HYS01180
            FU0(1)=0.                                                    HYS01190
            DU0(1)=0.                                                    HYS01200
          END IF                                                         HYS01210
          IF(DSINC.LT.0) THEN                                            HYS01220
            IF(KKLLST.EQ.8 .OR. KKLLST.EQ.10) THEN                       HYS01230
              FU0(1)=MAX(FU0(1),FPP)                                     HYS01240
              DU0(1)=MAX(DU0(1),KEYDS2)                                  HYS01250
            ELSE                                                         HYS01260
              FU0(1)=FPP                                                 HYS01270
              DU0(1)=KEYDS2                                              HYS01280
            END IF                                                       HYS01290
          END IF                                                         HYS01300
          IF(DSINC.GT.0) THEN                                            HYS01310
            IF(KKLLST.EQ.7 .OR. KKLLST.EQ.9) THEN                        HYS01320
              FU0(2)=MIN(FU0(2),FPP)                                     HYS01330
              DU0(2)=MIN(DU0(2),KEYDS2)                                  HYS01340
            ELSE                                                         HYS01350
              FU0(2)=FPP                                                 HYS01360
              DU0(2)=KEYDS2                                              HYS01370
            END IF                                                       HYS01380
          END IF                                                         HYS01390
        ELSE                                                             HYS01400
          IF(KKLLST.EQ.2 .AND. KKL.EQ.5) THEN                            HYS01410
            FU0(1)=0.                                                    HYS01420
            DU0(1)=0.                                                    HYS01430
          END IF                                                         HYS01440
          IF(KKLLST.EQ.4 .AND. KKL.EQ.6) THEN                            HYS01450
            FU0(2)=0.                                                    HYS01460
            DU0(2)=0.                                                    HYS01470
          END IF                                                         HYS01480
          IF(DSINC.GT.0) THEN                                            HYS01490
            IF(KKLLST.EQ.7 .OR. KKLLST.EQ.9) THEN                        HYS01500
              FU0(1)=MIN(FU0(1),FPP)                                     HYS01510
              DU0(1)=MIN(DU0(1),KEYDS2)                                  HYS01520
            ELSE                                                         HYS01530
              FU0(1)=FPP                                                 HYS01540
              DU0(1)=KEYDS2                                              HYS01550
            END IF                                                       HYS01560
          END IF                                                         HYS01570
          IF(DSINC.LT.0) THEN                                            HYS01580
            IF(KKLLST.EQ.8 .OR. KKLLST.EQ.10) THEN                       HYS01590
              FU0(2)=MAX(FU0(2),FPP)                                     HYS01600
              DU0(2)=MAX(DU0(2),KEYDS2)                                  HYS01610
            ELSE                                                         HYS01620
              FU0(2)=FPP                                                 HYS01630
              DU0(2)=KEYDS2                                              HYS01640
            END IF                                                       HYS01650
          END IF                                                         HYS01660
        END IF                                                           HYS01670
C                                                                        HYS01680
        IF((ABS(KEYDS2).LT.DC).AND.(KKL.EQ.6).AND.(KEYDS2.GT.0.)) THEN   HYS01690
          KKL=2                                                          HYS01700
          GO TO 23                                                       HYS01710
        END IF                                                           HYS01720
        IF((ABS(KEYDS2).LT.DC).AND.(KKL.EQ.5).AND.(KEYDS2.LT.0.)) THEN   HYS01730
          KKL=4                                                          HYS01740
          GO TO 23                                                       HYS01750
        END IF                                                           HYS01760
        IF((KKL.EQ.2).AND.((DM.GE.DC).OR.(DMAX.GE.DC))) THEN             HYS01770
          KKL=8                                                          HYS01780
          GO TO 23                                                       HYS01790
        END IF                                                           HYS01800
        IF((KKL.EQ.2).AND.(DM.LT.DC).AND.(DMAX.LT.DC)) THEN              HYS01810
          KKL=1                                                          HYS01820
          GO TO 23                                                       HYS01830
        END IF                                                           HYS01840
        IF((KKL.EQ.4).AND.((DM.GE.DC).OR.(DMAX.GE.DC))) THEN             HYS01850
          KKL=7                                                          HYS01860
          GO TO 23                                                       HYS01870
        END IF                                                           HYS01880
        IF((KKL.EQ.4).AND.(DM.LT.DC).AND.(DMAX.LT.DC)) THEN              HYS01890
          KKL=3                                                          HYS01900
          GO TO 23                                                       HYS01910
        END IF                                                           HYS01920
C                                                                        HYS01930
        IF(KKL.EQ.1) THEN                                                HYS01940
          KKL=2                                                          HYS01950
          GO TO 23                                                       HYS01960
        ELSE IF(KKL.EQ.3) THEN                                           HYS01970
          KKL=4                                                          HYS01980
          GO TO 23                                                       HYS01990
        ELSE IF((KKL.EQ.2).AND.(FP.LE.0).AND.(DPP.NE.0)                  HYS02000
     &          .AND.(DSINC.LT.0)) THEN                                  HYS02010
          KKL=5                                                          HYS02020
          GO TO 23                                                       HYS02030
        ELSE IF((KKL.EQ.4).AND.(FP.LE.0).AND.(DPP.NE.0)                  HYS02040
     &          .AND.(DSINC.GT.0)) THEN                                  HYS02050
          KKL=6                                                          HYS02060
          GO TO 23                                                       HYS02070
        END IF                                                           HYS02080
        IF(KKL.EQ.8) THEN                                                HYS02090
          KKL=2                                                          HYS02100
          GO TO 23                                                       HYS02110
        END IF                                                           HYS02120
        IF(KKL.EQ.7) THEN                                                HYS02130
          KKL=4                                                          HYS02140
          GO TO 23                                                       HYS02150
        END IF                                                           HYS02160
        IF(KKL.EQ.10) THEN                                               HYS02170
          KKL=2                                                          HYS02180
          GO TO 23                                                       HYS02190
        END IF                                                           HYS02200
        IF(KKL.EQ.9) THEN                                                HYS02210
          KKL=4                                                          HYS02220
          GO TO 23                                                       HYS02230
        END IF                                                           HYS02240
        IF(KKL.EQ.5) THEN                                                HYS02250
          KKL=10                                                         HYS02260
          PSEOLD=AREA1                                                   HYS02270
          GO TO 13                                                       HYS02280
        END IF                                                           HYS02290
        IF(KKL.EQ.6) THEN                                                HYS02300
          KKL=9                                                          HYS02310
          PSEOLD=AREA1                                                   HYS02320
          GO TO 13                                                       HYS02330
        END IF                                                           HYS02340
C                                                                        HYS02350
        IF(KEYDS2.LT. 0.) THEN                                           HYS02360
          IF((ABS(KEYDS2).LT.DC).AND.(KKL.EQ.5)) KKL=4                   HYS02370
        END IF                                                           HYS02380
        IF(KEYDS2.GT. 0.) THEN                                           HYS02390
          IF((KEYDS2.LT.DC).AND.(KKL.EQ.6)) KKL=2                        HYS02400
        END IF                                                           HYS02410
      END IF                                                             HYS02420
C                                                                        HYS02430
   13 IF((KKL.EQ.10).AND.((DSINC*DSINCL).LT.0)) THEN                     HYS02440
        IF(POSTFG) THEN                                                  HYS02450
          FFFF=FU0(1)                                                    HYS02460
          DDDD=DU0(1)                                                    HYS02470
        ELSE                                                             HYS02480
          FFFF=FU0(2)                                                    HYS02490
          DDDD=DU0(2)                                                    HYS02500
        END IF                                                           HYS02510
        FP1=FFFF*0.9                                                     HYS02520
        DP1=DDDD                                                         HYS02530
        KP=(FP1-0.)/(DP1-DPP)                                            HYS02540
        K=KP                                                             HYS02550
        USE=MAX(K,SOB)                                                   HYS02560
          IF(POSTFG) CYCLEN=CYCLEN+1                                     HYS02570
        IOPT2=3                                                          HYS02580
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DPP,0.,AREA1,           HYS02590
     &              K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FU00,LRULE)    HYS02600
        FP=0.+(DP-DPP)*KP                                                HYS02610
        FPP=0.                                                           HYS02620
        LRULE=51                                                         HYS02630
        CALL NEWSTI(MQD,ODC,ODY,ODU,OFC,OFY,OFU,AREA1,DM,FC,FY,          HYS02640
     &              FU,FP,DC,DY,DU,DF,HO,H,LWP,OSOC,SRFLAG,OSCY,SOC,     HYS02650
     &              SCY,STU,SUF,SR,SOB,SOE,SQ6,DPP,CO1,PSQ6,DPT,         HYS02660
     &              SDP,OSUF)                                            HYS02670
        CALL DINTER(FC,FY,FU,FF,FPP,DC,DY,DU,DF,DPP,SOC,SCY,STU,SUF,K,   HYS02680
     &              FP,DP,DSINC,KKL,LRULE,POSTFG,DU0,FU0)                HYS02690
        DMAX=ABS(DP)                                                     HYS02700
        DSINCL=DSINC                                                     HYS02710
        RETURN                                                           HYS02720
      END IF                                                             HYS02730
      IF(KKL.EQ.10) THEN                                                 HYS02740
        IF(POSTFG) THEN                                                  HYS02750
          DDKK=DU0(1)                                                    HYS02760
        ELSE                                                             HYS02770
          DDKK=DU0(2)                                                    HYS02780
        END IF                                                           HYS02790
        IF(ABS(DP).LT.ABS(DDKK)) THEN                                    HYS02800
          LRULE=52                                                       HYS02810
          CALL DINTER(FC,FY,FU,FF,FPP,DC,DY,DU,DF,DPP,SOC,SCY,STU,SUF,   HYS02820
     &                K,FP,DP,DSINC,KKL,LRULE,POSTFG,DU0,FU0)          HYS02830
        END IF                                                           HYS02840
        IF(ABS(DP).GT.ABS(DDKK)) THEN                                    HYS02850
          K=KP*0.5                                                       HYS02860
          IF(K.NE.KLAST) THEN                                            HYS02870
            IF(POSTFG) FP=FPP+(DU0(1)-DPP)*KLAST+(DP-DU0(1))*K           HYS02880
            IF(.NOT.POSTFG) FP=FPP+(DU0(2)-DPP)*KLAST+(DP-DU0(2))*K      HYS02890
          END IF                                                         HYS02900
          LRULE=53                                                       HYS02910
          CALL DINTER(FC,FY,FU,FF,FPP,DC,DY,DU,DF,DPP,SOC,SCY,STU,SUF,   HYS02920
     &                K,FP,DP,DSINC,KKL,LRULE,POSTFG,DU0,FU0)          HYS02930
        END IF                                                           HYS02940
        DSINCL=DSINC                                                     HYS02950
        DMAX=MAX(ABS(DP),DMAX)                                           HYS02960
        IOPT2=1                                                          HYS02970
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DP,FP,AREA1,            HYS02980
     &              K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FU00,LRULE)    HYS02990
        RETURN                                                           HYS03000
      END IF                                                             HYS03010
C                                                                        HYS03020
      IF((KKL.EQ.9).AND.((DSINC*DSINCL).LT.0)) THEN                      HYS03030
        IF(POSTFG) THEN                                                  HYS03040
          FFFF=FU0(2)                                                    HYS03050
          DDDD=DU0(2)                                                    HYS03060
        ELSE                                                             HYS03070
          FFFF=FU0(1)                                                    HYS03080
          DDDD=DU0(1)                                                    HYS03090
        END IF                                                           HYS03100
        FP1=FFFF*0.9                                                     HYS03110
        DP1=DDDD                                                         HYS03120
        KP=(FP1-0.)/(DP1-DPP)                                            HYS03130
        K=KP                                                             HYS03140
        USE=MAX(K,SOB)                                                   HYS03150
        IF(.NOT.POSTFG) CYCLEN=CYCLEN+1                                  HYS03160
        IOPT2=3                                                          HYS03170
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DPP,0.,AREA1,           HYS03180
     &              K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FU00,LRULE)    HYS03190
        FP=0.+(DP-DPP)*KP                                                HYS03200
        FPP=0.                                                           HYS03210
        LRULE=51                                                         HYS03220
        CALL NEWSTI(MQD,ODC,ODY,ODU,OFC,OFY,OFU,AREA1,DM,FC,FY,          HYS03230
     &              FU,FP,DC,DY,DU,DF,HO,H,LWP,OSOC,SRFLAG,OSCY,SOC,     HYS03240
     &              SCY,STU,SUF,SR,SOB,SOE,SQ6,DPP,CO1,PSQ6,DPT,         HYS03250
     &              SDP,OSUF)                                            HYS03260
        CALL DINTER(FC,FY,FU,FF,FPP,DC,DY,DU,DF,DPP,SOC,SCY,STU,SUF,K,   HYS03270
     &              FP,DP,DSINC,KKL,LRULE,POSTFG,DU0,FU0)                HYS03280
        DMAX=ABS(DP)                                                     HYS03290
        DSINCL=DSINC                                                     HYS03300
        RETURN                                                           HYS03310
      END IF                                                             HYS03320
      IF(KKL.EQ.9) THEN                                                  HYS03330
        IF(POSTFG) THEN                                                  HYS03340
          DDKK=DU0(2)                                                    HYS03350
        ELSE                                                             HYS03360
          DDKK=DU0(1)                                                    HYS03370
        END IF                                                           HYS03380
        IF(ABS(DP).LT.ABS(DDKK)) THEN                                    HYS03390
          LRULE=52                                                       HYS03400
          CALL DINTER(FC,FY,FU,FF,FPP,DC,DY,DU,DF,DPP,SOC,SCY,STU,SUF,   HYS03410
     &                K,FP,DP,DSINC,KKL,LRULE,POSTFG,DU0,FU0)          HYS03420
        END IF                                                           HYS03430
        IF(ABS(DP).GT.ABS(DDKK)) THEN                                    HYS03440
          K=KP*0.5                                                       HYS03450
          IF(K.NE.KLAST) THEN                                            HYS03460
```

```
            IF(POSTFG)  FP=FPP+(DUO(2)-DPP)*KLAST+(DP-DUO(2))*K
            IF(.NOT.POSTFG)  FP=FPP+(DUO(1)-DPP)*KLAST+(DP-DUO(1))*K
          END IF
          KRULE=53
          CALL DINTER(FC,FY,FU,FF,FPP,DC,DY,DU,DF,DPP,SOC,SCY,SYU,SUF,
     &              K,FP,DP,DSINC,KKL,LRULE,POSTFG,DUO,FUO)
        END IF
        DMAX=MAX(ABS(DP),DMAX)
        DSINCL=DSINC
        IOPT2=1
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DP,FP,AREA1,
     &              K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FUO0,LRULE)
        RETURN
      END IF
C----- CHECK IF PASS TURNING POINT
   23 CALL DCURRD(KKL,KEYDS2,DSINC,FP,FPP,DM,FC,FY,FU,FF,
     &            K,DP,DMAX,DPP,DC,DY,DU,CYCLEN,AREA1,FMAX,TNPTFG,
     &            LRULE,MQD,ODC,ODY,ODU,OFU,DF,NO,H,LWP,OSOC,POSTFG,
     &            OSCY,SOC,SCY,SYU,SUF,SR,SOB,SOE,DP20,SQ6,OFC,OFY,
     &            SRFLAG,CO1,PSQ6,KLAST,DPT,DPTT,KKLLST,USE,
     &            OSUF,PSEOLD,UPOS,UNEG,ESE,PSE,CSE,EXCR,IOPT,IOPT2)
C
      IF((K.EQ.0).AND.(FP.EQ.0).AND.(FP.EQ.FP)) THEN
        GO TO 77
      END IF
C
C -- TO CALCULATE THE STIFFNESS IN EACHLOAD STEP
      CALL DNCKUL(FC,FY,FU,FF,DC,DY,DU,DF,FP,DP,DMAX,DM,FM,KP,
     &            SOC,SCY,SOY,SYU,SUF,SCU,SOB,SOE,KKL,LRULE,FPP,DPP,
     &            FUO,DUO,DSINC,KEYDS1,K,SZY,DU17,FU17,CYCLEN,OSOB,
     &            AREA1,HS,WID,AV,AD,FU00,DU00,POSTFG,PSEOLD,PSE,ESE,
     &            DKP,HO,H,RAT,RAT1,OFC,OFY,OFU,ODC,ODY,ODU,
     &            SQ6,OSOC,OSCY,OSYU,OSUF,MQD,LWP,SR,DP20,KEYDS2,
     &            SRFLAG,CO1,PSQ6,KLAST,DPTT,DPT,FPT,TNPTFG,
     &            KKLLST,LRULST,DSINCL,IOPT2,USE)
C
      IF(K.EQ.0.) RETURN
   77 IF(TNPTFG) THEN
        IF(KKL.EQ.1 .OR. KKL.EQ.8 .OR. KKL.EQ.10 .OR. KKL.EQ.3
     &     .OR. KKL.EQ.7 .OR. KKL.EQ.9 .OR. KKL.EQ.5
     &     .OR. KKL.EQ.6) THEN
          IF(ABS(FP).LE.(0.7*FY) .AND. KKL.NE.1 .AND. KKL.NE.3) THEN
            USE=K
            IOPT2=3
          END IF
          IF(ABS(FP).GT.(0.7*FY)) IOPT2=1
          CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPT,FPT,DP,FP,AREA1,
     &                K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FUO0,
     &                LRULE)
        ELSE IF(KKL.EQ.2 .OR. KKL.EQ.4) THEN
          IOPT2=3
          CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPT,FPT,DP,FP,AREA1,
     &                K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FUO0,
     &                LRULE)
        END IF
        TNPTFG=.FALSE.
       ELSE IF((KKL.NE.5 .AND. KKL.NE.6).OR.
     &        ((KKL.EQ.5 .OR. KKL.EQ.6) .AND. ((DP*FP).GT.0.))) THEN
        IF(ABS(FP).LE.(0.7*FY) .AND. KKL.NE.1 .AND. KKL.NE.3) THEN
          USE=K
          IOPT2=3
        END IF
        IF(ABS(FP).GT.(0.7*FY)) IOPT2=1
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DP,FP,AREA1,
     &              K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FUO0,LRULE)
       ELSE IF((KKL.EQ.5 .OR. KKL.EQ.6) .AND.((DP*FP).LT.0.)) THEN
        USE=1E25
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DP,FP,AREA1,
     &              K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FUO0,LRULE)
      END IF
      DSINCL=DSINC
C CALC DUCTILITY
      IF(KKL.EQ.1  .OR. KKL.EQ.8 .OR. KKL.EQ.10) THEN
        DLI=MAX(DP,DPP)
        CALL DNE(1,UPOS,EXCR,ODY,DLI,ESE(2),PSE,PSEOLD,CSE)
      END IF
      IF(KKL.EQ.3 .OR. KKL.EQ.7 .OR. KKL.EQ.9 ) THEN
        DLI=MIN(DP,DPP)
        CALL DNE(1,UNEG,EXCR,-ODY,DLI,ESE(3),PSE,PSEOLD,CSE)
      END IF
   66 RETURN
      END
C
C*********************************************************************
C       DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE
C       END DEBUG
C*********************************************************************
C== SUBROUTINE DNCKUL  SUBRTN (FIND STIFFNESS AND FORCE IN EACH STEP) ==
C*********************************************************************
      SUBROUTINE DNCKUL(FC,FY,FU,FF,DC,DY,DU,DF,FP,DP,DMAX,DM,FM,KP,
     &            SOC,SCY,SOY,SYU,SUF,SCU,SOB,SOE,KKL,LRULE,FPP,DPP,
     &            FUO,DUO,DSINC,KEYDS1,K,SZY,DU17,FU17,CYCLEN,OSOB,
     &            AREA1,HS,WID,AV,AD,FU00,DU00,POSTFG,PSEOLD,PSE,ESE,
     &            DKP,HO,H,RAT,RAT1,OFC,OFY,OFU,ODC,ODY,ODU,
     &            SQ6,OSOC,OSCY,OSYU,OSUF,MQD,LWP,SR,DP20,KEYDS2,
     &            SRFLAG,CO1,PSQ6,KLAST,DPTT,DPT,FPT,TNPTFG,
     &            KKLLST,LRULST,DSINCL,IOPT2,USE)
C
      REAL K,DUO(2),FUO(2),KP,DSINC,MQD,LWP,KLAST,KEYDS1,KEYDS2
      INTEGER CYCLEN
      DIMENSION ESE(3)
      LOGICAL SRFLAG,TNPTFG,POSTFG
C 1. .....LOAD ..................................................RULE 1.0
      IF((KKL.EQ.1).OR.(KKL.EQ.3)) THEN
        LRULE=10
        IF(ABS(DP).LE.DC) THEN
          K=SOC
          USE=K
          IOPT2=3
          RETURN
        ELSE IF((ABS(DP).GT.DC).AND.(ABS(DP).LE.DY)) THEN
          DDT=SIGN(DC,DP)
          K=SCY
          IF(ABS(FP).LE.(0.7*FY)) IOPT2=2
          IF(ABS(FP).GT.(0.7*FY)) IOPT2=1
        ELSE IF((ABS(DP).GT.DY).AND.(ABS(DP).LE.DU)) THEN
          DDT=SIGN(DY,DP)
          K=SYU
          IOPT2=1
        ELSE IF(ABS(DP).GT.DU .AND. ABS(DP).LE.DF) THEN
          DDT=SIGN(DU,DP)
          K=SUF
          IOPT2=1
        ELSE IF(ABS(DP).GT.DF .AND. ABS(DP).LE.(20*DY)) THEN
          DDT=SIGN(DF,DP)
          K=((0.1*FU)-FF)/((20*DY)-DF)
          USE=OSOC
        ELSE IF(ABS(DP).GT.(20*DY)) THEN
          DDT=SIGN(20*DY,DP)
          K=-0.0001*SCY
          USE=OSOC
        END IF
        IF(KLAST.EQ.K) RETURN
        IF(TNPTFG) THEN
          FPPP=FPP
          FP=FPPP+(DDT-DPP)*KLAST+(DP-DDT)*K
          FPT=FPPP+(DDT-DPP)*KLAST
          DPT=DDT
        END IF
        RETURN
      END IF
C 2. .....UNLOAD ....
      IF((KKL.EQ.2).OR.(KKL.EQ.4)) THEN
        IOPT2=3
        IF(DM.LT.ODC .AND. DMAX.LT.ODC) THEN
C .....................................................RULE 2.1
          LRULE=21
          K=SOC
          IF(KKLLST.EQ.5 .OR. KKLLST.EQ.6 .OR. KKLLST.EQ.7
     &       .OR. KKLLST.EQ.8) THEN
            DP1=DP
            FP1=FP
            CALL DDISSI(3,ESE,PSE,PSEOLD,DPP,FPP,DPT,0. ,AREA1,
     &                  K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FUO0,
     &                  LRULE)
            FP=0.+K*DPTT
            DPTT=0.
            DPP=DPT
            FPP=0.
            DP=DP1
            FP=FP1
          END IF
          IOPT2=3
          RETURN
        END IF
C
        IF(LRULE.EQ.54 .OR. LRULE.EQ.23)   THEN
          K=KLAST
          RETURN
        END IF
C
        IF(KKLLST.EQ.5 .OR. KKLLST.EQ.6 .OR. KKLLST.EQ.7 .OR.
     &     KKLLST.EQ.8 .OR. KKLLST.EQ.9 .OR. KKLLST.EQ.10) THEN
          SK1=(0.-FPP)/(DPP/3-DPP)
          K=SK1
          IF((DM.GE.ODC).AND.(ABS(KEYDS2).LE.DC)) THEN
C ......
            IF(KKLLST.EQ.7 .OR. KKLLST.EQ.8 .OR.
     &        KKLLST.EQ.9 .OR. KKLLST.EQ.10) THEN
              K=MAX(KP,SK1)
            END IF
            FP=FPP+(DP-DPP)*K
C .....................................................RULE 5.4
            LRULE=54
            IF((FP*FPP).LT.0) THEN
              IF(FPP.GT.0) K=FC/(DC-DPT)
              IF(FPP.LT.0) K=-FC/(-DC-DPP)
              FP=0.+K*DPTT
              IF(KKL.EQ.2) KKL=5
              IF(KKL.EQ.4) KKL=6
            END IF
            RETURN
          END IF
          IF((ABS(FPP).LE.(0.7*FY)).AND.((KKLLST.EQ.7 .OR.
     &      KKLLST.EQ.8 .OR. KKLLST.EQ.9 .OR. KKLLST.EQ.10)) THEN
            K=MAX(KLAST,OSOB)
            K=MAX(K,SK1)
C .....................................................RULE 2.3
            LRULE=23
            USE=K
            FP=FPP+(DP-DPP)*K
            IF((FP*FPP).LT.0.) THEN
              DPT=DPP+(0.-FPP)/K
              CALL NEWSTI(MQD,ODC,ODY,ODU,OFC,OFY,OFU,AREA1,DM,FC,FY,
     &                    FU,FF,DC,DY,DU,DF,HO,H,LWP,OSOC,SRFLAG,OSCY,
     &                    SOC,SCY,SYU,SUF,SR,SOB,SOE,SQ6,DPP,CO1,PSQ6,
     &                    DPT,SDP,OSUF)
              IF(SQ6.LE.0.3 .AND. SQ6.GT.0.) THEN
                DJP=SIGN(DC,-DPT)
                FJP=SIGN(FC,DJP)
                K=FJP/(DJP-DPT)
              ELSE
                K=SR
              END IF
              IOPT2=3
              IF(KKL.EQ.2) KKL=5
              IF(KKL.EQ.4) KKL=6
              DPP=DPT
              FPP=0.
              FP=0.+(DP-DPT)*K
              RETURN
            END IF
            IOPT2=3
            RETURN
          END IF
        END IF
        IF(KKL.NE.KKLLST) THEN
          IF(POSTFG) THEN
            FFFF=FUO(1)
            GGGG=FUO(2)
          ELSE
            FFFF=FUO(2)
            GGGG=FUO(1)
          END IF
          IF(DSINC.LT.0.) THEN
            FUO0=FFFF
            FU09=0.85*FFFF
            FU17=FFFF/6
          END IF
          IF(DSINC.GT.0.) THEN
            FU00=GGGG
            FU09=0.85*GGGG
            FU17=GGGG/5
          END IF
        END IF
        IF((FP*FPP).LT.0) THEN
          IF(FPP.GT.0) K=FC/(DC-DPT)
          IF(FPP.LT.0) K=-FC/(-DC-DPP)
          FP=0.+K*DPTT
          IF(KKL.EQ.2) KKL=5
          IF(KKL.EQ.4) KKL=6
        END IF
        IF((KKLLST.EQ.7 .OR. KKLLST.EQ.8 .OR. KKLLST.EQ.9
     &    .OR. KKLLST.EQ.10).AND.(ABS(FPP).LT.ABS(FUO0))) THEN
          FU09=FPP*0.85
          FU17=FPP/6
        END IF
        IF(ABS(FU17*6).LE.(0.70*FY)) THEN
          IF((ABS(FP).GT.FC/3).AND.(ABS(FP).LE.(0.70*FY))) THEN
C .....................................................RULE 2.2
            LRULE=22
            K=OSOC
            IF(K.NE.KLAST) THEN
              FP=FPP+(DP-DPP)*K
            END IF
          END IF
          IF(ABS(FP).LE.FC/3) THEN
C .....................................................RULE 2.3
            LRULE=23
            IF(LRULE.NE.LRULST) TNPTFG=.TRUE.
            IF(TNPTFG) THEN
              IF(ABS(DP).GT.(20*DY)) THEN
                K=SOC
              ELSE
                FPT=SIGN(FC/3,FPP)
                DPT=DPP+(FPT-FPP)/KLAST
                DCP=SIGN(DC,-DPP)
                FCP=SIGN(FC,DCP)
                K=(FCP-FPT)/(DCP-DPT)
                IF(K.GT.OSOC) THEN
                  FCP=0
                  DCP=DPP/3
                  K=(0.-FPT)/(DCP-DPT)
                  USE=K
```

```
                END IF
                FPT=FPP+(DPT-DPP)*KLAST
                FP=FPP+(DPT-DPP)*KLAST+(DP-DPT)*K
              END IF
              RETURN
            END IF
          END IF
          RETURN
        ELSE
          IF((ABS(FP).GT.ABS(FU17)).AND.(ABS(FP).LE.ABS(FU17*6)
     &      .OR. KLAST.LT.0.)) THEN
            FPP=FU09
C .....................................................RULE 2.5
            LRULE=25
            IF(LRULE.NE.LRULST) THEN
              DDPP=SIGN(DC,-DPP)
              FPPP=SIGN(FC,DPPP)
              SKK1=(FPPP-FPP)/(DPPP-DPP)
              K=MAX(SKK1,OSOB)
              FP=FPP+(DP-DPP)*K
            END IF
          END IF
          IF((ABS(ABS(FP)-ABS(FU17)).LT.ABS(K*DSINC)).AND.
     &      (ABS(FP).LT.ABS(FU17))) THEN
            FPT=FU17
            DPT=DPP+(FPT-FPP)/K
            TNPTFG=.TRUE.
          END IF
          IF((TNPTFG).OR.(ABS(FP).LE.ABS(FU17))) THEN
C .....................................................RULE 2.6
            LRULE=26
            IF(TNPTFG) THEN
              FTP=SIGN(FT,-DP)
              DTP=SIGN(DT,-DP)
              K=(FTP-FPT)/(DTP-DPT)
              FP=FPP+(DPT-DPP)*KLAST+(DP-DPT)*K
              FPT=FPP+(DPT-DPP)*KLAST
              IF((FP*FPP).LT.0.) THEN
                DPT=DPP+(0.-FPP)/K
                CALL NEWSTI(MQD,ODC,ODT,ODU,OFC,OFT,OFU,AREA1,DM,
     &                      FC,FT,FU,FF,DC,DT,DU,DF,HO,H,LWP,OSOC,
     &                      SRFLAG,OSCT,SOC,SCT,STU,SUF,SR,SOB,SOE,
     &                      SQ6,DPP,CO1,PSQ6,DPT,SDP,OSUF)
                IF(SQ6.LE.0.3 .AND. SQ6.GT.0.) THEN
                  DJP=SIGN(DC,-DPT)
                  FJP=SIGN(FC,DJP)
                  K=FJP/(DJP-DPT)
                ELSE
                  K=6R
                END IF
                IOPT2=3
                IF(KKL.EQ.2) KKL=5
                IF(KKL.EQ.4) KKL=6
                DPP=DPT
                FPP=0.
                FP=0.+(DP-DPT)*K
                RETURN
              END IF
            END IF
            RETURN
          END IF
          RETURN
        END IF
      END IF
C 3. ......REVERSAL LOADING .....
      IF((KKL.EQ.5).OR.(KKL.EQ.6)) THEN
        IF((SQ6.LE.CO1).OR.(LRULE.EQ.31)) THEN
          LRULE=31
          IF(DM.LT.DC .AND. DMAX.LT.DC) THEN
            K=KLAST
            PSE=0.
            DPP=0.
            FPP=0.
            USE=K
            IOPT2=3
            IF(KKL.EQ.5) KKL=3
            IF(KKL.EQ.6) KKL=1
            RETURN
          END IF
          IF(KKL.NE.KKLLST) THEN
            DJP=SIGN(DC,-DPT)
            FJP=SIGN(FC,DJP)
            IF(DPTT.NE.0) THEN
              K=FJP/(DJP-DPT)
            ELSE
              K=FJP/(DJP-DPP)
            END IF
            IF(DPTT.NE.0) THEN
              IOPT2=3
              CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DPT,0.,AREA1,
     &                    K,USE,KKL,FC,DC,FT,DT,DM,DMAX,ODC,SOB,FU00,
     &                    LRULE)
              FP=0.+K*DPTT
              DPTT=0.
              DPP=DPT
              FPP=0.
            END IF
          END IF
          IF((DP*FP).GT.0.) THEN
            USE=(0.-FP)/(DP/3-DP)
          END IF
          IOPT2=3
          RETURN
        END IF
C .....................................................RULE 3.2
        LRULE=32
        IF((FP*FPP).LT.0.) THEN
          K=SR
        END IF
        IF(DPTT.NE.0) THEN
          IOPT2=3
          CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DPT,0. ,AREA1,
     &                K,USE,KKL,FC,DC,FT,DT,DM,DMAX,ODC,SOB,FU00,
     &                LRULE)
          FP=0.+K*DPTT
          DPTT=0.
          DPP=DPT
          FPP=0.
        END IF
        IF((DP*FP).GT.0.) THEN
          USE=(0.-FP)/(DP/3-DP)
        END IF
        IOPT2=3
        RETURN
      END IF
C4.....RELOADING AFTER UNLOADING.....
      IF((KKL.EQ.7).OR.(KKL.EQ.8)) THEN
        IF(DM.LT.ODC .AND. DMAX.LT.ODC) THEN
          K=KLAST
          USE=K
          IOPT2=3
          IF(KKL.EQ.7) KKL=3
          IF(KKL.EQ.8) KKL=1
          RETURN
        END IF
        IF(KKL.NE.KKLLST) THEN
          IF(POSTFG) THEN
            FFFF=FU0(1)
            GGGG=FU0(2)
            DDDD=DU0(1)
            EEEE=DU0(2)
          ELSE
            FFFF=FU0(2)
            DDDD=DU0(2)
            GGGG=FU0(1)
            EEEE=DU0(1)
          END IF
          IF(DSINC.GT.0) THEN
            FU09=0.9*FFFF
            IF(ABS(FFFF).LT.0.7*FT) DPT=DDDD-(FFFF-FU09)/OSOC
            IF(ABS(FFFF).GE.0.7*FT) DPT=DDDD
          END IF
          IF(DSINC.LT.0) THEN
            FU09=0.9*GGGG
            IF(ABS(GGGG).LT.0.7*FT) DPT=EEEE-(GGGG-FU09)/OSOC
            IF(ABS(GGGG).GE.0.7*FT) DPT=EEEE
          END IF
          IF(LRULE.EQ.54) THEN
            KP=KLAST
          ELSE
            SKP=(FU09-FPP)/(DPT-DPP)
            KP=MAX(SKP,KLAST)
          END IF
          TNPTFG=.TRUE.
        END IF
        IF(ABS(DP).LE.ABS(DPT)) THEN
          LRULE=41
          K=KP
          USE=K
          IOPT2=3
          IF(TNPTFG) THEN
            FP=FPP+(DP-DPP)*K
          END IF
        ELSE
          IF(LRULST.EQ.42) GO TO 688
          IF(LRULST.EQ.54) THEN
            KP=KLAST
          END IF
          IF(LRULST.NE.41) THEN
            K=MAX(0.5*KP,KLAST)
            FP=FPP+(DP-DPP)*K
            IOPT2=1
            DPP=DPT
            FPP=FPT
            LRULE=42
            TNPTFG=.FALSE.
            GO TO 688
          END IF
C .....................................................RULE 4.2
          LRULE=42
          K=0.5*KP
          IOPT2=1
          IF(K.NE.KLAST) TNPTFG=.TRUE.
          IF(TNPTFG) THEN
            FP=FPP+(DPT-DPP)*KLAST+(DP-DPT)*K
            FPT=FPP+(DPT-DPP)*KLAST
          END IF
        END IF
  688   CALL DINTER(FC,FT,FU,FF,FPP,DC,DT,DU,DP,DPP,SOC,SCT,STU,SUF,
     &              K,FP,DP,DSINC,KKL,LRULE,POSTFG,DU0,FU0)
        IF(DM.GE.ODC .AND. ABS(KEYDS2).LE.DC) THEN
          SK1=(0.-FP)/(DP/3-DP)
          USE=MAX(KP,SK1)
          IOPT2=3
        ELSE IF(ABS(FP).LE.(0.7*FT)) THEN
          SK1=(0.-FP)/(DP/3-DP)
          USE=MAX(KLAST,OSOB)
          USE=MAX(USE,SK1)
          IOPT2=3
        ELSE IF(ABS(FP).GT.(0.7*FT)) THEN
          IOPT2=1
        END IF
C
        RETURN
      END IF
   99 FORMAT(2F11.5,I6,F9.4,I3)
  199 FORMAT(2F11.5,I4,I5)
  299 FORMAT(5F12.4)
      RETURN
      END
C*********************************************************************
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE
C         END DEBUG
C*********************************************************************
C ==  SUBROUTINE DCURRD SUBRTN (TO COMPUTE CURRENT DISPL,DM,DMAX)
C *********************************************************************
      SUBROUTINE DCURRD(KKL,KEYDS2,DSINC,FF,FPP,DM,FC,FT,FU,FF,
     &           K,DP,DMAX,DPP,DC,DT,DU,CYCLEN,AREA1,FMAX,TNPTFG,
     &           LRULE,MQD,ODC,ODT,ODU,OFU,DF,HO,H,LWP,OSOC,POSTFG,
     &           OSCT,SOC,SCT,STU,SUF,SR,SOB,SOE,DP20,SQ6,OFC,OFT,
     &           SRFLAG,CO1,PSQ6,KLAST,DPT,DPTT,KKLLST,USE,
     &           OSUF,PSEOLD,UPOS,UNEG,ESE,PSE,CSE,EXCR,IOPT,IOPT2)
C
C DPP ----DISPL BEFORE DISPL INCREMENT ADDED
      LOGICAL SRFLAG,TNPTFG,POSTFG
      INTEGER CYCLEN
      DIMENSION ESE(3),EXCR(6)
      REAL KEYDS2,DSINC,DU0(2),KLAST,K,LWP,MQD,K12,UNEG(3),UPOS(3)
      IF((KKLLST.EQ.1 .AND. DSINC.GT.0.) .OR.
     &  (KKLLST.EQ.3 .AND. DSINC.LT.0.)) THEN
        IF((ABS(DP)-DC).LT.ABS(DSINC)) THEN
          IF(ABS(DP).GT.DC) THEN
            TNPTFG=.TRUE.
          END IF
        END IF
        IF((ABS(DP)-DT).LT.ABS(DSINC)) THEN
          IF(ABS(DP).GT.DT) THEN
            TNPTFG=.TRUE.
          END IF
        END IF
        IF((ABS(DP)-DU).LT.ABS(DSINC)) THEN
          IF(ABS(DP).GT.DU) THEN
            TNPTFG=.TRUE.
          END IF
        END IF
        IF((ABS(DP)-DF).LT.ABS(DSINC)) THEN
          IF(ABS(DP).GT.DF) THEN
            TNPTFG=.TRUE.
          END IF
        END IF
        IF((ABS(DP)-(20*DT)).LT.ABS(DSINC)) THEN
          IF(ABS(DP).GT.(20*DT)) THEN
            TNPTFG=.TRUE.
          END IF
        END IF
        GO TO 131
      END IF
      IF((KKL.EQ.5 .OR. KKL.EQ.6) .AND. ((DP*FP).GT.0.)) THEN
        IF((ABS(DP)-DC).LT.ABS(DSINC)) THEN
          IF(ABS(DP).GT.DC) THEN
            TNPTFG=.TRUE.
            IF(KKL.EQ.5) KKL=3
            IF(KKL.EQ.6) KKL=1
          END IF
        END IF
        GO TO 131
      END IF
  131 DMAX=MAX(ABS(DP),DMAX)
C
      IF((FP.EQ.0).OR.((FP*FPP).LT.0)) THEN
        DM=MAX(DM,DMAX)
C DPTT,DPT      (0,DPT)     |(0,0)
C---------------*----------|------------
```

```
C       *    ->|DPTT |<- |                                                  HYS09150
C     ->|   DSINC   |<- |                                                  HYS09160
C                                                                           HYS09170
        IF((FP*FPP).LT. 0.) THEN                                            HYS09180
          DPT=DPP-FPP/K                                                     HYS09190
          DPTT=DP-DPT                                                       HYS09200
          CALL DNE(0,UPOS,EXCR,ODT,DP,ESE,PSE,PSEOLD,CSE)                   HYS09210
C ----- IF K=SUF AND REACH FORCE=0 THEN SET FORCE=0 AND DP=DPTT             HYS09220
C       ,AFTER THAT K=0                                                     HYS09230
          IF(K.EQ.(-0.0001*SCY)) THEN                                       HYS09240
            DP=DP                                                           HYS09250
            FP=0.                                                           HYS09260
            K=0.                                                            HYS09270
            STOP                                                            HYS09280
          END IF                                                            HYS09290
          IF(DM.LT.ODC .AND. DMAX.LT.ODC) THEN                              HYS09300
            IOPT2=3                                                         HYS09310
          ELSE                                                              HYS09320
            IOPT2=2                                                         HYS09330
          END IF                                                            HYS09340
C                                                                           HYS09350
          CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DPT,0. ,AREA1,           HYS09360
     &                K,USE,KKL,FC,DC,FY,DT,DM,DMAX,ODC,SOB,FU00,LRULE)     HYS09370
          IF((POSTFG).AND.(FPP.LT.0)) THEN                                  HYS09380
            CYCLEN=CYCLEN+1                                                 HYS09390
            FMAX=0.                                                         HYS09400
            DMAX=ABS(DP)                                                    HYS09410
          END IF                                                            HYS09420
          IF(.NOT. POSTFG .AND. FPP.GT.0 ) THEN                             HYS09430
            CYCLEN=CYCLEN+1                                                 HYS09440
            FMAX=0.                                                         HYS09450
            DMAX=ABS(DP)                                                    HYS09460
          END IF                                                            HYS09470
        END IF                                                              HYS09480
        IF(DM.GT.ODC) THEN                                                  HYS09490
          CALL NEWSTI(MQD,ODC,ODT,ODU,OFC,OFT,OFU,AREA1,DM,FC,FY,           HYS09500
     &                FU,FF,DC,DT,DU,DF,HO,H,LWP,OSOC,SRFLAG,OSCY,SOC,     HYS09510
     &                SCY,STU,SUF,SR,SOB,SOE,SQ6,DPP,CO1,PSQ6,DPT,         HYS09520
     &                SDP,OSUF)                                            HYS09530
        END IF                                                              HYS09540
        IF(KKL.EQ.2) KKL=5                                                  HYS09550
        IF(KKL.EQ.4) KKL=6                                                  HYS09560
      END IF                                                                HYS09570
      IF(((KKL.EQ.7).AND.(KKLLST.EQ.4)).OR.                                 HYS09580
     & ((KKL.EQ.8).AND.(KKLLST.EQ.2))) THEN                                 HYS09590
        CYCLEN=CYCLEN+1                                                     HYS09600
        DM=MAX(DM,DMAX)                                                     HYS09610
      END IF                                                                HYS09620
      RETURN                                                                HYS09630
      END                                                                   HYS09640
C-----------------------------------------------------------------------    HYS09650
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                          HYS09660
C         END DEBUG                                                         HYS09670
C-----------------------------------------------------------------------    HYS09680
C ==  NEWSTIF SUBRTN (FIND STIFFNESS WHEN PASSING HORIZ. AXIS (FORCE=0)     HYS09690
C ***********************************************************************  HYS09700
      SUBROUTINE NEWSTI(MQD,ODC,ODT,ODU,OFC,OFT,OFU,AREA1,DM,FC,FY,         HYS09710
     &            FU,FF,DC,DT,DU,DF,HO,H,LWP,OSOC,SRFLAG,OSCY,SOC,          HYS09720
     &            SCY,STU,SUF,SR,SOB,SOE,SQ6,DPP,CO1,PSQ6,DPT,              HYS09730
     &            SDP,OSUF)                                                 HYS09740
C                                                                           HYS09750
      REAL MQD,LWP                                                          HYS09760
      LOGICAL SRFLAG                                                        HYS09770
C                                                                           HYS09780
      SQ6=AREA1/(OFU*ODU)                                                   HYS09790
      TOL=1E-10                                                             HYS09800
      IF(ABS(DM).LT.ODC .AND. SQ6.LT.TOL) SQ6=0.                            HYS09810
      PSDP=SDP                                                              HYS09820
      SDP=DPT                                                               HYS09830
      ALPHA=HO/H                                                            HYS09840
      IF((SQ6.GT.0.).AND.(SQ6.LE.0.3)) THEN                                 HYS09850
        DC=DC                                                               HYS09860
        FC=FC                                                               HYS09870
      ELSE                                                                  HYS09880
        B=-0.25473*SQ6                                                      HYS09890
        SR=(0.27798*10**B)*OSOC                                             HYS09900
        DC=ABS(SDP)*0.25                                                    HYS09910
        DC=MAX(DC,ODC)                                                      HYS09920
        IF(DC.GT.(1.5*ODC)) THEN                                            HYS09930
          DC=1.5*ODC                                                        HYS09940
        END IF                                                              HYS09950
        IF(ABS(SDP).GT.DU) THEN                                             HYS09960
          FC=0.98*FC                                                        HYS09970
          SR=FC/(ABS(SDP)+DC)                                               HYS09980
        END IF                                                              HYS09990
C                                                                           HYS10000
        FC=SR*(ABS(SDP)+DC)                                                 HYS10010
      END IF                                                                HYS10020
      IF(SQ6.NE.0.) THEN                                                    HYS10030
        B=-0.20102*SQ6                                                      HYS10040
        BB=1.1578*10**B                                                     HYS10050
        IF(BB.GT.1.) BB=1                                                   HYS10060
        SCY=BB*OSCY                                                         HYS10070
      END IF                                                                HYS10080
      COEF1=(OFY-OFC)/(ODY-ODC)                                             HYS10090
      COEF2=(OFU-OFY)/(ODU-ODY)                                             HYS10100
      STU=(COEF2/COEF1)*SCY                                                 HYS10110
      IF(SQ6.LE.0.) THEN                                                    HYS10120
        DY=DT                                                               HYS10130
      ELSE IF((SQ6.GT.0.).AND.(SQ6.LE.0.312)) THEN                          HYS10140
        DY=ODY                                                              HYS10150
      ELSE                                                                  HYS10160
        DY=(1.5299+0.4545*LOG(SQ6))*ODY                                     HYS10170
      END IF                                                                HYS10180
      IF(SQ6.EQ.0.) THEN                                                    HYS10190
        DU=DU                                                               HYS10200
      ELSE                                                                  HYS10210
        DU=DY*(ODU/ODY)                                                     HYS10220
        DU=MAX(DU,1.5*DY)                                                   HYS10230
      END IF                                                                HYS10240
      FY=FC+(DY-DC)*SCY                                                     HYS10250
      FU=FY+(DU-DY)*STU                                                     HYS10260
C ASSUME REFERENCE FAILURE POINT F                                          HYS10270
CC        DF=DU+(FF-FU)/SUF                                                 HYS10280
C 7/11/92 10:30 SAT. BASED ON OBSERVATION, SUF=-0.5*SCY                     HYS10290
      SUF=-0.5*SCY                                                          HYS10300
      FF=(FC+FY)/2                                                          HYS10310
      DF=DU+(FF-FU)/SUF                                                     HYS10320
      SOC=FC/DC                                                             HYS10330
      SCY=FY/DY                                                             HYS10340
      SCU=(FU-FC)/(DU-DC)                                                   HYS10350
      XA=FY/SOC                                                             HYS10360
      XB=(XA+DY)/2                                                          HYS10370
      SOB=FY/XB                                                             HYS10380
      KD=FU/SOY                                                             HYS10390
      XE=(XD+DU)/2                                                          HYS10400
      SOE=FU/XE                                                             HYS10410
      PSQ6=SQ6                                                              HYS10420
      RETURN                                                                HYS10430
      END                                                                   HYS10440
C-----------------------------------------------------------------------    HYS10450
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                          HYS10460
C         END DEBUG                                                         HYS10470
C-----------------------------------------------------------------------    HYS10480
C ===  SUBROUTINE DDISSI  SUBRTN (AREA1=ENERGY DISSIPATION)                 HYS10490
C ===  CALCULATE THE ELASTIC STRAIN ENERGY(ESE(1)) AND                      HYS10500
C ===  PLASTIC STRAIN ENERGY (PSE)                                          HYS10510
C ***********************************************************************  HYS10520
      SUBROUTINE DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DP,FP,AREA1,          HYS10530
     &                  K,USE,KKL,FC,DC,FY,DT,DM,DMAX,ODC,SOB,FU00,       HYS10540
     &                  LRULE)                                            HYS10550
      INTEGER LRULE                                                         HYS10560
      REAL K                                                                HYS10570
      DIMENSION ESE(3)                                                      HYS10580
C                                                                           HYS10590
C AREA1 REPRESENTING AREA ENCLOSED BY THE HYSTERESIS RESPONSE               HYS10600
C TSE,ESE(1),ESE(2),ESE(3),PSE,PSEOLD SEE STRENG SUBRTN                     HYS10610
C USE      = EQUIVALENT UNLOADING STIFFNESS                                 HYS10620
C-------------------------------------------------------------------        HYS10630
      IF(IOPT2.EQ.1 .AND. K.NE.0.) THEN                                     HYS10640
C ... CALCULATE THE EQUIVALENT UNLOADING STIFFNESS                          HYS10650
        IF((ABS(FU00).GT.0.7*FT).OR.(KKL.EQ.7 .OR. KKL.EQ.8 .OR.            HYS10660
     &      KKL.EQ.9 .OR. KKL.EQ.10)) THEN                                  HYS10670
          DDPP=SIGN(DC,-DP)                                                 HYS10680
          FPPP=SIGN(FC,DDPP)                                                HYS10690
          SKK1=(FPPP-FU00*0.05)/(DDPP-DP)                                   HYS10700
          SK1=MAX(SKK1,SOB)                                                 HYS10710
          DP1=DP+(FP/6-FP*0.05)/SK1                                         HYS10720
          FYP=SIGN(FY,-DP1)                                                 HYS10730
          DYP=SIGN(DY,FYP)                                                  HYS10740
          SK2=(FYP-FP/6)/(DYP-DP1)                                          HYS10750
          DP2=DP1+(0.-FP/6)/SK2                                             HYS10760
          AA=0.5*(0.85*FP+FP/6)*(DP-DP1)+0.5*FP/6*(DP1-DP2)                 HYS10770
          USE=(FP**2)/(2*ABS(AA))                                           HYS10780
        ELSE IF(ABS(FP).LE.0.7*FT) THEN                                     HYS10790
          SK1 = FC/DC                                                       HYS10800
          FP1=SIGN(FC/3,FP)                                                 HYS10810
          DP1=DP+(FP1-FP)/SK1                                               HYS10820
          DCP = SIGN(DC,-DP)                                                HYS10830
          FCP = SIGN(FC,DCP)                                                HYS10840
          SK2 = (FCP-FP1)/(DCP-DP1)                                         HYS10850
          DP2 = DP1 + (0.-FP1)/SK2                                          HYS10860
          AA=0.5*(FP1+FP)*(DP-DP1)+0.5*FP1*(DP1-DP2)                        HYS10870
          USE = (FP**2)/(2*ABS(AA))                                         HYS10880
        END IF                                                              HYS10890
      END IF                                                                HYS10900
      IF(IOPT2.EQ.2 .AND. K.NE.0.) THEN                                     HYS10910
        IF(DM.LE.ODC .AND. DMAX.LE.ODC .AND.(KKL.EQ.1 .OR.                  HYS10920
     &     KKL.EQ.3 .OR. KKL.EQ.5 .OR. KKL.EQ.6)) THEN                      HYS10930
          USE=K                                                             HYS10940
        ELSE IF((DP*FP).GT.0.) THEN                                         HYS10950
          USE=(0.-FP)/(DP/3-DP)                                             HYS10960
        END IF                                                              HYS10970
      END IF                                                                HYS10980
      CALL STRENG(ESE,PSE,PSEOLD,FP,FPP,DP,DPP,K,USE)                       HYS10990
      AREA1=ESE(1)+PSE                                                      HYS11000
      RETURN                                                                HYS11010
      END                                                                   HYS11020
C-----------------------------------------------------------------------    HYS11030
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                          HYS11040
C         END DEBUG                                                         HYS11050
C-----------------------------------------------------------------------    HYS11060
C === SUBROUTINE DINTER SUBRTN (CHECK IF NEW CURRENT TRACK INTERSECT         HYS11070
C === WITH TES,FIND CONNECT PT.(UPDATED CURRENT PT.)                        HYS11080
C ***********************************************************************  HYS11090
      SUBROUTINE DINTER(FC,FY,FU,FF,FPP,DC,DT,DU,DF,DPP,SOC,SCY,STU,        HYS11100
     &            SUF,K,FP,DP,DSINC,KKL,LRULE,POSTFG,DU0,FU0)               HYS11110
C                                                                           HYS11120
      REAL K,KJ,DU0(4),FU0(4)                                               HYS11130
      LOGICAL POSTFG                                                        HYS11140
      IF(ABS(DP).LT.DC) THEN                                                HYS11150
        KJ=FP/DP                                                            HYS11160
        IF(KJ.LT.SOC) RETURN                                                HYS11170
        IF(KJ.EQ.SOC) THEN                                                  HYS11180
          IF(DSINC.GT.0) KKL=1                                              HYS11190
          IF(DSINC.LT.0) KKL=3                                              HYS11200
          RETURN                                                            HYS11210
        END IF                                                              HYS11220
        IF(KJ.GT.SOC .AND. K.GT.SOC) THEN                                   HYS11230
          IF(LRULE.EQ.51 .OR. LRULE.EQ.52) THEN                             HYS11240
            X=(K*DPP-FPP)/(K-SOC)                                           HYS11250
            Y=K*SOC                                                         HYS11260
            ADP=SIGN(X,DPP)                                                 HYS11270
            AFP=SIGN(Y,DPP)                                                 HYS11280
            FP=AFP+SOC*(DP-ADP)                                             HYS11290
            IF(DSINC.GT.0) KKL=1                                            HYS11300
            IF(DSINC.LT.0) KKL=3                                            HYS11310
            K=SOC                                                           HYS11320
          END IF                                                            HYS11330
C --------FOR RELOAD DURING REVERSAL STAGE                                  HYS11340
          IF(LRULE.EQ.41 .OR. LRULE.EQ.42) THEN                             HYS11350
            IF(DP.GT.0) THEN                                                HYS11360
              IF(POSTFG) THEN                                               HYS11370
                FFFF=FU0(1)                                                 HYS11380
                GGGG=DU0(1)                                                 HYS11390
              ELSE                                                          HYS11400
                FFFF=FU0(2)                                                 HYS11410
                GGGG=DU0(2)                                                 HYS11420
              END IF                                                        HYS11430
            ELSE                                                            HYS11440
              IF(POSTFG) THEN                                               HYS11450
                FFFF=FU0(2)                                                 HYS11460
                GGGG=DU0(2)                                                 HYS11470
              ELSE                                                          HYS11480
                FFFF=FU0(1)                                                 HYS11490
                GGGG=DU0(1)                                                 HYS11500
              END IF                                                        HYS11510
            END IF                                                          HYS11520
            IF(ABS(DP).GT.ABS(GGGG) .AND. (DP*GGGG).GT.0) THEN              HYS11530
              T2=FPP+(GGGG-DPP)*K                                           HYS11540
              SK=(ABS(FC)-ABS(T2))/(ABS(DC)-ABS(GGGG))                      HYS11550
              FP=T2+(DP-GGGG)*SK                                            HYS11560
              IF(DP.GT.0) KKL=6                                             HYS11570
              IF(DP.LT.0) KKL=5                                             HYS11580
              K=SK                                                          HYS11590
              RETURN                                                        HYS11600
            END IF                                                          HYS11610
          END IF                                                            HYS11620
        END IF                                                              HYS11630
      END IF                                                                HYS11640
      IF((ABS(DP).GE.DC).AND.(ABS(DP).LT.DT)) THEN                          HYS11650
        IF(DP.GT.0) THEN                                                    HYS11660
          IF(POSTFG) THEN                                                   HYS11670
            FFFF=FU0(1)                                                     HYS11680
            GGGG=DU0(1)                                                     HYS11690
          ELSE                                                              HYS11700
            FFFF=FU0(2)                                                     HYS11710
            GGGG=DU0(2)                                                     HYS11720
          END IF                                                            HYS11730
        ELSE                                                                HYS11740
          IF(POSTFG) THEN                                                   HYS11750
            FFFF=FU0(2)                                                     HYS11760
            GGGG=DU0(2)                                                     HYS11770
          ELSE                                                              HYS11780
            FFFF=FU0(1)                                                     HYS11790
            GGGG=DU0(1)                                                     HYS11800
          END IF                                                            HYS11810
        END IF                                                              HYS11820
        IF(ABS(DPP).LT.0 .AND. (LRULE.EQ.41 .OR. LRULE.EQ.42)               HYS11830
     &    .AND. (DP*GGGG).GT.0) THEN                                        HYS11840
          T2=FPP+(GGGG-DPP)*K                                               HYS11850
          SK=(ABS(FC)-ABS(T2))/(ABS(DC)-ABS(GGGG))                          HYS11860
          IF(DP.GT.0) FP=T2+(DC-GGGG)*SK+(DP-DC)*SCY                        HYS11870
          IF(DP.LT.0) FP=T2+(-DC-GGGG)*SK+(DP+DC)*SCY                       HYS11880
          IF(DSINC.GT.0) KKL=1                                              HYS11890
          IF(DSINC.LT.0) KKL=3                                              HYS11900
          K=SCY                                                             HYS11910
          RETURN                                                            HYS11920
        END IF                                                              HYS11930
        KJ=(ABS(FP)-FC)/(ABS(DP)-DC)                                        HYS11940
        IF(KJ.LT.SCY) RETURN                                                HYS11950
        IF(KJ.EQ.SCY) THEN                                                  HYS11960
          IF(DSINC.GT.0) KKL=1                                              HYS11970
          IF(DSINC.LT.0) KKL=3                                              HYS11980
```

```
        RETURN                                                          HYS11990
      END IF                                                            HYS12000
      IF(KJ.GT.SCY) THEN                                                HYS12010
        X=(K*ABS(DPP)-DC*SCY-ABS(FPP)+FC)/(K-SCY)                       HYS12020
        Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                  HYS12030
        IF(ABS(X).LT.DC) THEN                                           HYS12040
          X=(K*DPP-FPP)/(K-SOC)                                         HYS12050
          Y=K*SOC                                                       HYS12060
          ADP=SIGN(X,DPP)                                               HYS12070
          AFP=SIGN(Y,DPP)                                               HYS12080
          IF(DP.GT.0) FP=AFP+SOC*(DC-ADP)+SCY*(DP-DC)                   HYS12090
          IF(DP.LT.0) FP=AFP+SOC*(-DC-ADP)+SCY*(DP+DC)                  HYS12100
          IF(DSINC.GT.0) KKL=1                                          HYS12110
          IF(DSINC.LT.0) KKL=3                                          HYS12120
          K=SCY                                                         HYS12130
          RETURN                                                        HYS12140
        END IF                                                          HYS12150
        ADP=SIGN(X,DPP)                                                 HYS12160
        AFP=SIGN(Y,DPP)                                                 HYS12170
        FP=AFP+SCY*(DP-ADP)                                             HYS12180
        IF(DSINC.GT.0) KKL=1                                            HYS12190
        IF(DSINC.LT.0) KKL=3                                            HYS12200
        K=SCY                                                           HYS12210
      END IF                                                            HYS12220
      RETURN                                                            HYS12230
    END IF                                                              HYS12240
    IF((ABS(DP).GE.DY).AND.(ABS(DP).LT.DU)) THEN                        HYS12250
      KJ=(ABS(FP)-FY)/(ABS(DP)-DY)                                      HYS12260
      IF(KJ.LT.SYU) RETURN                                              HYS12270
      IF(KJ.EQ.SYU) THEN                                                HYS12280
        IF(DSINC.GT.0) KKL=1                                            HYS12290
        IF(DSINC.LT.0) KKL=3                                            HYS12300
        RETURN                                                          HYS12310
      END IF                                                            HYS12320
      IF(KJ.GT.SYU) THEN                                                HYS12330
        X=(FY-DY*SYU-K*ABS(DPP)-ABS(FPP))/(K-SYU)                       HYS12340
        Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                  HYS12350
        IF(ABS(X).LT.DY) THEN                                           HYS12360
          X=(K*ABS(DPP)-DC*SCY-ABS(FPP)+FC)/(K-SCY)                     HYS12370
          Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                HYS12380
          ADP=SIGN(X,DPP)                                               HYS12390
          AFP=SIGN(Y,DPP)                                               HYS12400
          IF(DP.GT.0) FP=AFP+SCY*(DY-ADP)+SYU*(DP-DY)                   HYS12410
          IF(DP.LT.0) FP=AFP+SCY*(-DY-ADP)+SYU*(DP+DY)                  HYS12420
          IF(DSINC.GT.0) KKL=1                                          HYS12430
          IF(DSINC.LT.0) KKL=3                                          HYS12440
          K=SYU                                                         HYS12450
          RETURN                                                        HYS12460
        END IF                                                          HYS12470
        ADP=SIGN(X,DPP)                                                 HYS12480
        AFP=SIGN(Y,DPP)                                                 HYS12490
        FP=AFP+SYU*(DP-ADP)                                             HYS12500
        IF(DSINC.GT.0) KKL=1                                            HYS12510
        IF(DSINC.LT.0) KKL=3                                            HYS12520
        K=SYU                                                           HYS12530
      END IF                                                            HYS12540
      RETURN                                                            HYS12550
    END IF                                                              HYS12560
    IF(ABS(DP).GE.DU .AND. ABS(DP).LT.DF) THEN                          HYS12570
      IF((ABS(DP).EQ.DU).AND.(ABS(FP).LT.FU)) RETURN                    HYS12580
      KJ=(ABS(FP)-FU)/(ABS(DP)-DU)                                      HYS12590
      IF((KJ.LT.0).AND.(KJ.LT.SUF)) RETURN                              HYS12600
      IF(KJ.EQ.SUF) THEN                                                HYS12610
        IF(DSINC.GT.0) KKL=1                                            HYS12620
        IF(DSINC.LT.0) KKL=3                                            HYS12630
        RETURN                                                          HYS12640
      END IF                                                            HYS12650
      IF(KJ.GT.SUF) THEN                                                HYS12660
        X=(FU-DU*SUF+K*ABS(DPP)-ABS(FPP))/(K-SUF)                       HYS12670
        Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                  HYS12680
        IF(ABS(X).LT.DU) THEN                                           HYS12690
          X=(FY-DY*SYU-K*ABS(DPP)-ABS(FPP))/(K-SYU)                     HYS12700
          Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                HYS12710
          ADP=SIGN(X,DPP)                                               HYS12720
          AFP=SIGN(Y,DPP)                                               HYS12730
          IF(DP.GT.0) FP=AFP+SYU*(DU-ADP)+SUF*(DP-DU)                   HYS12740
          IF(DP.LT.0) FP=AFP+SYU*(-DU-ADP)+SUF*(DP+DU)                  HYS12750
          IF(DSINC.GT.0) KKL=1                                          HYS12760
          IF(DSINC.LT.0) KKL=3                                          HYS12770
          K=SUF                                                         HYS12780
          RETURN                                                        HYS12790
        END IF                                                          HYS12800
        ADP=SIGN(X,DPP)                                                 HYS12810
        AFP=SIGN(Y,DPP)                                                 HYS12820
        FP=AFP+SUF*(DP-ADP)                                             HYS12830
        IF(DSINC.GT.0) KKL=1                                            HYS12840
        IF(DSINC.LT.0) KKL=3                                            HYS12850
        K=SUF                                                           HYS12860
      END IF                                                            HYS12870
      RETURN                                                            HYS12880
    END IF                                                              HYS12890
C                                                                       HYS12900
    IF(ABS(DP).GE.DF .AND. ABS(DP).LT.(20*DY)) THEN                     HYS12910
      KJ=(ABS(FP)-FF)/(ABS(DP)-DF)                                      HYS12920
      DG=20*DY                                                          HYS12930
      SFG=(0.1*FU-FF)/(DG-DF)                                           HYS12940
      IF(KJ.EQ.SFG) THEN                                                HYS12950
        IF(DSINC.GT.0) KKL=1                                            HYS12960
        IF(DSINC.LT.0) KKL=3                                            HYS12970
        RETURN                                                          HYS12980
      END IF                                                            HYS12990
      IF(KJ.GT.SFG) THEN                                                HYS13000
        X=(0.1*FU-DG*SFG+K*ABS(DPP)-ABS(FPP))/(K-SFG)                   HYS13010
        Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                  HYS13020
        IF(ABS(X).LT.DU) THEN                                           HYS13030
CC  ... TRY WHETHER IT IS LOCATED IN BWN U AND F .....                  HYS13040
          X=(FU-DU*SUF+K*ABS(DPP)-ABS(FPP))/(K-SUF)                     HYS13050
          Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                HYS13060
          IF(ABS(X).LT.DU) THEN                                         HYS13070
            X=(FY-DY*SYU+K*ABS(DPP)-ABS(FPP))/(K-SYU)                   HYS13080
            Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                              HYS13090
            ADP=SIGN(X,DPP)                                             HYS13100
            AFP=SIGN(Y,DPP)                                             HYS13110
            IF(DP.GT.0) FP=AFP+SYU*(DU-ADP)+SUF*(DP-DU)+SFG*(DP-DF)     HYS13120
            IF(DP.LT.0) FP=AFP+SYU*(-DU-ADP)+SUF*(-DF+DU)+SFG*(DP+DF)   HYS13130
            IF(DSINC.GT.0) KKL=1                                        HYS13140
            IF(DSINC.LT.0) KKL=3                                        HYS13150
            K=SFG                                                       HYS13160
            RETURN                                                      HYS13170
          END IF                                                        HYS13180
          ADP=SIGN(X,DPP)                                               HYS13190
          AFP=SIGN(Y,DPP)                                               HYS13200
          IF(DP.GT.0) FP=AFP+SUF*(DF-ADP)+SFG*(DP-DF)                   HYS13210
          IF(DP.LT.0) FP=AFP+SUF*(-DF-ADP)+SFG*(DP+DF)                  HYS13220
          IF(DSINC.GT.0) KKL=1                                          HYS13230
          IF(DSINC.LT.0) KKL=3                                          HYS13240
          K=SFG                                                         HYS13250
          RETURN                                                        HYS13260
        END IF                                                          HYS13270
        IF(ABS(X).LT.DF .AND. ABS(X).GT.DU) THEN                        HYS13280
          X=(FF-DF*SUF+K*ABS(DPP)-ABS(FPP))/(K-SUF)                     HYS13290
          Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                HYS13300
          ADP=SIGN(X,DPP)                                               HYS13310
          AFP=SIGN(Y,DPP)                                               HYS13320
          IF(DP.GT.0) FP=AFP+SUF*(DF-ADP)+SFG*(DP-DF)                   HYS13330
          IF(DP.LT.0) FP=AFP+SUF*(-DF-ADP)+SFG*(DP+DF)                  HYS13340
          IF(DSINC.GT.0) KKL=1                                          HYS13350
          IF(DSINC.LT.0) KKL=3                                          HYS13360
          K=SFG                                                         HYS13370
          RETURN                                                        HYS13380
        END IF                                                          HYS13390
        ADP=SIGN(X,DPP)                                                 HYS13400
        AFP=SIGN(Y,DPP)                                                 HYS13410
        FP=AFP+SFG*(DP-ADP)                                             HYS13420
        IF(DSINC.GT.0) KKL=1                                            HYS13430
        IF(DSINC.LT.0) KKL=3                                            HYS13440
        K=SFG                                                           HYS13450
      END IF                                                            HYS13460
      RETURN                                                            HYS13470
    END IF                                                              HYS13480
C                                                                       HYS13490
    DH=20*DY+(0.-0.1*FU)/(-0.0001*SCY)                                  HYS13500
    SGH=-0.0001*SCY                                                     HYS13510
    SFG=(0.1*FU-FF)/(20*DY-DF)                                          HYS13520
    IF(ABS(DP).GE.(20*DY) .AND. ABS(DP).LT.DH) THEN                     HYS13530
      IF(ABS(DP).EQ.(20*DY) .AND. ABS(FP).LT.(0.1*FU)) RETURN           HYS13540
      KJ=(ABS(FP)-(0.1*FU))/(ABS(DP)-(20*DY))                           HYS13550
      IF(KJ.LT.0 .AND. KJ.LT.SGH) RETURN                                HYS13560
      IF(KJ.EQ.SGH) THEN                                                HYS13570
        IF(DSINC.GT.0) KKL=1                                            HYS13580
        IF(DSINC.LT.0) KKL=3                                            HYS13590
        RETURN                                                          HYS13600
      END IF                                                            HYS13610
      IF(KJ.GT.SGH) THEN                                                HYS13620
        X=(0.1*FU-20*DY*SGH+K*ABS(DPP)-ABS(FPP))/(K-SGH)                HYS13630
        Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                  HYS13640
        IF(ABS(X).LT.DF) THEN                                           HYS13650
CC  ... TRY WHETHER IT IS LOCATED IN BTN F AND G .....                  HYS13660
          X=(0.1*FU-DG*SFG+K*ABS(DPP)-ABS(FPP))/(K-SFG)                 HYS13670
          Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                HYS13680
          IF(ABS(X).LT.DF) THEN                                         HYS13690
            X=(FU-DU*SUF+K*ABS(DPP)-ABS(FPP))/(K-SUF)                   HYS13700
            Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                              HYS13710
            ADP=SIGN(X,DPP)                                             HYS13720
            AFP=SIGN(Y,DPP)                                             HYS13730
            IF(DP.GT.0) FP=AFP+SUF*(DF-ADP)+SFG*(DG-DF)+SGH*(DP-DG)     HYS13740
            IF(DP.LT.0) FP=AFP+SUF*(-DF-ADP)+SFG*(-DG+DF)+SGH*(DP+DG)   HYS13750
            IF(DSINC.GT.0) KKL=1                                        HYS13760
            IF(DSINC.LT.0) KKL=3                                        HYS13770
            K=SGH                                                       HYS13780
            RETURN                                                      HYS13790
          END IF                                                        HYS13800
          ADP=SIGN(X,DPP)                                               HYS13810
          AFP=SIGN(Y,DPP)                                               HYS13820
          IF(DP.GT.0) FP=AFP+SFG*(DF-ADP)+SGH*(DP-DF)                   HYS13830
          IF(DP.LT.0) FP=AFP+SFG*(-DF-ADP)+SGH*(DP+DF)                  HYS13840
          FP=SIGN(FP,DP)                                                HYS13850
          IF(DSINC.GT.0) KKL=1                                          HYS13860
          IF(DSINC.LT.0) KKL=3                                          HYS13870
          K=SGH                                                         HYS13880
          RETURN                                                        HYS13890
        END IF                                                          HYS13900
        IF(ABS(X).LT.(20*DY)) THEN                                      HYS13910
          X=(FF-DF*SFG+K*ABS(DPP)-ABS(FPP))/(K-SFG)                     HYS13920
          Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                HYS13930
          ADP=SIGN(X,DPP)                                               HYS13940
          AFP=SIGN(Y,DPP)                                               HYS13950
          IF(DP.GT.0) FP=AFP+SFG*(20*DY-ADP)+SGH*(DP-20*DY)             HYS13960
          IF(DP.LT.0) FP=AFP+SFG*(-20*DY-ADP)+SGH*(DP+20*DY)            HYS13970
          IF(DSINC.GT.0) KKL=1                                          HYS13980
          IF(DSINC.LT.0) KKL=3                                          HYS13990
          K=SGH                                                         HYS14000
          RETURN                                                        HYS14010
        END IF                                                          HYS14020
        ADP=SIGN(X,DPP)                                                 HYS14030
        AFP=SIGN(Y,DPP)                                                 HYS14040
        FP=AFP+SGH*(DP-ADP)                                             HYS14050
        IF(DSINC.GT.0) KKL=1                                            HYS14060
        IF(DSINC.LT.0) KKL=3                                            HYS14070
        K=SGH                                                           HYS14080
      END IF                                                            HYS14090
      RETURN                                                            HYS14100
    END IF                                                              HYS14110
    IF(ABS(DP).GE.DH) THEN                                              HYS14120
      K=0.                                                              HYS14130
      IF(DSINC.GT.0) KKL=1                                              HYS14140
      IF(DSINC.LT.0) KKL=3                                              HYS14150
      RETURN                                                            HYS14160
    END IF                                                              HYS14170
    RETURN                                                              HYS14180
    END                                                                 HYS14190
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP                                  IDR00010
C*********************************************************************IDR00020
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                      IDR00030
C         END DEBUG                                                     IDR00040
C                                                                       IDR00050
      SUBROUTINE IDRECT (IDR,IDRV,IDRVO,PI,PS,VI,VS)                    IDR00060
C                                                                       IDR00070
      LOGICAL A,NA,B,NB,C,NC,D,ND,E                                     IDR00080
      LOGICAL BTEST                                                     IDR00090
C                                                                       IDR00100
C     WRITE (6,*) 'IDR,IDRV,IDRVO,PI,PS,VI,VS'                          IDR00110
C     WRITE (6,*)  IDR,IDRV,IDRVO,PI,PS,VI,VS                           IDR00120
C                                                                       IDR00130
  1   A = ABS(PI).GE.ABS(PS)                                            IDR00140
      B = PI*PS .GE.0                                                   IDR00150
C     B = PI*PS .GT.0                                                   IDR00160
      C = PI.EQ.0                                                       IDR00170
      D = VS*VI.GE.0                                                    IDR00180
      NA= .NOT. A                                                       IDR00190
      NB= .NOT. B                                                       IDR00200
      NC= .NOT. C                                                       IDR00210
      ND= .NOT. D                                                       IDR00220
C                                                                       IDR00230
      IF ( A .AND.  B .AND. NC .AND.  D ) IDR=1                         IDR00240
      IF (          NB .AND. NC .AND. ND ) IDR=1                        IDR00250
      IF (NA .AND.  B .AND. NC .AND. ND ) IDR=1                         IDR00260
      IF (                C .AND.  D ) IDR=0                          IDR00270
      IF (          NB .AND. NC .AND.  D ) IDR=0                        IDR00280
      IF ( A .AND.  B .AND. NC .AND. ND ) IDR=-1                        IDR00290
      IF (                C .AND. ND ) IDR=-1                         IDR00300
      IF (NA .AND.  B .AND. NC .AND.  D ) IDR=-1                        IDR00310
           E = PI .GE. PS                                               IDR00320
           IDRV=1                                                       IDR00330
           IDRVO=2                                                      IDR00340
           IF (E) IDRV=2                                                IDR00350
           IF (E) IDRVO=1                                               IDR00360
      RETURN                                                            IDR00370
      END                                                               IDR00380
C*********************************************************************INT00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE                                 INT00020
C         END DEBUG                                                     INT00030
      SUBROUTINE INTDVA(BUG,NDOF,MAXNOD,NNODE,TITLE,                    INT00040
     &                  IDOF,ID,JTFLG,DISP,VEL,ACC,FORCE )              INT00050
C                                                                       INT00060
      COMMON /ISPEC/ ISP1, SGD(3000,2),REACF(6)                         INT00070
      DIMENSION DISP(NDOF),VEL(NDOF),ACC(NDOF),A(6),FORCE(NDOF)         INT00080
      DIMENSION ID(MAXNOD), JTFLG(MAXNOD), IDOF(MAXNOD,6)               INT00090
      DIMENSION RINPUT(100)                                             INT00100
      CHARACTER*80  FORMAT(2)                                           INT00110
      CHARACTER*(*) TITLE(2)                                            INT00120
      CHARACTER*20 OPTION                                               INT00130
      LOGICAL  BUG,HEAD,TEST,FLAG                                       INT00140
      LOGICAL BTEST                                                     INT00150
C                                                                       INT00160
C=======================================                                INT00170
C= INPUT INITIAL DISPL, VELOC AND ACCEL  ===                            INT00180
C=======================================                                INT00190
C                                                                       INT00200
      HEAD=.TRUE.                                                       INT00210
C= CONSIDER EXTERNAL DEFORMATION DUE TO STATICS CASE, ISP1=1 ===        INT00220
C     NOTE: 'RELEASE' SHOULD NOT USED AFTER 'SOL01'                     INT00230
C= NEGLECT  EXTERNAL DEFORMATION DUE TO STATICS CASE, ISP1=0 ===        INT00240
C     NOTE: 'RELEASE' SHOULD USED AFTER 'SOL01'                         INT00250
```

```
      READ(5,*)ISP1
      IF(ISP1 .EQ. 0 ) THEN
        WRITE(6,*)
        WRITE(6,*)' NOT INCLUDE STATICS CASE EXTERNAL DEFORMATIONS '
        WRITE(6,*)' ***************************************************'
      ENDIF
      IF(ISP1 .EQ. 1 ) THEN
        WRITE(6,*)
        WRITE(6,*)' STATICS CASE EXTERNAL DEFORMATIONS ARE INCLUDED'
        WRITE(6,*)' ***************************************************'
      ENDIF
C*
      IF( ISP1 .EQ. 1 ) THEN
       DO 49 I=1,NDOF
         DISP(I)=SGD(I,2)
         VEL(I)=0
         ACC(I)=0
         FORCE(I)=0
   49 CONTINUE
      ENDIF
C*
C--- INPUT: NODE,IGEN,INC,OPTION,DX,DY,DZ,RX,RY,RZ
C
      IF( ISP1 .NE. 1 ) THEN
   10  READ (5,*) NODE,IGEN,INC,OPTION,(A(I),I=1,6)
       IF (NODE.LE.0) GO TO 500
       IF (HEAD) THEN
            HEAD=.FALSE.
            WRITE (6,5) TITLE(1),TITLE(2)
       ENDIF
       WRITE (6,7) NODE,IGEN,INC,OPTION,(A(I),I=1,6)
    5  FORMAT ('1 STRUCTURE....: ',A,/
     &       '  SOLUTION.....: ',A,//
     &       '  INITIAL ACCELERATIONS, VELOCITIES DISPLACEMENTS ',
     &      'AND LOADS '/
     &       ' ---------------------------------------------------',
     &       '----------'//
     &    4X,'NODE',2X,'IGEN',3X,'INC',1X,'COMPONENT',
     &         T44,'TX',10X,'TY',10X,'TZ',
     &        10X,'RX',10X,'RY',10X,'RZ')
    7  FORMAT (2X,3I6,1X,A,1P,6G12.4)
C
C------- DETERMINE IF DISPLACMENT, VELOCITY, ACCELERATION OR LOAD
      IOPT=0
      IF (TEST(OPTION,'DISPL',.FALSE.)) THEN
        IOPT=IBSET(IOPT,1)
      ELSE IF (TEST(OPTION,'VEL',.FALSE.)) THEN
        IOPT=IBSET(IOPT,2)
      ELSE IF (TEST(OPTION,'ACC',.FALSE.)) THEN
        IOPT=IBSET(IOPT,3)
      ELSE IF (TEST(OPTION,'LOA',.FALSE.)) THEN
        IOPT=IBSET(IOPT,4)
      ELSE IF (TEST(OPTION,'FOR',.FALSE.)) THEN
        IOPT=IBSET(IOPT,4)
      ELSE IF (TEST(OPTION,'END',.FALSE.)) THEN
        GO TO 500
      ELSE
        WRITE (6,*) 'INVALID OPTION:',OPTION
        GO TO 10
      ENDIF
C
C------- DETERMINE INTERNAL JOINT NUMBER
   20 JOINT=IQUICK(NODE,ID,NNODE)
      IF (JOINT.LE.0) THEN
        WRITE (6,*) 'NODE #',NODE,
     &  ' WAS NOT FOUND, CHECK INPUT, INITIAL VALUE IS OMITTED '
        GO TO 10
      ENDIF
C
C------- PUT INITIAL VALUES IN PROPER ARRAYS
      DO 30 I=1,6
         IF (A(I).EQ.0) GO TO 30
         II=IDOF(JOINT,I)
         IF (BTEST(JTFLG(JOINT),I+11)) THEN
           WRITE (6,*) 'CONSTRAINED DOF #',I,' COMPONENT IS SKIPED'
         ELSE IF (BTEST(IOPT,1)) THEN
           DISP(II)=A(I)
         ELSE IF (BTEST(IOPT,2)) THEN
           VEL(II)=A(I)
         ELSE IF (BTEST(IOPT,3)) THEN
           ACC(II)=A(I)
         ELSE IF (BTEST(IOPT,4)) THEN
           FORCE(II)=A(I)
         ENDIF
   30 CONTINUE
C
C------- GENERATE VALUES AT OTHER NODES
      IF (IGEN.GT.0) THEN
         IGEN=IGEN-1
         NODE=NODE+INC
         GO TO 20
      ENDIF
C
C------- LOOP TO READ ADDITIONAL VALUES
      GO TO 10
      ENDIF
C
  500 CONTINUE
C     ISP1=0
C
C------- PRINT OUT NON ZERO INITIAL VALUES
      FLAG=.TRUE.
      DO 520 I=1,NDOF
        IF (DISP(I).NE.0 .OR. VEL(I).NE.0 .OR. ACC(I).NE.0 .OR.
     &      FORCE(I).NE.0 ) THEN
            IF (FLAG) WRITE (6,510)
            WRITE (6,530) I,FORCE(I),DISP(I),VEL(I),ACC(I)
            FLAG=.FALSE.
        ENDIF
  520 CONTINUE
C
  510 FORMAT(//5X,'INITIAL, NON-ZERO VALUES'/
     &  7X,'DOF',7X,'FORCE',10X,'DISP',11X,'VEL ',11X,'ACC ')
  530 FORMAT (I10,4G15.6)
C
      RETURN
C
      END
      SUBROUTINE INTPO(PERIOD,RSV,NAP,RSNP)
      DIMENSION RSNP(2,NAP)
C-----------------------------------------
C -- INTERPORATION OF RESPONSE SPECTRUM VALUE
C-----------------------------------------
C
      IF( PERIOD .LT. RSNP(1,1) ) THEN
            RSV=RSNP(2,1)
            RETURN
      ELSE IF( PERIOD .GT. RSNP(1,NAP) ) THEN
            RSV=RSNP(2,NAP)
            RETURN
      ENDIF
C
      I=0
  300 I=I+1
      J=I+1
      IF( PERIOD .GE. RSNP(1,I) .AND. PERIOD .LT. RSNP(1,J) ) THEN
          COEF=(RSNP(2,J)-RSNP(2,I))/(RSNP(1,J)-RSNP(1,I))
          RSV=(PERIOD-RSNP(1,I))*COEF + RSNP(2,I)
      ELSE IF( PERIOD .GE. RSNP(1,J)) THEN
          GO TO 300
      ENDIF
C
      RETURN
      END
C - INTERSECTION OF TWO LINES IN POINT SLOPE FORM .....
C          DEBUG UNIT(6),SUBCHK,SUBTRACE
C          END DEBUG
C     X       = X COORDINATE OF INTERSECTION
C     Y       = Y COORDINATE OF INTERSECTION
C     X1      = X COORDINATE OF LINE 1
C     Y1      = Y COORDINATE OF LINE 1
C     S1      = SLOPE OF LINE 1
C     X2      = X COORDINATE OF LINE 2
C     Y2      = Y COORDINATE OF LINE 2
C     S2      = SLOPE OF LINE 2
C
      SUBROUTINE INTSCT(X,Y,X1,Y1,S1,X2,Y2,S2)
      IMPLICIT REAL(A-H,O-Z)
      IMPLICIT INTEGER(L-N)
      LOGICAL BTEST
      IF (S1.NE.S2) THEN
           B1= Y1 - S1*X1
           B2= Y2 - S2*X2
           X=(B1-B2)/(S2-S1)
           Y=(B1*S2-B2*S1)/(S2-S1)
      ELSE
           WRITE (6,10) X1,Y1,S1,X2,Y2,S2
   10      FORMAT (/' ****** ER','ROR IN INTSCT....',1X,
     &          'S1=S2, NO SOLN,  THUS SET X=Y=0.',/
     &          '   X1=',1P,G15.5,'  Y1=',G15.5,'  S1=',G15.5/
     &          '   X2=',1P,G15.5,'  Y2=',G15.5,'  S2=',G15.5/)
           X=0
           Y=0
           CALL ERRTRA
      ENDIF
CC         WRITE (6,5) X1,Y1,S1,B1,X2,Y2,S2,B2,X,Y
CC  5      FORMAT (/' IN INTSCT....',/
CC     &    '   X1=',1P,G15.5,'  Y1=',G15.5,'  S1=',G15.5,'  B1=',G15.5/
CC     &    '   X2=',1P,G15.5,'  Y2=',G15.5,'  S2=',G15.5,'  B2=',G15.5/
CC     &    '    X=',1P,G15.5,'   Y=',G15.5/)
      RETURN
      END
C-----------------------------------------------------------------------
C          DEBUG UNIT(6),SUBCHK,SUBTRACE
C          END DEBUG
      INTEGER FUNCTION IQUICK(TARGET,ID,N)
      INTEGER TARGET,ID(N)
      LOGICAL BTEST
C
      IF (ID(1).GT.TARGET .OR. ID(N).LT.TARGET ) GO TO 100
      IQUICK=1
      IF (ID(IQUICK).EQ.TARGET ) RETURN
      IQUICK=N
      IF (ID(IQUICK).EQ.TARGET ) RETURN
      J=(N+1)/2
      IQUICK=J
   10 J= J    /2
      IF (IQUICK.LT.1) IQUICK=1
      IF (IQUICK.GT.N) IQUICK=N
      IF (ID(IQUICK).EQ.TARGET)  RETURN
      IF (ID(IQUICK).GT.TARGET .AND. J.NE.0 )  THEN
         IQUICK=IQUICK-J
         GO TO 10
      ELSE IF (ID(IQUICK).LT.TARGET .AND. J.NE.0 )  THEN
         IQUICK=IQUICK+J
         GO TO 10
      ENDIF
C
      I=MAX(IQUICK-2,1)
      J=MIN(I+4,N)
      DO 20 IQUICK=I,J,1
       IF (ID(IQUICK).EQ.TARGET)  RETURN
   20 CONTINUE
C
      DO 30 IQUICK=1,N
       IF (ID(IQUICK).EQ.TARGET)  RETURN
   30 CONTINUE
C
  100 WRITE (6,110) TARGET,IQUICK,ID(IQUICK)
  110 FORMAT (' TARGET=',I8,'   NOT FOUND, LAST ID(',I6,')=',I8)
      IQUICK=0
      RETURN
      END
C-----------------------------------------------------------------------
C          DEBUG UNIT(6),SUBCHK,SUBTRACE
C          END DEBUG
C
      SUBROUTINE ISORT(ID,INDEX,N)
      INTEGER ID(N)
      INTEGER INDEX(N)
      LOGICAL FLAG
      LOGICAL BTEST
      DO 10 I=1,N
       INDEX(I)=I
C      WRITE (6,*) I,INDEX(I),ID(I)
   10  CONTINUE
C
      DO 30 I=N,2,-1
       FLAG=.TRUE.
       DO 20 J=2,I
        IF ( ID(INDEX(J)).LT.ID(INDEX(J-1)) ) THEN
            K=INDEX(J)
            INDEX(J)=INDEX(J-1)
            INDEX(J-1)=K
            FLAG=.FALSE.
        ENDIF
   20  CONTINUE
       IF (FLAG) RETURN
   30  CONTINUE
      RETURN
      END
C@PROCESS SDUMP GOSTMT XREF
      SUBROUTINE ITERS(PI,TMXI,TMXJ,TMPX,PY,TMPYI,TMPXJ,
     &                 TMYI,TMYJ,TMPY,   TMPYI,TMPYJ,IAFAG )
C     ---------------------------------------------------------------
C     THIS SUBROUTINE IS FOR JUDGING BOX SECTION STRENGTH CRITERIA
C     BASED ON W.F.C'S PAPER
C     NOTE:
C      PI    : CURRENT AXIAL LOAD
C      PY    : AXIAL YIELDING LOAD CAPACITY
C      TMXI  : CURRENT MOMENT IN X DIRECTION AT I END
C      TMXJ  : CURRENT MOMENT IN X DIRECTION AT J END
C      TMYI  : CURRENT MOMENT IN Y DIRECTION AT I END
C      TMYJ  : CURRENT MOMENT IN Y DIRECTION AT J END
C      TMPX  : FULLY PLASTIC MOMENT IN X DIRECTION
C      TMPY  : FULLY PLASTIC MOMENT IN Y DIRECTION
C      TMPXI : CURRENT REDUCED PLASTIC MOMENT IN X DIRECTION AT I END
C      TMPXJ : CURRENT REDUCED PLASTIC MOMENT IN X DIRECTION AT J END
C      TMPYI : CURRENT REDUCED PLASTIC MOMENT IN Y DIRECTION AT I END
C      TMPYJ : CURRENT REDUCED PLASTIC MOMENT IN Y DIRECTION AT J END
C      IAFAG : FLAG FOR JUDGING FULLY PLASTIC SECTION
C              1: MEMBER TWO ENDS STILL ARE NOT FULLY PLASTIC
C              2: MEMBER A END IS IN FULLY PLASTIC
C              3: MEMBER B END IS IN FULLY PLASTIC
C              4: MEMBER A AND B ENDS ARE IN FULLY PLASTIC
C     ---------------------------------------------------------------
C
      IF( PI .GE. 0 ) RETURN
      SP=ABS(PI/PY)
      IF( SP .EQ. 0 ) THEN
          RETURN
      ELSE IF( SP .LT. 1 .AND. SP .GT. 0 ) THEN
          A1=1.7-(SP/LOG(SP))
      ELSE IF( SP .GE. 1 ) THEN
```

```
          IAFAG=4                                                       ITE00340
          RETURN                                                        ITE00350
       ENDIF                                                            ITE00360
       TMPCXI=1.2*(1-SP)*TMPX                                           ITE00370
       IF(ABS(TMPCXI) .GT. ABS(TMPX)) TMPCXI=TMPX                       ITE00380
       TMPCXJ=TMPCXI                                                    ITE00390
       TMPCYI=1.2*(1-SP)*TMPY                                           ITE00400
       IF(ABS(TMPCYI) .GT. ABS(TMPY)) TMPCYI=TMPY                       ITE00410
       TMPCYJ=TMPCYI                                                    ITE00420
       RMXI=ABS(TMXI)                                                   ITE00430
       RMXJ=ABS(TMXJ)                                                   ITE00440
       RMYI=ABS(TMYI)                                                   ITE00450
       RMYJ=ABS(TMYJ)                                                   ITE00460
C                                                                       ITE00470
       IF( IAFAG .EQ. 1 ) THEN                                          ITE00480
          IF( (RMXI/TMPCXI) .GE. 1 .OR. (RMYI/TMPCYI) .GE. 1 ) THEN    ITE00490
              YIELD1=1                                                  ITE00500
          ELSE IF((RMXI/TMPCXI) .LT. 1 .AND. (RMYI/TMPCYI) .LT. 1 ) THEN ITE00510
             YIELD1=(RMXI/TMPCXI)**A1 + (RMYI/TMPCYI)**A1               ITE00520
          ENDIF                                                         ITE00530
          IF( (RMXJ/TMPCXJ) .GE. 1 .OR. (RMYJ/TMPCYJ) .GE. 1 ) THEN     ITE00540
              YIELD2=1                                                  ITE00550
          ELSE IF((RMXJ/TMPCXJ) .LT. 1 .AND. (RMYJ/TMPCYJ) .LT. 1 ) THEN ITE00560
             YIELD2=(RMXJ/TMPCXJ)**A1 + (RMYJ/TMPCYJ)**A1               ITE00570
          ENDIF                                                         ITE00580
C                                                                       ITE00590
          IF( YIELD1 .GE. 1 .AND. YIELD2 .LT. 1 ) THEN                  ITE00600
             IAFAG=2                                                    ITE00610
          ELSE IF( YIELD1 .LT. 1 .AND. YIELD2 .GE. 1 ) THEN             ITE00620
             IAFAG=3                                                    ITE00630
          ELSE IF( YIELD1 .GE. 1 .AND. YIELD2 .GE. 1 ) THEN             ITE00640
             IAFAG=4                                                    ITE00650
          ENDIF                                                         ITE00660
       ELSE IF( IAFAG .EQ. 2 ) THEN                                     ITE00670
          IF( (RMXJ/TMPCXJ) .GE. 1 .OR. (RMYJ/TMPCYJ) .GE. 1 ) THEN     ITE00680
            YIELD2=1                                                    ITE00690
          ELSE IF((RMXJ/TMPCXJ) .LT. 1 .AND. (RMYJ/TMPCYJ) .LT. 1 ) THEN ITE00700
            YIELD2=(RMXJ/TMPCXJ)**A1 + (RMYJ/TMPCYJ)**A1                ITE00710
          ENDIF                                                         ITE00720
          IF( YIELD2 .GE. 1) THEN                                       ITE00730
             IAFAG=4                                                    ITE00740
          ENDIF                                                         ITE00750
       ELSE IF( IAFAG .EQ. 3 ) THEN                                     ITE00760
          IF( (RMXI/TMPCXI) .GE. 1 .OR. (RMYI/TMPCYI) .GE. 1 ) THEN     ITE00770
            YIELD1=1                                                    ITE00780
          ELSE IF((RMXI/TMPCXI) .LT. 1 .AND. (RMYI/TMPCYI) .LT. 1 ) THEN ITE00790
            YIELD1=(RMXI/TMPCXI)**A1 + (RMYI/TMPCYI)**A1                ITE00800
          ENDIF                                                         ITE00810
          IF( YIELD1 .GE. 1) THEN                                       ITE00820
             IAFAG=4                                                    ITE00830
          ENDIF                                                         ITE00840
       ENDIF                                                            ITE00850
C                                                                       ITE00860
       RETURN                                                           ITE00870
       END                                                              ITE00880
       SUBROUTINE ITERST(PI,PY, TMXI,TMXJ,TMPX,                         ITE00010
      &                          TMYI,TMYJ,TMPY,                       ITE00020
      &                          TIX,TIY,B,CL,AR,TJ,                   ITE00030
      &                          SLKX,SLKY,SX,B,                       ITE00040
      &                          H,IFAG)                               ITE00050
C     ------------------------------------------------------------      ITE00060
C     THIS SUBROUTINE IS FOR JUDGING BOX SECTION STABILITY CRITERIA     ITE00070
C     BASED ON W.F.C'S PAPER                                            ITE00080
C     NOTE:                                                             ITE00090
C      PI    : CURRENT AXIAL LOAD                                       ITE00100
C      PY    : AXIAL YIELDING LOAD CAPACITY                             ITE00110
C      TMXI  : CURRENT MOMENT IN X DIRECTION AT I END                   ITE00120
C      TMXJ  : CURRENT MOMENT IN X DIRECTION AT J END                   ITE00130
C      TMYI  : CURRENT MOMENT IN Y DIRECTION AT I END                   ITE00140
C      TMYJ  : CURRENT MOMENT IN Y DIRECTION AT J END                   ITE00150
C      TMPX  : FULLY PLASTIC MOMENT IN X DIRECTION                      ITE00160
C      TMPY  : FULLY PLASTIC MOMENT IN Y DIRECTION                      ITE00170
C      TIX   : MOMENT OF INERTIA ABOUT X DIRECTION                      ITE00180
C      TIY   : MOMENT OF INERTIA ABOUT Y DIRECTION                      ITE00190
C      B     : BOX SECTION WILDTH                                       ITE00200
C      H     : BOX SECTION HIGH                                         ITE00210
C      E     : ELASTIC MODULUS OF ELASTICITY                            ITE00220
C      AR    : NET AREA OS BOX SECTION                                  ITE00230
C      IFAG  : FLAG  0: CRITERIA SATISFIED, 1: CRITERIA .GT. 1          ITE00240
C      SLKX  : SLENDERNESS COEFFICIENT IN X DIRECTION                   ITE00250
C      SLKY  : SLENDERNESS COEFFICIENT IN Y DIRECTION                   ITE00260
C      SX    : SECTION MODULUS IN X DIRECTION ( STRONG AXIS DIRECTION   ITE00270
C      J     : TORSIONAL RIGIDITY                                       ITE00280
C      VALUE: STABILITY CRITERIA FORMULA                                ITE00290
C             >=1 : UNSTABLE, <1 : STABLE                               ITE00300
C      NTYPE: TYPE OF SECTION                                           ITE00310
C             1 : FOR BOX SECTION                                       ITE00320
C     ------------------------------------------------------------      ITE00330
C                                                                       ITE00340
C.... SKIP TENSION AXIAL LOAD CASE                                      ITE00350
       IF( PI .LT. 0 ) RETURN                                           ITE00360
C                                                                       ITE00370
       PIC=ABS(PI)                                                      ITE00380
       PII=3.1415926                                                    ITE00390
       PY=PY/AR                                                         ITE00400
       RX=SQRT(TIX/AR)                                                  ITE00410
       RY=SQRT(TIY/AR)                                                  ITE00420
       TNAX=SLKX*CL/RX                                                  ITE00430
       TNAY=SLKY*CL/RY                                                  ITE00440
       TNAC=TNAY/SQRT(PII*PII*E/PY)                                     ITE00450
       IF( TNAC .LE. 1.5 ) THEN                                         ITE00460
          PN=PY*EXP(-0.419*TNAC*TANC)                                   ITE00470
       ELSE IF( TNAC .GT. 1.5 ) THEN                                    ITE00480
          PN=0.877*PY/(TNAC*TNAC)                                       ITE00490
       ENDIF                                                            ITE00500
       PEX=PII*PII*E*AR/((SLKX*CL/RX)**2)                               ITE00510
       PEY=PII*PII*E*AR/((SLKY*CL/RY)**2)                               ITE00520
C                                                                       ITE00530
C     CLACULATE COLUMN LATERAL BUCKLING MOMENT                          ITE00540
C                                                                       ITE00550
       IF( ABS(TMXI) .GT. ABS(TMXJ)) THEN                               ITE00560
          TM1=TMXJ                                                      ITE00570
          TM2=TMXI                                                      ITE00580
       ELSE IF( ABS(TMXI) .LE. ABS(TMXJ)) THEN                          ITE00590
          TM1=TMXI                                                      ITE00600
          TM2=TMXJ                                                      ITE00610
       ENDIF                                                            ITE00620
       IF ( TM2 .EQ. 0 ) RETURN                                         ITE00630
       CB=1.75+1.05*(TM1/TM2)+0.3*(TM1/TM2)**2                          ITE00640
       IF( CB .GT. 2.3 ) CB=2.3                                         ITE00650
       TMR=PI*SX                                                        ITE00660
       TLR=57000*RY*SQRT(TJ*AR)/TMR                                     ITE00670
       TLP=3750*RY*SQRT(TJ*AR)/TMPX                                     ITE00680
       TMCR=57000*CB*SQRT(TJ*AR)/(CL/RY)                                ITE00690
       IF( CL .GT. TLR) THEN                                            ITE00700
          TMNX=TMCR                                                     ITE00710
          IF(TMNX .GT. (CB*TMR)) TMNX=CB*TMR                            ITE00720
       ELSE IF (TLP .LT. CL .AND. CL .LT. TLR ) THEN                    ITE00730
          TMNX=CB*(TMPX-(TMPX-TMR)*((CL-TLP)/(TLR-TLP)))                ITE00740
          IF(TMNX .GT. TMPX) TMNX=TMPX                                  ITE00750
       ELSE IF(CL .LT. TLP) THEN                                        ITE00760
          TMNX=TMPX                                                     ITE00770
       ENDIF                                                            ITE00780
C                                                                       ITE00790
C     CALCULATE REDUCED PLASTIC MOMENT CAPACITY                         ITE00800
C                                                                       ITE00810
       SP=ABS(PI/PY)                                                    ITE00820
       IF(SP .LE. 0.4) THEN                                             ITE00830
          AA=0.06                                                       ITE00840
          BB=1                                                          ITE00850
       ELSE IF(SP .GT. 0.4) THEN                                        ITE00860
          AA=0.15                                                       ITE00870
          BB=2                                                          ITE00880
       ENDIF                                                            ITE00890
       BETA=1.7-(SP/LOG(SP))-AA*TNAX*(SP)**BB                           ITE00900
       IF( BETA .LT. 1.1)BETA=1.1                                       ITE00910
       TMPCX=TMNX*(1-(PIC/PN))*(1-(1.25*PIC/(PEX*(B/H)**(1./3.))))      ITE00920
       IF(TMPCX .LT. 0 ) TMPCX=0.05*TMNX                                ITE00930
       TMPCY=TMPY*(1-(PIC/PN))*(1-(1.25*PIC/(PEY*(B/H)**(1./2.))))      ITE00940
       IF(TMPCY .LT. 0 ) TMPCY=0.05*TMPY                                ITE00950
       IF(ABS(TMYI) .GT. ABS(TMYJ)) THEN                                ITE00960
          RMY=ABS(TMYI)                                                 ITE00970
       ELSE IF(ABS(TMYI) .LE. ABS(TMYJ)) THEN                           ITE00980
          RMY=ABS(TMYJ)                                                 ITE00990
       ENDIF                                                            ITE01000
       RMX=ABS(TM2)                                                     ITE01010
C                                                                       ITE01020
       VALUE=(RMX/TMPCX)**BETA + (RMY/TMPCY)**BETA                      ITE01030
       IF( VALUE .GE. 1 ) THEN                                          ITE01040
         IFAG=1                                                         ITE01050
       ENDIF                                                            ITE01060
C                                                                       ITE01070
       RETURN                                                           ITE01080
       END                                                              ITE01090
C@PROCESS SOUMP GOSTMT XREF                                             ITE01010
       SUBROUTINE ITERS2(PI,TMXI,TMXJ,TMPX,PY,TMPXI,TMPXJ,             ITE00020
      &                  TMYI,TMYJ,TMPY,   TMPYI,TMPYJ,IAFAG )        ITE00030
C     ------------------------------------------------------------      ITE00040
C     THIS SUBROUTINE IS FOR JUDGING BOX SECTION STRENGTH CRITERIA      ITE00050
C     BASED ON W.F.C'S PAPER                                            ITE00060
C     NOTE:                                                             ITE00070
C      PI    : CURRENT AXIAL LOAD                                       ITE00080
C      PY    : AXIAL YIELDING LOAD CAPACITY                             ITE00090
C      TMXI  : CURRENT MOMENT IN X DIRECTION AT I END                   ITE00100
C      TMXJ  : CURRENT MOMENT IN X DIRECTION AT J END                   ITE00110
C      TMYI  : CURRENT MOMENT IN Y DIRECTION AT I END                   ITE00120
C      TMYJ  : CURRENT MOMENT IN Y DIRECTION AT J END                   ITE00130
C      TMPX  : FULLY PLASTIC MOMENT IN X DIRECTION                      ITE00140
C      TMPY  : FULLY PLASTIC MOMENT IN Y DIRECTION                      ITE00150
C      TMPXI : CURRENT REDUCED PLASTIC MOMENT IN X DIRECTION AT I END   ITE00160
C      TMPXJ : CURRENT REDUCED PLASTIC MOMENT IN X DIRECTION AT J END   ITE00170
C      TMPYI : CURRENT REDUCED PLASTIC MOMENT IN Y DIRECTION AT I END   ITE00180
C      TMPYJ : CURRENT REDUCED PLASTIC MOMENT IN Y DIRECTION AT J END   ITE00190
C      IAFAG : FLAG FOR JUDGING FULLY PLASTIC SECTION                   ITE00200
C              5: MEMBER TWO ENDS STILL ARE NOT FULLY PLASTIC           ITE00210
C              6: MEMBER A END IS IN FULLY PLASTIC                      ITE00220
C              7: MEMBER B END IS IN FULLY PLASTIC                      ITE00230
C              8: MEMBER A AND B ENDS ARE IN FULLY PLASTIC              ITE00240
C     ------------------------------------------------------------      ITE00250
C                                                                       ITE00260
       IF( PI .GE. 0 ) RETURN                                           ITE00270
       SP=ABS(PI/PY)                                                    ITE00280
       IF( SP .EQ. 0 ) THEN                                             ITE00290
          RETURN                                                        ITE00300
       ELSE IF( SP .LT. 1 .AND. SP .GT. 0 ) THEN                        ITE00310
          A1=1.7-(SP/LOG(SP))                                           ITE00320
       ELSE IF( SP .GE. 1 ) THEN                                        ITE00330
          IAFAG=4                                                       ITE00340
          RETURN                                                        ITE00350
       ENDIF                                                            ITE00360
       TMPCXI=1.2*(1-SP)*TMPX                                           ITE00370
       IF(ABS(TMPCXI) .GT. ABS(TMPX)) TMPCXI=TMPX                       ITE00380
       TMPCXJ=TMPCXI                                                    ITE00390
       TMPCYI=1.2*(1-SP)*TMPY                                           ITE00400
       IF(ABS(TMPCYI) .GT. ABS(TMPY)) TMPCYI=TMPY                       ITE00410
       TMPCYJ=TMPCYI                                                    ITE00420
       RMXI=ABS(TMXI)                                                   ITE00430
       RMXJ=ABS(TMXJ)                                                   ITE00440
       RMYI=ABS(TMYI)                                                   ITE00450
       RMYJ=ABS(TMYJ)                                                   ITE00460
C                                                                       ITE00470
       IF( IAFAG .EQ. 5 ) THEN                                          ITE00480
          IF( (RMXI/TMPCXI) .GE. 1 .OR. (RMYI/TMPCYI) .GE. 1 ) THEN    ITE00490
              YIELD1=1                                                  ITE00500
          ELSE IF((RMXI/TMPCXI) .LT. 1 .AND. (RMYI/TMPCYI) .LT. 1 ) THEN ITE00510
             YIELD1=(RMXI/TMPCXI)**A1 + (RMYI/TMPCYI)**A1               ITE00520
          ENDIF                                                         ITE00530
          IF( (RMXJ/TMPCXJ) .GE. 1 .OR. (RMYJ/TMPCYJ) .GE. 1 ) THEN     ITE00540
              YIELD2=1                                                  ITE00550
          ELSE IF((RMXJ/TMPCXJ) .LT. 1 .AND. (RMYJ/TMPCYJ) .LT. 1 ) THEN ITE00560
             YIELD2=(RMXJ/TMPCXJ)**A1 + (RMYJ/TMPCYJ)**A1               ITE00570
          ENDIF                                                         ITE00580
C                                                                       ITE00590
          IF( YIELD1 .GE. 1 .AND. YIELD2 .LT. 1 ) THEN                  ITE00600
             IAFAG=6                                                    ITE00610
          ELSE IF( YIELD1 .LT. 1 .AND. YIELD2 .GE. 1 ) THEN             ITE00620
             IAFAG=7                                                    ITE00630
          ELSE IF( YIELD1 .GE. 1 .AND. YIELD2 .GE. 1 ) THEN             ITE00640
             IAFAG=8                                                    ITE00650
          ENDIF                                                         ITE00660
C                                                                       ITE00670
       ENDIF                                                            ITE00680
C                                                                       ITE00690
       RETURN                                                           ITE00700
       END                                                              ITE00710
C=======================================================================JOI00010
C        DEBUG UNIT(6),SUBCHK,SUBTRACE                                  JOI00020
C        END DEBUG                                                      JOI00030
       SUBROUTINE JOIDSP(IOPT,MAXNOD,NDOF,NNODE,NLOAD,NCIN,NF,ID,IDOF,  JOI00040
      &    CONST,DISP,TITLE,NAME,STEPID,HEAD,NCOS.COSINE,JTCOS,JTFLG)    JOI00050
       COMMON /IEPEC/ ISP1,SSD(3000,2),REACF(6)                         JOI00060
       DIMENSION IDOF(MAXNOD,6)  ,CONST(MAXNOD,6),JTFLG(MAXNOD)         JOI00070
       DIMENSION DISP(NDOF,NLOAD)                                       JOI00080
       DIMENSION ID(MAXNOD)                                             JOI00090
       DIMENSION D(6),G(6),COSINE(3,3,NCOS),JTCOS(MAXNOD)               JOI00100
       CHARACTER*(*) TITLE(2),STEPID,NAME                               JOI00110
       CHARACTER*15 CD(6)                                               JOI00120
       CHARACTER*60 EQUAL                                               JOI00130
       LOGICAL HEAD,FLAG,BTEST                                          JOI00140
       INTEGER DOF                                                      JOI00150
       DATA EQUAL                                                       JOI00160
      & /'============================================================'/JOI00170
C                                                                       JOI00180
       NLEN=MIN(LEN(NAME)+1,60)                                         JOI00190
       NC=1                                                             JOI00200
C                                                                       JOI00210
C-----------------------------                                          JOI00220
C- LOOP FOR EACH LOAD CASE  ---                                         JOI00230
C-----------------------------                                          JOI00240
C                                                                       JOI00250
       DO 300 L=1,NLOAD                                                 JOI00260
         NDSPLS=0                                                       JOI00270
C                                                                       JOI00280
         IF (.NOT.(NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND.             JOI00290
      &       COSINE(2,2,1).EQ.1  .AND. COSINE(3,3,1).EQ.1 )            JOI00300
      &       .AND. IOPT.EQ.2 ) GO TO 210                               JOI00310
C                                                                       JOI00320
C-------------------------------                                        JOI00330
C- PRINT GLOBAL DISPLACEMENTS ---                                       JOI00340
C-------------------------------                                        JOI00350
C  NOTE:DISP MAY CONTAIN DISPLACEMENTS, VELOCITIES, ACCELERATIONS OR    JOI00360
C       EIGENVALUES...                                                  JOI00370
C       ALL OF WHICH ARE PRINTED OUT AT EACH NODE....                   JOI00380
C                                                                       JOI00390
         IF (HEAD) WRITE (6,310) TITLE(1),TITLE(2)                      JOI00400
         IF (IOPT.NE.2) WRITE (6,315) NAME,L,STEPID,EQUAL(1:NLEN)       JOI00410
         IF (IOPT.EQ.2) WRITE (6,415) NAME,          EQUAL(1:NLEN)      JOI00420
         DO 100 I=1,NNODE                                               JOI00430
C                                                                       JOI00440
C.............. TRANSFER DISPL TO LOCAL VECTOR ..............          JOI00450
           DO 20 J=1,6                                                  JOI00460
             IJ=IDOF(I,J)                                               JOI00470
   20        D(J)=DISP(IJ,L)                                            JOI00480
C                                                                       JOI00490
```

```
C..... ADD ROTATIONS TO DISPL FOR CONSTRAINTS                          JOI00500
C..... IF UNCONSTRAINED, CONST(I,J)=0                                  JOI00510
          D(1) = D(1) + D(5)*CONST(I,3) + D(6)*CONST(I,5)              JOI00520
          D(2) = D(2) + D(4)*CONST(I,1) + D(6)*CONST(I,6)              JOI00530
          D(3) = D(3) + D(4)*CONST(I,2) + D(5)*CONST(I,4)              JOI00540
C                                                                      JOI00550
C..... TRANSFER TO GLOBAL DISPLACEMENTS .....                         JOI00560
          JCOS=JTCOS(I)                                                JOI00570
          DO 25 I1=1,3                                                 JOI00580
            I2=I1+3                                                    JOI00590
            G(I1)=0                                                    JOI00600
            G(I2)=0                                                    JOI00610
            DO 25 J1=1,3                                               JOI00620
               J2=J1+3                                                 JOI00630
               G(I1)=G(I1)+COSINE(J1,I1,JCOS)*D(J1)                    JOI00640
   25          G(I2)=G(I2)+COSINE(J1,I1,JCOS)*D(J2)                    JOI00650
C                                                                      JOI00660
C..... WRITE DISPL                                                     JOI00670
          DO 30 J=1,6                                                  JOI00680
   30          WRITE (CD(J),9) G(J)                                    JOI00690
    9     FORMAT (1P,G14.6,' ')                                        JOI00700
         WRITE (6,220) ID(I),(CD(K),K=1,6)                             JOI00710
C                                                                      JOI00720
C         IF( ISP1 .EQ. 2 ) THEN                                       JOI00730
C         DO 399 K=1,6                                                 JOI00740
C 399       SGD(ID(I),K)=G(K)                                          JOI00750
C         ENDIF                                                        JOI00760
C                                                                      JOI00770
          NDSPLS=NDSPLS+1                                              JOI00780
  100 CONTINUE                                                         JOI00790
C                                                                      JOI00800
      IF (NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND. COSINE(2,2,1).EQ.1  JOI00810
     &               .AND. COSINE(3,3,1).EQ.1 ) GO TO 300              JOI00820
C                                                                      JOI00830
C--------------------------------                                      JOI00840
C- PRINT JOINT DISPLACEMENTS ---                                       JOI00850
C--------------------------------                                      JOI00860
C  NOTE:DISP MAY CONTAIN DISPLACEMENTS, VELOCITIES, ACCELERATIONS OR   JOI00870
C       EIGENVALUES...                                                 JOI00880
C       ALL OF WHICH ARE PRINTED OUT AT EACH NODE....                  JOI00890
C                                                                      JOI00900
  210   IF (HEAD.AND. NDSPLS.GT.20) WRITE (6,310) TITLE(1),TITLE(2)    JOI00910
        IF (IOPT.NE.2) WRITE (6,316) NAME,L,STEPID,EQUAL(1:NLEN)       JOI00920
        IF (IOPT.EQ.2) WRITE (6,416) NAME,          EQUAL(1:NLEN)      JOI00930
        DO 200 I=1,NNODE                                               JOI00940
C                                                                      JOI00950
C..... TRANSFER GLOBAL DISPL TO LOCAL VECTOR .....                    JOI00960
          DO 219 J=1,6                                                 JOI00970
            IJ=IDOF(I,J)                                               JOI00980
  219       D(J)=DISP(IJ,L)                                            JOI00990
C                                                                      JOI01000
C..... ADD ROTATIONS TO DISPL FOR CONSTRAINTS                          JOI01010
C..... IF UNCONSTRAINED, CONST(I,J)=0                                  JOI01020
          D(1) = D(1) + D(5)*CONST(I,3) + D(6)*CONST(I,5)              JOI01030
          D(2) = D(2) + D(4)*CONST(I,1) + D(6)*CONST(I,6)              JOI01040
          D(3) = D(3) + D(4)*CONST(I,2) + D(5)*CONST(I,4)              JOI01050
C                                                                      JOI01060
C..... WRITE DISPL                                                     JOI01070
          DO 230 J=1,6                                                 JOI01080
             WRITE (CD(J),9) D(J)                                      JOI01090
             IJ=IDOF(I,J)                                              JOI01100
  230        IF (IJ.GT.NF) CD(J)(15:15)='*'                            JOI01110
        WRITE (6,221) ID(I),(CD(K),K=1,6),JTCOS(I)                     JOI01120
C                                                                      JOI01130
C         IF( ISP1 .EQ. 2 ) THEN                                       JOI01140
C         DO 388 K=1,6                                                 JOI01150
C 388       SGD(ID(I),K)=D(K)                                          JOI01160
C         ENDIF                                                        JOI01170
C                                                                      JOI01180
  200 CONTINUE                                                         JOI01190
      WRITE (6,320)                                                    JOI01200
C                                                                      JOI01210
  300 CONTINUE                                                         JOI01220
C                                                                      JOI01230
      RETURN                                                           JOI01240
C                                                                      JOI01250
   10 FORMAT (1P,G15.7)                                                JOI01260
  220 FORMAT (5X,I5,6A)                                                JOI01270
  221 FORMAT (5X,I5,6A,I10)                                            JOI01280
  310 FORMAT ('1 STRUCTURE....: ',A,/                                  JOI01290
     &        '  SOLUTION.....: ',A)                                   JOI01300
  315 FORMAT (//                                                       JOI01310
     &        2X, 'GCS ',A,',  LOADING #',I5,5X,A, /                  JOI01320
     &        1X,'=====',A//                                          JOI01330
     &  6X,'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ'/)JOI01340
  316 FORMAT (//                                                       JOI01350
     &        2X, 'JCS ',A,',  LOADING #',I5,5X,A, /                  JOI01360
     &        1X,'=====',A//                                          JOI01370
     &  6X,'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ', JOI01380
     &  6X,'  COSINE #'/)                                              JOI01390
  415 FORMAT (//                                                       JOI01400
     &        2X, 'GCS ',A/                                           JOI01410
     &        1X,'=====',A/                                           JOI01420
     &      25X,'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'//  JOI01430
     &  6X,'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ'/)JOI01440
  416 FORMAT (//                                                       JOI01450
     &        2X, 'JCS ',A/                                           JOI01460
     &        1X,'=====',A/                                           JOI01470
     &      25X,'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'//  JOI01480
     &  6X,'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ', JOI01490
     &  6X,'  COSINE #'/)                                              JOI01500
  320 FORMAT (4X,'NOTE: * DENOTES A RESTRAINED DEGREE OF FREEDOM')     JOI01510
C                                                                      JOI01520
      END                                                              JOI01530
C---------------------------------------------------------------------JOI00010
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                     JOI00020
C         END DEBUG                                                    JOI00030
C---------------------------------------------------------------------JOI00040
      SUBROUTINE JOILOA(BUG,MAXNOD,NDOF,NNODE.NLOAD,NF,ID,IDOF,CONST,  JOI00050
     &    FORCE,DSPFLG,JTFLG,TITLE,HEAD)                               JOI00060
      DIMENSION IDOF(MAXNOD,6),CONST(MAXNOD,6),FORCE(NDOF,NLOAD)       JOI00070
      DIMENSION ID(MAXNOD),JTFLG(MAXNOD)                               JOI00080
      CHARACTER*(*) TITLE(2)                                           JOI00090
      CHARACTER*4  DIR                                                 JOI00100
      CHARACTER*1  CX(6)                                               JOI00110
      CHARACTER*25 SUPT                                                JOI00120
      CHARACTER*18 DOF                                                 JOI00130
      CHARACTER*14 F(6)                                                JOI00140
      LOGICAL DSPFLG,FIRST,BUG,HEAD                                    JOI00150
      LOGICAL BTEST                                                    JOI00160
      LOAD=1                                                           JOI00170
      FIRST =.TRUE.                                                    JOI00180
      DSPFLG=.FALSE.                                                   JOI00190
    5 FORMAT ('1 STRUCTURE....: ',A,/                                  JOI00200
     &        '  SOLUTION.....: ',A,/)                                 JOI00210
    6 FORMAT ('    APPLIED JOINT LOADS',/                              JOI00220
     &        '    --------------------',/)                           JOI00230
C                                                                      JOI00240
C-----------------------                                               JOI00250
C- INPUT LOADING DATA --                                               JOI00260
C-----------------------                                               JOI00270
   10 READ (5,*) LOAD,JOINT,JTGEN,JTINC,DIR,VALUE                     JOI00280
      IF (INDEX(DIR,'END').NE.0) GO TO 100                             JOI00290
   15 DOF=' '                                                          JOI00300
      IF (FIRST) THEN                                                  JOI00310
         IF (HEAD) WRITE (6,5) TITLE(1),TITLE(2)                       JOI00320
         WRITE (6,6)                                                   JOI00330
         FIRST=.FALSE.                                                 JOI00340
      ENDIF                                                            JOI00350
C                                                                      JOI00360
C------- CHECK FOR A VALID JOINT NUMBER...                             JOI00370
      IF (JOINT.LT.ID(1) .OR. JOINT.GT.ID(NNODE) ) THEN                JOI00380
         WRITE (6,21) LOAD                                             JOI00390
         GO TO 10                                                      JOI00400
      ENDIF                                                            JOI00410
C                                                                      JOI00420
      J=IQUICK(JOINT,ID,NNODE)                                         JOI00430
      SUPT=' '                                                         JOI00440
C                                                                      JOI00450
C---------------------------------------------------------------------JOI00460
C------- PUT A LOAD ON A JOINT                                         JOI00470
C NOTES:  - IX, IY, IZ, MX, MY AND Z ARE THE DOF THAT THE LOADS ARE    JOI00480
C           APPLIED TO.                                                JOI00490
C         - THE MOMENTS RESULTING FROM FX, FY AND FZ ARE DUE TO        JOI00500
C           CONSTRAINTS.  IF A NODE IS NOT CONSTRAINED, THEN THE VALUE JOI00510
C           OF CONST IS ZERO....                                       JOI00520
C         - LOADS APPLIED TO RESTRAINED DOF ARE INTERPERATED AS        JOI00530
C           SUPPORT DISPLACEMENTS. IE: SUPPORT SETTELMENT OR           JOI00540
C           DISPLACEMENT CONTROLL.....                                 JOI00550
C---------------------------------------------------------------------JOI00560
      IF (INDEX(DIR,'FX').NE.0) THEN                                   JOI00570
         IX=IDOF(J,1)                                                  JOI00580
         FORCE(IX,LOAD)=FORCE(IX,LOAD)+VALUE                           JOI00590
         IF ( BTEST(JTFLG(J),12) ) THEN                                JOI00600
           MY=IDOF(J,5)                                                JOI00610
           FORCE(MY,LOAD)=FORCE(MY,LOAD)+VALUE*CONST(J,3)              JOI00620
           MZ=IDOF(J,6)                                                JOI00630
           FORCE(MZ,LOAD)=FORCE(MZ,LOAD)+VALUE*CONST(J,5)              JOI00640
           WRITE (DOF,35) IX,MY,MZ                                     JOI00650
         ELSE                                                          JOI00660
           WRITE (DOF,35) IX                                           JOI00670
         ENDIF                                                         JOI00680
         IF (IX.GT.NF) THEN                                            JOI00690
           DSPFLG=.TRUE.                                               JOI00700
           SUPT='...JOINT DISPLACEMENT...'                             JOI00710
         ENDIF                                                         JOI00720
      ELSE IF (INDEX(DIR,'FY').NE.0) THEN                              JOI00730
         IY=IDOF(J,2)                                                  JOI00740
         FORCE(IY,LOAD)=FORCE(IY,LOAD)+VALUE                           JOI00750
         IF (BTEST(JTFLG(J),13)) THEN                                  JOI00760
           MX=IDOF(J,4)                                                JOI00770
           FORCE(MX,LOAD)=FORCE(MX,LOAD)+VALUE*CONST(J,1)              JOI00780
           MZ=IDOF(J,6)                                                JOI00790
           FORCE(MZ,LOAD)=FORCE(MZ,LOAD)+VALUE*CONST(J,6)              JOI00800
           WRITE (DOF,35) IY,MX,MZ                                     JOI00810
         ELSE                                                          JOI00820
           WRITE (DOF,35) IY                                           JOI00830
         ENDIF                                                         JOI00840
         IF (IY.GT.NF) THEN                                            JOI00850
           DSPFLG=.TRUE.                                               JOI00860
           SUPT='...JOINT DISPLACEMENT...'                             JOI00870
         ENDIF                                                         JOI00880
      ELSE IF (INDEX(DIR,'FZ').NE.0) THEN                              JOI00890
         JZ=IDOF(J,3)                                                  JOI00900
         FORCE(JZ,LOAD)=FORCE(JZ,LOAD)+VALUE                           JOI00910
         IF (BTEST(JTFLG(J),14)) THEN                                  JOI00920
           MX=IDOF(J,4)                                                JOI00930
           FORCE(MX,LOAD)=FORCE(MX,LOAD)+VALUE*CONST(J,2)              JOI00940
           MY=IDOF(J,5)                                                JOI00950
           FORCE(MY,LOAD)=FORCE(MY,LOAD)+VALUE*CONST(J,4)              JOI00960
           WRITE (DOF,35) JZ,MX,MY                                     JOI00970
         ELSE                                                          JOI00980
           WRITE (DOF,35) JZ                                           JOI00990
         ENDIF                                                         JOI01000
         IF (JZ.GT.NF) THEN                                            JOI01010
           DSPFLG=.TRUE.                                               JOI01020
           SUPT='...JOINT DISPLACEMENT...'                             JOI01030
         ENDIF                                                         JOI01040
      ELSE IF (INDEX(DIR,'MX').NE.0) THEN                              JOI01050
         MX=IDOF(J,4)                                                  JOI01060
         FORCE(MX,LOAD)=FORCE(MX,LOAD)+VALUE                           JOI01070
         WRITE (DOF,35) MX                                             JOI01080
         IF (MX.GT.NF) THEN                                            JOI01090
           DSPFLG=.TRUE.                                               JOI01100
           SUPT='...JOINT DISPLACEMENT...'                             JOI01110
         ENDIF                                                         JOI01120
      ELSE IF (INDEX(DIR,'MY').NE.0) THEN                              JOI01130
         MY=IDOF(J,5)                                                  JOI01140
         FORCE(MY,LOAD)=FORCE(MY,LOAD)+VALUE                           JOI01150
         WRITE (DOF,35) MY                                             JOI01160
         IF (MY.GT.NF) THEN                                            JOI01170
           DSPFLG=.TRUE.                                               JOI01180
           SUPT='...JOINT DISPLACEMENT...'                             JOI01190
         ENDIF                                                         JOI01200
      ELSE IF (INDEX(DIR,'MZ').NE.0) THEN                              JOI01210
         MZ=IDOF(J,6)                                                  JOI01220
         FORCE(MZ,LOAD)=FORCE(MZ,LOAD)+VALUE                           JOI01230
         WRITE (DOF,35) MZ                                             JOI01240
         IF (MZ.GT.NF) THEN                                            JOI01250
           DSPFLG=.TRUE.                                               JOI01260
           SUPT='...JOINT DISPLACEMENT...'                             JOI01270
         ENDIF                                                         JOI01280
      ELSE                                                             JOI01290
         WRITE (6,20) JOINT,DIR,VALUE                                  JOI01300
      ENDIF                                                            JOI01310
      WRITE (6,30) LOAD,JOINT,DIR,DOF,VALUE,SUPT                       JOI01320
      IF (JTGEN.GT.0) THEN                                             JOI01330
         JTGEN=JTGEN-1                                                 JOI01340
         JOINT=JOINT+JTINC                                             JOI01350
         GO TO 15                                                      JOI01360
      ENDIF                                                            JOI01370
      GO TO 10                                                         JOI01380
C                                                                      JOI01390
   20 FORMAT ('  INVALID JOINT LOAD... ',I5,2X,A,2X,1P,G14.6,          JOI01400
     &     ' --- LOAD IS SKIPPED')                                     JOI01410
   21 FORMAT ('  INVALID LOAD CASE ... ',I6,                          JOI01420
     &     ' --- LOAD IS SKIPPED')                                     JOI01430
   30 FORMAT (' LOAD CASE:',I3,'  JOINT:',I6,'  DIRECTION: ',A,       JOI01440
     &  ' DOF(S)',A,'  MAGNITUDE:',1P,G14.6,2X,A)                      JOI01450
   35 FORMAT (I5,',',',I5,',',',I5)                                    JOI01460
C                                                                      JOI01470
C------- RETURN IF NO JOINT LOADS WERE INPUT                           JOI01480
  100 IF (FIRST) RETURN                                                JOI01490
C                                                                      JOI01500
      IF (.NOT.BUG) RETURN                                             JOI01510
C-------- PRINT JOINT LOADS ON EACH DOF                                JOI01520
      N5=NDOF*5                                                        JOI01530
      DO 120 L=1,NLOAD                                                 JOI01540
        IF (HEAD) WRITE (6,5  ) TITLE(1),TITLE(2)                      JOI01550
        WRITE (6,130) L                                                JOI01560
        DO 120 I=1,N5,5                                                JOI01570
        I5=MIN((I+4),NDOF)                                             JOI01580
        DO 110 J=I,I5                                                  JOI01590
          K=J-I+1                                                      JOI01600
          CX(K)=' '                                                    JOI01610
 110      IF (J.GT.NF) CX(K)='*'                                       JOI01620
          WRITE (6,140) (J,FORCE(J,L),CX(J-I+1),J=I,I5)                JOI01630
  120 CONTINUE                                                         JOI01640
C                                                                      JOI01650
  130 FORMAT ('  APPLIED JOINT LOADS,   LOADING #',I5,10X,            JOI01660
     &        ' NOTE: * DENOTES SUPPORT DISPLACEMENT'/                 JOI01670
     &        ' ---------------------- '//                            JOI01680
     &  5X,5('  DOF',7X,'LOAD     '))                                 JOI01690
  140   FORMAT (5X,5(0P,I5,1P,G14.6,A) )                               JOI01700
C                                                                      JOI01710
      DO 220 L=1,NLOAD                                                 JOI01720
        IF (HEAD) WRITE (6,5  ) TITLE(1),TITLE(2)                      JOI01730
        WRITE (6,200) L                                                JOI01740
        DO 220 I=1,NNODE                                               JOI01750
        DO 210 J=1,6                                                   JOI01760
          CX(J)=' '                                                    JOI01770
          IF (BTEST(JTFLG(I),J+11)) THEN                               JOI01780
            F(J)=' '                                                   JOI01790
          ELSE                                                         JOI01800
```

```
            K=IDOF(I,J)                                                 JOI01810
            WRITE (F(J),240) FORCE(K,L)                                 JOI01820
            IF (K.GT.NF) CX(J)='*'                                      JOI01830
          ENDIF                                                         JOI01840
 210      CONTINUE                                                      JOI01850
          WRITE (6,230) ID(I),(F(J),CX(J),J=1,6)                        JOI01860
  220 CONTINUE                                                          JOI01870
C                                                                       JOI01880
  200 FORMAT ('  APPLIED JOINT LOADS,   LOADING #',I5,10X,             JOI01890
     &        '  NOTE: * DENOTES SUPPORT DISPLACEMENT'/                 JOI01900
     &        ' ************************ '//                           JOI01910
     &  6X,'NODE',7X,'FX',13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)JOI01920
  230   FORMAT (5X,I5,12A )                                           JOI01930
  240   FORMAT (1P,G14.6)                                             JOI01940
C                                                                       JOI01950
      RETURN                                                            JOI01960
      END                                                               JOI01970
C  - KMIN - CALCULATES STIFFNESS BASED ON DX AND DY ....................KMI00010
C        DEBUG UNIT(6),SUBCHK,SUBTRACE                                  KMI00020
C        END DEBUG                                                      KMI00030
C     KMIN    =                                                         KMI00040
C     DD      = CHANGE IN DISPLACMENT                                   KMI00050
C     DP      = CHANGE IN LOAD                                          KMI00060
C     SI      = MAXIMUN AND DEFULT STIFFNESS                            KMI00070
      REAL FUNCTION KMIN(DD,DP,SI)                                      KMI00080
      LOGICAL BTEST                                                     KMI00090
          IF ((DD*DP).GT. 0.00)  THEN                                   KMI00100
            KMIN=MIN((DP/DD),SI)                                        KMI00110
          ELSE                                                          KMI00120
            KMIN=SI                                                     KMI00130
          ENDIF                                                         KMI00140
      RETURN                                                            KMI00150
      END                                                               KMI00160
C-----------------------------------------------------------------------LIN00010
C        DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT                             LIN00020
C        END DEBUG                                                      LIN00030
      SUBROUTINE LINACC(IOPT,BUG,DT,TO,TF,T,THETA,NEWKDT,               LIN00040
     &    L1,L2,L3,L4,L5,L6,IMDMAS,IMD,IMDS,IMDKG,NDOF,NFREE,           LIN00050
     &    KGLOAD,KGFORM,                                                LIN00060
     &     A     ,V     ,D     ,P     ,                                LIN00070
     &     DA    ,DV    ,DD    ,DP    ,WORK  ,                         LIN00080
     &     MASS  ,MDMASS,DAMP  ,STIFF ,MD    ,                         LIN00090
     &     KG    ,MDKG  ,ACC   ,GRAV  ,                                LIN00100
     &     S     ,MDS   ,Q     ,R     ,PBAR  ,                         LIN00110
     &     FDAMP,FMASS,C0,C1,C2,C3,C4,C5,C6,ALPHA,BETA,C10,C11,ABORT)   LIN00120
      INTEGER FDAMP,FMASS                                               LIN00130
      REAL MASS,KG                                                      LIN00140
      LOGICAL BUG,NEWKDT,BTEST,ABORT                                    LIN00150
      DIMENSION A(NDOF),V(NDOF),D(NDOF),DA(NDOF),DV(NDOF),DD(NDOF)      LIN00160
      DIMENSION P(NDOF),DP(NDOF),DUMMY(1),IDUMMY(1),WORK(2*NDOF)        LIN00170
      DIMENSION PBAR(L3:L4),Q(L3:L4),R(L3:L4)                           LIN00180
      DIMENSION MDMASS(NDOF+1),MASS(IMDMAS)                             LIN00190
      DIMENSION MDKG(NDOF+1)   ,KG(IMDKG)                               LIN00200
      DIMENSION MD(NDOF+1)     ,STIFF(IMD)                              LIN00210
      DIMENSION MDS(NDOF+1)    ,S(IMDS)     ,DAMP(3)                    LIN00220
C                                                                       LIN00230
      GO TO (100,200,300) IOPT                                          LIN00240
C                                                                       LIN00250
C=======================                                                LIN00260
C= INITIALIZE VALUES ===                                                LIN00270
C=======================                                                LIN00280
  100 CONTINUE                                                          LIN00290
      ALPHA=DAMP(1)                                                     LIN00300
      BETA =DAMP(2)                                                     LIN00310
      C0=3.*ALPHA/DT + 6./(DT**2)                                       LIN00320
      C1=1.D/(1+3.*BETA/DT)                                             LIN00330
      C2=C0*C1                                                          LIN00340
      C3=ALPHA-C2*BETA                                                  LIN00350
      C4=6./DT                                                          LIN00360
      C5=DT/2.                                                          LIN00370
      C6=0.                                                             LIN00380
      C10=3./DT                                                         LIN00390
      C11=6./(DT**2)                                                    LIN00400
      IF (BUG) WRITE (6,101) ALPHA,BETA,C0,C1,C2,C3,C4,C5,C6,C10,C11    LIN00410
  101 FORMAT (1P,                                                       LIN00420
     &' LINACC: ALPHA=',G12.5,' BETA=',G12.5,'  C0=',G12.5,' C1=',G12.5/LIN00430
     &' LINACC:     C2=',G12.5,'   C3=',G12.5,'  C4=',G12.5,' C5=',G12.5/LIN00440
     &' LINACC:     C6=',G12.5,'  C10=',G12.5,' C11=',G12.5)           LIN00450
C                                                                       LIN00460
C------- DEVELOP SKYLINE OF DYNAMIC STIFFNESS (S)                       LIN00470
C....... MDS(I)    = ADDRESS OF DYNAMIC STIFF. MAIN DIAGONAL TERM ROW I LIN00480
C....... MD(I)     = ADDRESS OF STIFFNESS MAIN DIAGONAL TERM ROW I      LIN00490
C....... MDMAS(I) = ADDRESS OF MASS MAIN DIAGONAL TERM ROW I           LIN00500
C....... MDKG(I)   = ADDRESS OF GEOMETRIC STIFF. MAIN DIAGONAL TERM ROW ILIN00510
C                                                                       LIN00520
C....... LSTIFF = # OF STIFFNESS TERMS IN COLUMN J                      LIN00530
C....... LMASS  = # OF MASS TERMS IN COLUMN J                           LIN00540
C....... LKG    = # OF KG TERMS IN COLUMN J                             LIN00550
C....... L      = # OF S  TERMS IN COLUMN J                             LIN00560
C                                                                       LIN00570
C     MDS(1)=1                                                          LIN00580
C     LKG=0                                                             LIN00590
C     DO 110 J1=2,NFREE+1                                               LIN00600
C       J=J1+L3-1                                                       LIN00610
C       LSTIFF=MD(J)-MD(J-1)                                            LIN00620
C       LMASS =MDMASS(J)-MDMASS(J-1)                                    LIN00630
C       IF (KGFORM.EQ.2)  LKG = MDKG(J)-MDKG(J-1)                       LIN00640
C       L=MIN(J1-1,MAX(LSTIFF,LMASS,LKG))                               LIN00650
C 110   MDS(J1)=MDS(J1-1)+L                                            LIN00660
C                                                                       LIN00670
C------ CHECK FOR PROPORTIONAL DAMPING ....                            LIN00680
      IF (FDAMP.NE.1) THEN                                              LIN00690
        WRITE (6,*) '*********************************************'     LIN00700
        WRITE (6,*) '*  PROPORTIONAL DAMPING WAS NOT SPECIFIED   *'     LIN00710
        WRITE (6,*) '*  SOLUTION IS ABORTED...                   *'     LIN00720
        WRITE (6,*) '*********************************************'     LIN00730
        STOP                                                            LIN00740
      ENDIF                                                             LIN00750
      RETURN                                                            LIN00760
C                                                                       LIN00770
C-----------------------------------------------------------------------LIN00780
C= SOLVE FOR INCRIMENTAL DISPLACEMENTS, VELOCITIES AND ACCELERATIONS ==LIN00790
C-----------------------------------------------------------------------LIN00800
  200 CONTINUE                                                          LIN00810
      IF (BUG) THEN                                                     LIN00820
C       CALL ERRTRA                                                     LIN00830
        DO 205 I=L3,L4                                                  LIN00840
  205   WRITE (6,206) I,D(I),V(I),A(I),P(I),DP(I)                       LIN00850
  206   FORMAT (' DOF:',I6,'  DISP:',1P,G12.4,'  VEL:',G12.4,           LIN00860
     &    '  ACC:',G12.4,',','  LOAD:',  G12.4,'  DP:',G12.4)         LIN00870
      ENDIF                                                             LIN00880
C                                                                       LIN00890
C------- FORM DYNAMIC STIFFNESS                                         LIN00900
C....... SOME LOCAL VARIABLES                                           LIN00910
C J     = COLUMN NUMBER FOR MASS AND STIFF                              LIN00920
C I     = ROW NUMBER FOR MASS AND STIFF                                 LIN00930
C J1    = COLUMN NUMBER FOR S                                           LIN00940
C I1    = ROW NUMBER FOR S                                              LIN00950
C IJSTIF = ADDRESS OF ELEMENT I,J IN MATRIX STIFF                       LIN00960
C IJMASS = ADDRESS OF ELEMENT I,J IN MATRIX MASS                        LIN00970
C IJKG   = ADDRESS OF ELEMENT I,J IN GEOMETRIC STIFFNESS MATRIX         LIN00980
C IJS    = ADDRESS OF ELEMENT I1,J1 IN MATRIX S                         LIN00990
C LSTIFF = ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF STIFF LIN01000
C LMASS  = ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF MASS  LIN01010
C LKG    = ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF KG    LIN01020
C LS     = ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF S     LIN01030
C                                                                       LIN01040
C-------- FORM DYNAMIC STIFFNESS IF NEWKDT IS TRUE ---------------------LIN01050
      IF (NEWKDT) THEN                                                  LIN01060
C                                                                       LIN01070
C     DETERMINE THE FACTOR FOR KG                                       LIN01080
      IF (KGLOAD.EQ.1) THEN                                             LIN01090
            CKG=(1+ACC)                                                 LIN01100
      ELSE IF (KGLOAD.EQ.2) THEN                                        LIN01110
            CKG= 1.00                                                   LIN01120
      ENDIF                                                             LIN01130
C                                                                       LIN01140
C---- K(BAR) = STIFFNESS                                                LIN01150
        DO 225 J=L3,L4                                                  LIN01160
          J1=J-L3+1                                                     LIN01170
          MDJP=MD(J+1)                                                  LIN01180
          MDJ =MD(J)                                                    LIN01190
          LSTIFF=J-(MD(J+1)-MD(J)-1)                                    LIN01200
          MDSJ1P=MDS(J1+1)                                              LIN01210
          MDSJ1 =MDS(J1)                                                LIN01220
          LS    =J1-(MDS(J1+1)-MDS(J1)-1)+L2                            LIN01230
          L=MAX(LS,LSTIFF)                                              LIN01240
          DO 210 I=J,L,-1                                               LIN01250
            I1=I-L3+1                                                   LIN01260
            IJS=(MDS(J1)+J1)-I1                                         LIN01270
            IJSTIF=(MD(J)+J)-I                                          LIN01280
  210       S(IJS)=STIFF(IJSTIF)                                        LIN01290
C                                                                       LIN01300
          IF (LS.LT.LSTIFF) THEN                                        LIN01310
            DO 215 I=L-1,LS,-1                                          LIN01320
              IJS=(MDS(J1)+J1)-I1                                       LIN01330
  215         S(IJS)=0                                                  LIN01340
          ENDIF                                                         LIN01350
C                                                                       LIN01360
C---- K(BAR) = K(BAR) + C2*MASS                                         LIN01370
          LMASS=J-(MDMASS(J+1)-MDMASS(J)-1)                             LIN01380
          L=MAX(LS,LMASS)                                               LIN01390
          DO 220 I=J,L,-1                                               LIN01400
            I1=I-L3+1                                                   LIN01410
            IJS=(MDS(J1)+J1)-I1                                         LIN01420
            IJMASS=(MDMASS(J)+J)-I                                      LIN01430
  220       S(IJS)=MASS(IJMASS)*C2  + S(IJS)                            LIN01440
C                                                                       LIN01450
          IF (KGFORM.EQ.2) THEN                                         LIN01460
C                                                                       LIN01470
C.          NOTE: THE GEOMETRIC STIFFNESS WAS DERIVED WITH COMPRESSION  LIN01480
C.          BEING POSITIVE, KG HAS POSITIVE TERMS ON THE MAIN DIAGIONAL LIN01490
C.          REPRESENTING COMPRESSION. THUS KG IS SUBTRACTED FROM S      LIN01500
C.          CKG IS  AN INCREASE FOR VERT ACCEL                          LIN01510
C                                                                       LIN01520
C---- K(BAR) = K(BAR) - CKG*GEOMETRIC STIFFNESS                         LIN01530
            LKG=J-(MDKG(J+1)-MDKG(J)-1)                                 LIN01540
            L=MAX(LS,LKG)                                               LIN01550
            DO 222 I=J,L,-1                                             LIN01560
              I1=I-L3+1                                                 LIN01570
              IJS=(MDS(J1)+J1)-I1                                       LIN01580
              IJKG=(MDKG(J)+J)-I                                        LIN01590
  222         S(IJS)=S(IJS) - KG(IJKG)*CKG                              LIN01600
          ENDIF                                                         LIN01610
C                                                                       LIN01620
  225   CONTINUE                                                        LIN01630
      ENDIF                                                             LIN01640
C                                                                       LIN01650
C------- CALCULATE Q AND R VECTORS                                      LIN01660
        DO 230 I=L3,L4                                                  LIN01670
          Q(I)= C4*V(I) +  3*A(I)                                       LIN01680
  230     R(I)=  3*V(I) + C5*A(I)                                       LIN01690
C                                                                       LIN01700
C------- FORM DYNAMIC LOAD                                              LIN01710
C                                                                       LIN01720
C....... INITILISE PBAR= DP*THETA                                       LIN01730
C         THETA=1 FOR LINEAR ACCELERATION                               LIN01740
C         THETA=2 WHEN LINACC IS CALLED BY AVERAGE ACCELERATION         LIN01750
        DO 240 I=L3,L4                                                  LIN01760
  240     PBAR(I)=  DP(I)*THETA                                         LIN01770
C       WRITE (6,239) (DP(I),I=1,14)                                    LIN01780
C 239 FORMAT (1X,1P,14G9.2)                                             LIN01790
C                                                                       LIN01800
      IF (FDAMP.EQ.1) THEN                                              LIN01810
C                                                                       LIN01820
C....... PBAR=PBAR +MASS*( Q(I) + C3*R(I) )                            LIN01830
        DO 250 J=L3,L4                                                  LIN01840
          MDJ=MDMASS(J)+J                                               LIN01850
          M=J-(MDMASS(J+1)-MDMASS(J))+1                                 LIN01860
          M=MAX(M,L3)                                                   LIN01870
          DO 250 I=M,J                                                  LIN01880
            PBAR(I)=PBAR(I)+ MASS(MDJ-I)*( Q(J)+C3*R(J) )               LIN01890
  250       CONTINUE                                                    LIN01900
        DO 255 I=L3+1,L4                                                LIN01910
          MDI=MDMASS(I)+I                                               LIN01920
          M=I-(MDMASS(I+1)-MDMASS(I))+1                                 LIN01930
          M=MAX(M,L3)                                                   LIN01940
          DO 255 J=M,I-1                                                LIN01950
            PBAR(I)=PBAR(I)+ MASS(MDI-J)*( Q(J)+C3*R(J) )               LIN01960
  255       CONTINUE                                                    LIN01970
      ENDIF                                                             LIN01980
C                                                                       LIN01990
      IF (BUG) THEN                                                     LIN02000
        DO 241 I=L3,L4                                                  LIN02010
  241   WRITE (6,242) I,Q(I),R(I),PBAR(I)                               LIN02020
  242   FORMAT (' DOF#',I6,'     Q:',1P,G12.4,'    R:',G12.4,           LIN02030
     &      ' PBAR:',G12.4)                                             LIN02040
      ENDIF                                                             LIN02050
C                                                                       LIN02060
C------- SOLVE FOR INCRIMENTAL DISPLACEMENT OVER TIME INTERVAL DT       LIN02070
C....... NOTE: - WORK IS USED FOR TEMPORARY WORK SPACE                  LIN02080
C              - IF NEWKDT=.FALSE. THE DYNAMIC STIFFNESS HAS ALREADY    LIN02090
C                BEEN REDUCED TO UPPER TRIANGULAR FORM                  LIN02100
C                                                                       LIN02110
      IF (NEWKDT) THEN                                                  LIN02120
        CALL MSOLL(1,BUG,NFREE,IMDS,NFREE,1,1,NFREE,MDS,S,PBAR,WORK,    LIN02130
     &            .FALSE.,DUMMY,IDUMMY,1,                               LIN02140
     &            .FALSE.,DUMMY,IDUMMY,1,ABORT)                         LIN02150
      ELSE                                                              LIN02160
        CALL MSOLL(2,BUG,NFREE,IMDS,NFREE,1,1,NFREE,MDS,S,PBAR,WORK,    LIN02170
     &            .FALSE.,DUMMY,IDUMMY,1,                               LIN02180
     &            .FALSE.,DUMMY,IDUMMY,1,ABORT)                         LIN02190
      ENDIF                                                             LIN02200
      IF (ABORT) RETURN                                                 LIN02210
C                                                                       LIN02220
C....... NOTE: - PBAR CONTAINS THE INCRIMENTAL DISPLACMENT BETWEEN      LIN02230
C              TIMES T AND T+DT AFTER THE SECOND CALL TO MSOLL          LIN02240
      CALL MSOLL(3,BUG,NFREE,IMDS,NFREE,1,1,NFREE,MDS,S,PBAR,WORK,      LIN02250
     &            .FALSE.,DUMMY,IDUMMY,1,                               LIN02260
     &            .FALSE.,DUMMY,IDUMMY,1,ABORT)                         LIN02270
      IF (ABORT) RETURN                                                 LIN02280
C                                                                       LIN02290
C                                                                       LIN02300
C------- DETERMINE ACCELERATION, VELOCITY AND DISPLACEMENT AT TIME DT   LIN02310
C        DA = INCRIMENTAL ACCELERATION BETWEEN TIMES T AND T+DT         LIN02320
C        DV = INCRIMENTAL VELOCTIY BETWEEN TIMES T AND T+DT             LIN02330
C        DD = INCRIMENTAL DISPLACMENT BETWEEN TIMES T AND T+DT          LIN02340
C                                                                       LIN02350
      IF (BUG) THEN                                                     LIN02360
        WRITE (6,*) 'INCRIMENTAL RESULTS FROM LINACC....'               LIN02370
        WRITE (6,320)                                                   LIN02380
      ENDIF                                                             LIN02390
      DO 270 I=L3,L4                                                    LIN02400
        DD(I)=C1*( PBAR(I) + BETA*R(I) )                                LIN02410
        DV(I)=C10*DD(I) - R(I)                                          LIN02420
        DA(I)=C11*DD(I) - Q(I)                                          LIN02430
        IF (BUG) WRITE (6,330) I,DD(I),DV(I),DA(I)                      LIN02440
  270   CONTINUE                                                        LIN02450
C                                                                       LIN02460
      RETURN                                                            LIN02470
C                                                                       LIN02480
C-------------------------------------------------------               LIN02490
C= CALC TOTAL DISPLACEMENTS, VELOCITIES AND ACCELERATIONS ==            LIN02500
C-------------------------------------------------------               LIN02510
```

```
  300 CONTINUE                                                          LIN02520
C     IF (BUG) WRITE (6,320)                                          LIN02530
C     DO 310 I=1,NDOF                                                 LIN02540
C       A(I)=DA(I) + A(I)                                             LIN02550
C       V(I)=DV(I) + V(I)                                             LIN02560
C       D(I)=DD(I) + D(I)                                             LIN02570
C       P(I)=DP(I) + P(I)                                             LIN02580
C       IF (BUG) WRITE (6,330) I,DD(I),DV(I),DA(I),DP(I),             LIN02590
C    &                          D(I), V(I), A(I), P(I)                LIN02600
C 310   DP(I)=0.00                                                    LIN02610
  320   FORMAT ('   DOF',6X,'DD',12X,'DV',12X,'DA',12X,1,'DP',        LIN02620
     &                  12X,' D',12X,' V',12X,' A',12X,' P')          LIN02630
  330   FORMAT (1X,I6,1P,8G14.6)                                      LIN02640
C                                                                     LIN02650
      RETURN                                                          LIN02660
      END                                                             LIN02670
C=======================================================================LMA00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE                               LMA00020
C         END DEBUG                                                   LMA00030
      SUBROUTINE LMATRX(A,MD,IC,IMD,NAME)                             LMA00040
      DIMENSION MD(IC+1), A(IMD)                                      LMA00050
      LOGICAL BTEST                                                   LMA00060
      CHARACTER*(*) NAME                                              LMA00070
      CHARACTER*20  NUMB,ROW(6)                                       LMA00080
      LOGICAL PT,DMP                                                  LMA00090
      IJI(I,J)=MD(J)+J-I                                              LMA00100
C                                                                     LMA00110
      CALL GETINT(NAME,'DUMP=',IDUMP,0,.TRUE.,.FALSE.)                LMA00120
      IF (IDUMP.NE.0)  THEN                                           LMA00130
          DMP=.TRUE.                                                  LMA00140
      ELSE                                                            LMA00150
          DMP=.FALSE.                                                 LMA00160
      ENDIF                                                           LMA00170
          DO 1 I=1,6                                                  LMA00180
    1      ROW(I)=' '                                                 LMA00190
          IR=IC                                                       LMA00200
          WRITE (6,*) ' '                                             LMA00210
          WRITE (6,*) 'PRINTING MATRIX: ',NAME                        LMA00220
          DO 50 ICG=1,IC,6                                            LMA00230
            IF (ICG.GT.1) WRITE (6,*) ' '                             LMA00240
            K=MIN((ICG+5),IC)                                         LMA00250
            WRITE (6,10) (J,J=ICG,K)                                  LMA00260
            DO 20 I=1,IR                                              LMA00270
              L=0                                                     LMA00280
              PT=.FALSE.                                              LMA00290
              DO 15 J=ICG,K                                           LMA00300
                IJ=IJI(I,J)                                           LMA00310
                IF (IJ.GE.MD(J+1) .OR. I.GT.J) THEN                   LMA00320
                  NUMB=' '                                            LMA00330
                ELSE                                                  LMA00340
                  PT=.TRUE.                                           LMA00350
                  WRITE (NUMB,16) A(IJ)                               LMA00360
                  IF (DMP .AND. A(IJ).NE.0)                           LMA00370
     &             WRITE (IDUMP,*) I,J,A(IJ),NAME                     LMA00380
                ENDIF                                                 LMA00390
                L=L+1                                                 LMA00400
   15           ROW(L)=NUMB                                           LMA00410
   20         IF (PT) WRITE (6,30)  I,(ROW(L),L=1,6)                 LMA00420
   50    CONTINUE                                                     LMA00430
         RETURN                                                       LMA00440
   10 FORMAT(T10,6('|',4,' COLUMN',I5,'-------'),'|')                 LMA00450
   16 FORMAT(G20.6)                                                   LMA00460
   30 FORMAT(' ROW',I5,6A)                                            LMA00470
      END                                                             LMA00480
C=======================================================================LMA00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE                               LMA00020
C         END DEBUG                                                   LMA00030
      SUBROUTINE LMAX(A,MD,IC,IMD,NAME)                               LMA00040
      DIMENSION MD(IC+1), A(IMD),B(100,100)                           LMA00050
      LOGICAL BTEST                                                   LMA00060
      CHARACTER*(*) NAME                                              LMA00070
      CHARACTER*20  NUMB,ROW(6)                                       LMA00080
      LOGICAL PT,DMP                                                  LMA00090
C                                                                     LMA00100
      DO 90 I=1,100                                                   LMA00110
      DO 90 J=1,100                                                   LMA00120
   90 B(I,J)=0                                                        LMA00130
                                                                      LMA00140
      DO 50 I=1036,IC                                                 LMA00150
         II=I-1035                                                    LMA00160
      DO 50 J=I,IC                                                    LMA00170
         JJ=J-1035                                                    LMA00180
         IF( J .EQ. I ) THEN                                          LMA00190
           B(II,JJ)=A(MD(I))                                          LMA00200
         ELSE IF( J .NE. I ) THEN                                     LMA00210
           IF( (MD(J)+J-I) .LT. MD(J+1) ) B(II,JJ)=A(MD(J)+J-I)       LMA00220
           IF( (MD(J)+J-I) .GT. MD(J+1) ) B(II,JJ)=0                  LMA00230
         ENDIF                                                        LMA00240
         B(JJ,II)=B(II,JJ)                                            LMA00250
   50 CONTINUE                                                        LMA00260
C                                                                     LMA00270
      WRITE(16,*)' PRINTING MATRIX: ',NAME                            LMA00280
      DO 40 I=1,60                                                    LMA00290
         WRITE (16,30)(B(I,J),J=1,60)                                 LMA00300
   40 CONTINUE                                                        LMA00310
   30 FORMAT(6F20.6)                                                  LMA00320
C                                                                     LMA00330
      RETURN                                                          LMA00340
      END                                                             LMA00350
C=======================================================================MAS00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE                               MAS00020
C         END DEBUG                                                   MAS00030
      SUBROUTINE MASSIN(INMASS,FMASS,MCOND,BUG,IBUG,TITLE,MAXZ,MAXDZ,  MAS00040
     &    MAXNOD,NCOS,NNODE,IS,IW,IZID,Z,NZ,DZ,                       MAS00050
     &    IZMDMS,IZMASS,IZLM,IZKE,IZMDAT,                             MAS00060
     &    NDOF,NMASS,IZCOS,IZIDOF,IZCNST,IZJFLG,IZJCOS,NAME,          MAS00070
     &    NCOND,NFREE,IZMD)                                           MAS00080
C                                                                     MAS00090
      LOGICAL BUG,MCOND                                               MAS00100
      INTEGER FMASS                                                   MAS00110
      CHARACTER*(*) NAME,TITLE(2)                                     MAS00120
      DIMENSION Z(MAXZ),NZ(MAXZ),DZ(MAXDZ),MD(7)                      MAS00130
      LOGICAL BTEST                                                   MAS00140
      DOUBLE PRECISION DZ                                             MAS00150
      DATA MD/1,2,4,7,11,16,22/                                       MAS00160
C                                                                     MAS00170
      IF (FMASS.EQ.1) THEN                                            MAS00180
C-------- INITIALIZE THE DIAGONAL MASS MATRIX                         MAS00190
C       MCOND=.TRUE.                                                  MAS00200
        IZMDMS=IZ                                                     MAS00210
        IZ=IZ + NDOF+1                                                MAS00220
        CALL SKYLIN(0,1,NDOF,IBUG,TITLE,NZ(IZMDMS),                   MAS00230
     &              6,NZ(IZLM),Z(IZKE),MD,NAME)                       MAS00240
C                                                                     MAS00250
C-------- INPUT THE MASS DATA                                         MAS00260
        IZMDAT=IZ                                                     MAS00270
        IZ=IZ + INMASS*12                                             MAS00280
        IMD=NZ(IZMDMS+NDOF)                                           MAS00290
        IZMASS=IZ                                                     MAS00300
       CALL MFORM(1,INMASS,NMASS,Z(IZMASS),NZ(IZMDMS),MD,NDOF,IMD,TITLE,MAS00310
     &             NNODE,MAXNOD,NCOS,BUG,IBUG,                       MAS00320
     &             NZ(IZID),Z(IZCOS),NZ(IZIDOF),Z(IZCNST),NZ(IZJFLG),MAS00330
     &             NZ(IZJCOS),Z(IZMDAT),Z(IZ),NZ(IZ+21),Z(IZ+27),    MAS00340
     &              Z(IZ+ 63),Z(IZ+ 99))                             MAS00350
C                                                                     MAS00360
C-------- RELEASE UNUSED MASS DATA STORAGE                            MAS00370
        IZ=IZ - (INMASS-NMASS)*12                                     MAS00380
C                                                                     MAS00390
C=============================================                        MAS00400
C= MODIFY MASS MATRIX FOR CONDENSATION  ===                           MAS00410
C=============================================                        MAS00420
C THE CONDENSATION PROCESS MAY EXPAND THE MASS MATRIX'S               MAS00430

C SKYLINE.  THE MAXIMUN SKYLINE OF BOTH THE MASS AND                  MAS00440
C THE STIFFNESS MATRIX IS USED.                                       MAS00450
      IF (MCOND .AND. NCOND.GT.0)THEN                                 MAS00460
        DO 310 I=0,NDOF                                               MAS00470
          NHT=NZ(IZMD+I+1)-NZ(IZMD+I)                                 MAS00480
          NSTIFF=I+1+NHT+1                                            MAS00490
  310     NZ(IZMDMS+I)=MIN( NZ(IZMDMS+I),NSTIFF)                      MAS00500
      ENDIF                                                           MAS00510
C                                                                     MAS00520
C                                                                     MAS00530
C-------- DETERMINE THE MASS MATRIX SKYLINE                           MAS00540
        CALL SKYLIN(0,3,NDOF,IBUG,TITLE,NZ(IZMDMS),                   MAS00550
     &                           6,NZ(IZLM),Z(IZKE),MD,NAME)          MAS00560
C                                                                     MAS00570
C-------- GENERATE THE MASS MATRIX                                    MAS00580
        IZMASS=IZ                                                     MAS00590
        IZ=IZ + NZ(IZMDMS+NDOF)                                       MAS00600
        DO 400 I=IZMASS,IZ                                            MAS00610
  400     Z(I)=0.                                                     MAS00620
        IW=IZ                                                         MAS00630
        IZKE=IW                                                       MAS00640
        IZLM=IW+21                                                    MAS00650
       CALL MFORM(2,INMASS,NMASS,Z(IZMASS),NZ(IZMDMS),MD,NDOF,IMD,TITLE,MAS00660
     &             NNODE,MAXNOD,NCOS,BUG,IBUG,                       MAS00670
     &             NZ(IZID),Z(IZCOS),NZ(IZIDOF),Z(IZCNST),NZ(IZJFLG),MAS00680
     &             NZ(IZJCOS),Z(IZMDAT),Z(IW),NZ(IW+21),Z(IW+27),    MAS00690
     &              Z(IW+ 63),Z(IW+ 99))                             MAS00700
C                                                                     MAS00710
        IF (BUG) THEN                                                 MAS00720
          IMD=NZ(IZMDMS+NDOF)                                         MAS00730
          CALL LMATRX(Z(IZMASS),NZ(IZMDMS),NDOF,IMD,NAME)             MAS00740
        ENDIF                                                         MAS00750
      ENDIF                                                           MAS00760
C                                                                     MAS00770
      RETURN                                                          MAS00780
      END                                                             MAS00790
C=======================================================================MAT00010
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    MAT00020
C         END DEBUG                                                   MAT00030
C=======================================================================MAT00040
      SUBROUTINE MATLIB                                               MAT00050
     &  (IOPT  ,LSTTYP ,PRINT ,IREL,HEAD,NAME,EESE,EPSE,DUCT,EXCR,    MAT00060
     &   P     ,PL    ,D     ,DL   ,V     ,VL    ,LELEM ,LMAT  ,     MAT00070
     &   MAT   ,MATTYP,SE,IELNO,DAMAGE)                               MAT00080
      DIMENSION P(12),PL(12),D(12),DL(12),V(12),VL(12)                MAT00090
      LOGICAL KSAME,ERR,FIRST,PRINT,HEAD,BUG2                         MAT00100
      LOGICAL BTEST                                                   MAT00110
      DIMENSION SE(100),DUCT(3),EXCR(6)                               MAT00120
      CHARACTER*80 NAME                                               MAT00130
C                                                                     MAT00140
      INCLUDE (ZCOMN)                                                 MAT00150
C     WRITE (6,*) 'MAT=',MAT,'MATTYP=',MATTYP                         MAT00160
C                                                                     MAT00170
C-----CHECK FOR AN INVALID MATERIAL NUMBER ...                        MAT00180
      IF (MAT.LE.0 .OR. MAT.GT.NMAT) THEN                             MAT00190
        WRITE (6,189) MAT                                             MAT00200
  189   FORMAT (5X,20('*'),' ER','ROR, INVALID MATERIAL PROP.#',I5/     MAT00210
     &   5X,'CORRECT INPUT.........')                                 MAT00220
        CALL ERRTRA                                                   MAT00230
C     WRITE (6,*) 'MAT=',MAT,'MATTYP=',MATTYP                         MAT00240
        RETURN                                                        MAT00250
      ENDIF                                                           MAT00260
C                                                                     MAT00270
      IC     = NZ(IZMAT+MAT   )                                       MAT00280
      IOPT   = IOPT                                                   MAT00290
      IF (IOPT.NE.1) THEN                                             MAT00300
          MATTYP = NZ(IC)                                             MAT00310
      ELSE                                                            MAT00320
          NZ(IC) = MATTYP                                             MAT00330
      ENDIF                                                           MAT00340
      IF (LSTTYP.EQ.MATTYP) THEN                                      MAT00350
          FIRST=.FALSE.                                               MAT00360
      ELSE                                                            MAT00370
          FIRST=.TRUE.                                                MAT00380
          IF (HEAD)  WRITE (6,305) TITLE(1),TITLE(2)                  MAT00390
  305     FORMAT ('1 STRUCTURE.....: ',A,/                            MAT00400
     &            '  SOLUTION.....: ',A,/)                            MAT00410
      ENDIF                                                           MAT00420
C                                                                     MAT00430
      DAMAGE=0                                                        MAT00440
      BUG2=BTEST(IBUG,10)                                             MAT00450
      IF (MATTYP.EQ.1)  THEN                                          MAT00460
        CALL MAT01                                                    MAT00470
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,                   MAT00480
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                     MAT00490
     &              Z(IC+1),EESE,EPSE,DUCT,EXCR)                     MAT00500
      ELSE IF (MATTYP.EQ.2)  THEN                                     MAT00510
        CALL MAT02                                                    MAT00520
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,                   MAT00530
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                     MAT00540
     &              Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)              MAT00550
      ELSE IF (MATTYP.EQ.3)  THEN                                     MAT00560
        CALL MAT03                                                    MAT00570
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,                   MAT00580
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                     MAT00590
     &              Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)              MAT00600
      ELSE IF (MATTYP.EQ.4)  THEN                                     MAT00610
        CALL MAT04                                                    MAT00620
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,                   MAT00630
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                     MAT00640
     &              Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)              MAT00650
      ELSE IF (MATTYP.EQ.6)  THEN                                     MAT00660
        CALL MAT06                                                    MAT00670
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,                   MAT00680
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                     MAT00690
     &              Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)              MAT00700
      ELSE IF (MATTYP.EQ.7)  THEN                                     MAT00710
        CALL MAT07                                                    MAT00720
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,                   MAT00730
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                     MAT00740
     &              Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)              MAT00750
      ELSE IF (MATTYP.EQ.8)  THEN                                     MAT00760
        CALL MAT08                                                    MAT00770
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,                   MAT00780
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                     MAT00790
     &              Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)              MAT00800
      ELSE IF (MATTYP.EQ.9)  THEN                                     MAT00810
        CALL MAT09                                                    MAT00820
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,                   MAT00830
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                     MAT00840
     &              Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)              MAT00850
      ELSE IF (MATTYP.EQ.10)  THEN                                    MAT00860
        CALL MAT10                                                    MAT00870
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,                   MAT00880
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                     MAT00890
     &              Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)              MAT00900
      ELSE IF (MATTYP.EQ.11)  THEN                                    MAT00910
        CALL MAT11                                                    MAT00920
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,                   MAT00930
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                     MAT00940
     &              Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)              MAT00950
      ELSE IF (MATTYP.EQ.12)  THEN                                    MAT00960
        LMAT=Z(IC+29)                                                 MAT00970
        CALL MAT12                                                    MAT00980
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,                   MAT00985
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                     MAT00990
     &              Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)              MAT01000
      ELSE IF (MATTYP.EQ.13)  THEN                                    MAT01010
        LMAT=Z(IC+29)                                                 MAT01020
        CALL MAT13                                                    MAT01030
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,                   MAT01040
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                     MAT01050
```

```
     &           Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)
      ELSE IF (MATTYP.EQ.14)  THEN
        CALL MAT14
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG2,
     &   MAT    ,SE   ,P,PL,D,DL,V,VL,LELEM,LMAT,
     &           Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)
      ELSE
        WRITE (6,10) IELNO
        ERR=.TRUE.
   10   FORMAT(' ELEMENT ',I6,' IS NOT AVAILABLE' )
      ENDIF
      LSTTYP=MATTYP
C
C     WRITE(6,110) (I,NZ(I),Z(I),I=68,91)
  110 FORMAT (0P,' ZMATRX(',I6,'):',I15,1P,G20.10)
      MAT    = MAT
C     WRITE (6,*) 'MAT=',MAT,'MATTYP=',MATTYP
      RETURN
      END
C*******************************************************************************
C          DEBUG UNIT(6),SUBCHK,SUBTRACE
C          END DEBUG
      SUBROUTINE MAT01
     &  (IOPT   ,ISTOP   ,ERR ,FIRST  ,PRINT ,BUG,
     &   MAT    ,SE   ,P,PL,D,DL,V,VL,LELEM,LMAT,
     &   SMAT,EESE,EPSE,DUCT,EXCR)
C
      IMPLICIT REAL(A-H,O-Z)
      LOGICAL ERR,FIRST,PRINT,BUG
      LOGICAL BTEST
      DIMENSION SE(100),SMAT(14),DUCT(3),EXCR(6)
      CHARACTER*80 TYPE
C
      IF (IOPT.EQ.0) THEN
        LELEM=14
        RETURN
      ENDIF
C
C-------------------------------------------------------------------------------
C VARIABLES:
C-------------------------------------------------------------------------------
C---------- GLOBAL VARIABLES
C IOPT   = 1, INITIALIZE MATERIAL
C        = 2, GET STIFFNESS
C SMAT   = INTERNAL STORAGE
C SE     = OUTPUT STIFFNESS
C-------------------------------------------------------------------------------
C
      GO TO (100,200,300,400),IOPT
C
  100 CONTINUE
C
C----- INPUT DATA
C     E      = SMAT( 1)
C     G      = SMAT( 2)
C     AX     = SMAT( 3)
C     AY     = SMAT( 4)
C     AZ     = SMAT( 5)
C     IX     = SMAT( 6)
C     IY     = SMAT( 7)
C     IZ     = SMAT( 8)
C     KIIZ   = SMAT( 9)
C     KJJZ   = SMAT(10)
C     KIJZ   = SMAT(11)
C     KIJY   = SMAT(12)
C     KIIY   = SMAT(13)
C     KIJY   = SMAT(14)
C
      READ (5,*) TYPE ,( SMAT(I),I=1,14)
      READ (5,*) TYPE ,( SMAT(I),I=1, 8)
      LELEM = 12
      LMAT= 14
C
      IF (FIRST .OR. BUG) WRITE (6,191)
C     WRITE (6,192) MAT   ,(SMAT(K),K=1,14)
      WRITE (6,192) MAT   ,(SMAT(K),K=1, 8)
C
  191 FORMAT(/'     3-D ELASTIC BEAM ELEMENT'/
     &       '     ------------------------'//
     &  4X,'MATL.',T14,'E' ,T20,'GAMMA',T33,'AX',T43,'AY',T53,'AZ',
     & T63,'IX',T73,'IY'   ,T83,'IZ'/)
C 191 FORMAT(/'     3-D ELASTIC BEAM ELEMENT'/
C     &       '     ------------------------'//
C     &  4X,'MATL.',T14,'E' ,T20,'GAMMA',T33,'AX',T43,'AY',T53,'AZ',
C     & T63,'IX',T73,'IY'   ,T83,'IZ',T100,'K11',T110,'K22',T120,'K12'/)
  192 FORMAT (4X,I4,1P,8G10.3/)
C 192 FORMAT (4X,I4,1P,8G10.3,' Z-AXIS ',3G10.3/
C     &       4X,4X,1P,80X,   ' Y-AXIS ',3G10.3)
C
C
      RETURN
C*********************** GET STIFFNESS TERMS **********************************
  200 CONTINUE
          E      = SMAT( 1)
          G      = SMAT( 2)
          AX     = SMAT( 3)
          AY     = SMAT( 4)
          AZ     = SMAT( 5)
         RIX     = SMAT( 6)
         RIY     = SMAT( 7)
         RIZ     = SMAT( 8)
         RKIIZ   = SMAT( 9)
         RKJJZ   = SMAT(10)
         RKIJZ   = SMAT(11)
         RKIJY   = SMAT(12)
         RKIIY   = SMAT(13)
         RKIJY   = SMAT(14)
C
      SE(1)=E*AX
      SE(2)=G*AY
      SE(3)=G*AZ
      SE(4)=G*RIX
      SE(5)=E*RIY
      SE(6)=E*RIZ
      SE( 7)=RKIIZ
      SE( 8)=RKJJZ
      SE( 9)=RKIJZ
      SE(10)=RKIIY
      SE(11)=RKJJY
      SE(12)=RKIJY
      SE(13)=EESE
      SE(14)=EPSE
C
  300 SE(13)=0
      SE(14)= (P+PL)/2 * (D-DL)
C
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.
  400 EESE=SE(13)
      EPSE=SE(14)
C
C------ TRANSFER D & E FOR BOTH IOPT=3 AND 4.
      DO 410  I=1,3
      DUCT(I)=0
      EXCR(I)=0
  410 EXCR(I+3)=0
C
      RETURN
      END
C*******************************************************************************
C          DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE
C          END DEBUG
C*******************************************************************************
      SUBROUTINE MAT02
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG,
     &   MAT    ,SE   ,P,PL,D,DL,V,VL,LELEM,LMAT,
     &   SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)
C
      IMPLICIT REAL(A-H,O-Z)
      LOGICAL ERR,FIRST,PRINT,BUG
      LOGICAL BTEST
      DIMENSION SE(100),SMAT(23),DUCT(3),EXCR(6)
      CHARACTER*80 TYPE
C
C--------------- LENGTH OF ELEMENT DATA REQUIRED...
      LELEM=33
      IF (IOPT.EQ.0)   RETURN
C
C-------------------------------------------------------------------------------
C VARIABLES:
C-------------------------------------------------------------------------------
C---------- GLOBAL VARIABLES
C IOPT   = 1, INITIALIZE MATERIAL
C        = 2, GET STIFFNESS
C RINPUT = INPUT DATA
C SMAT   = INTERNAL STORAGE
C SE     = OUTPUT STIFFNESS
C-------------------------------------------------------------------------------
C           ---- AXLMOD DATA ----
C       SMAT(1)= SC,          COMPRESSION STIFFNESS
C       SMAT(2)= ST1,         INITIAL TENSILE STIFFNESS
C       SMAT(3)= ST2,         POST YEILD TENSILE STIFFNESS
C       SMAT(4)= PY,          YEILD POINT
C       SMAT(5)= ALPHA,       AXLMOD PARAMETER
C       SMAT(6)= BETA,        AXLMOD PARAMETER
C       SMAT(7)= DYT,         AXLMOD PARAMETER
C       SMAT(8)= DYC,         AXLMOD PARAMETER
C       SMAT(9)= CSE,         CONSTANT STRAIN ENERGY
C       SMAT(10)=DMAX,        ULTIMATE DISPLACEMENT
C       SMAT(11)=BDAM,        BETA FOR DAMAGE INDEX
C
C------------------ INPUT DATA FOR AXLMOD HYSTERESIS MODEL ------------
C     THE FOLLOWING DATA IS INPUT ONCE FOR EACH MODEL
C
C  SC       = SOMPRESSION STIFFNESS
C  ST1      = INITIAL TENSILE STIFFNESS
C  ST2      = POST YEILD TENSILE STIFFNESS
C  PY       = YEILD LOAD
C  ALPHA    = MODEL COIEFICIENTS
C  BETA     = MODEL COIEFICIENTS
C
C-------------------------------------------------------------------------------
C
      GO TO (100,200,300,400,500),IOPT
C
  100 CONTINUE
C
      READ (5,*) TYPE ,( SMAT(I),I=1,6),DUCMAX,BDAM
      LMAT = 11
      SMAT(7)=  SMAT(4)/SMAT(2)
      SMAT(8)= -SMAT(4)/SMAT(1)
      SMAT(9)=  SMAT(4)**2/(2*SMAT(2))
      SMAT(10)=DUCMAX*SMAT(7)
      SMAT(11)=BDAM
C
      IF (FIRST .OR. BUG) WRITE (6,192)
      WRITE (6,191) MAT,(SMAT(K),K=1,6),DUCMAX,BDAM
C
  191 FORMAT (1X,I5,8G15.6/)
  192 FORMAT (///'     AXLMOD HYSTERESIS MODEL- FOR UNIT LENGTH MEMBER'/
     &        '     ================================================='//
     &'  MAT.',5X,'  SC',10X,'  ST1',10X,'  ST2',10X,'  PY',
     &  10X,'ALPHA',10X,' BETA', 6X,'MAX DUCT.',7X,'BETA-DI')
C
      GO TO 9999
C************************* GET STIFFNESS TERMS ********************************
  200 CONTINUE
C           ---- AXLMOD MODEL DATA STORAGE FOR MEMBER ----
C     SE( 1)= S
C     SE( 2)= HRN
C     SE( 3)= FMAX
C     SE( 4)= DMAX
C     SE( 5)= FX
C     SE( 6)= DX
C     SE( 7)= FP
C     SE( 8)= DP
C     SE( 9)= KR
C     SE(10)= KS
C     SE(11)= KT
C     SE(12)= NRL
C     SE(13)= A
C     SE(14)= S_EQ_UNLOAD = EQUIVALENT UNLOADING STIFFNESS
C     SE(15)= UPOS(1)
C     SE(16)= UPOS(2)
C     SE(17)= UPOS(3)
C     SE(18)= UNEG(1)
C     SE(19)= UNEG(2)
C     SE(20)= UNEG(3)
C     SE(21)= EXCR(1)
C     SE(22)= EXCR(2)
C     SE(23)= EXCR(3)
C     SE(24)= EXCR(4)
C     SE(25)= EXCR(5)
C     SE(26)= EXCR(6)
C     SE(27)= ESE(1)
C     SE(28)= ESE(2)
C     SE(29)= ESE(3)
C     SE(30)= PSE
C     SE(31)= PSE_OLD
C     SE(32)= POS_DMAX
C     SE(33)= POS_PMAX
C
      DO 210 I=1,LELEM
  210   SE(I)=0
      CALL HYST02(P ,D ,PL,DL,V ,VL,SE(1),SE(2),
     &  SMAT(1),SMAT(2),SMAT(3),SMAT(4),SMAT(5),SMAT(6),SMAT(7),SMAT(8),
     &  SMAT(9),
     &  SE(3),SE(4),SE(5),SE(6),SE(7),SE(8),SE(9),SE(10),SE(11),SE(12),
     &  SE(13),BUG,  SE(14),SE(27),SE(30),SE(31),SE(15),SE(18),SE(21),
     &  IDR)
      IF (BUG)  WRITE (6,*) 'STIFF=',STIFF,'   A=',A
      GO TO 9999
C
C************************* GET STIFFNESS TERMS ********************************
  300 CONTINUE
C     SE( 1)= A
C     SE(13)= A
      ICYC=0
      IF (BUG) THEN
        WRITE (6,*) '-------------- IN MAT02 ---- IELNO:',IELNO,
     &              '-----------------------------------'
        WRITE (6,*) 'A:',SE(13)
        WRITE (6,*) '  P,D=',P,D  ,'  V=',V
        WRITE (6,*) 'PL,DL=',PL,DL,' VL=',VL
      ENDIF
      IDR=999
  310 STIFF=SE(1)
      P=PL+(D-DL)*STIFF
      A=SE(13)
      IF (A.EQ.0 .AND. SE(2).EQ.0) A=P
      ICYC=ICYC+1
      IF (BUG) THEN
       WRITE (6,*) 'STIFF=',STIFF,'   A=',A,'   IDR=',IDR
```

```
      WRITE (6,*) '    P=',P    ,'ICYC=',ICYC                                MAT01470
      ENDIF                                                                  MAT01480
C                                                                            MAT01490
C----- IS P WITHIN LIMITS ?                                                  MAT01500
      IF (ICYC.GT.5) THEN                                                    MAT01510
            WRITE (6,*) 'MORE THAN 5 CYCLES IN MAT02.  IELNO=',IELNO         MAT01520
      ELSE IF ((PL.LE.P .AND. P.LT.A) .OR. (PL.GE.P .AND. P.GT.A) .OR.       MAT01530
     &      (DL.EQ.D)) THEN                                                  MAT01540
            CALL STRENG (SE(27),SE(30),SE(31),P,PL,D,DL,SE(1),SE(14) )       MAT01550
            IF (BUG) WRITE (6,*) 'C----- IS P WITH IN LIMITS ? '             MAT01560
            CALL HYST02(P ,D ,PL,DL,V ,VL,SE(1),SE(2),                       MAT01570
     &      SMAT(1),SMAT(2),SMAT(3),SMAT(4),SMAT(5),SMAT(6),                 MAT01580
     &      SMAT(7),SMAT(8),SMAT(9),SE(3),SE(4),SE(5),SE(6),SE(7),SE(8),     MAT01590
     &      SE(9),SE(10),SE(11),SE(12),SE(13),BUG,                           MAT01600
     &      SE(14),SE(27),SE(30),SE(31),SE(15),SE(18),SE(21),IDR )           MAT01610
            V=VL                                                             MAT01620
C----- IS P GREATER THAN LIMITS ?                                            MAT01630
      ELSE IF (((PL.LE.A .AND. A.LE.P) .OR. (PL.GE.A .AND. A.GE.P)).AND.     MAT01640
     &      (STIFF.NE.0.00)) THEN                                            MAT01650
            DI=DL+(A-PL)/STIFF                                               MAT01660
            PI=A                                                             MAT01670
            CALL STRENG (SE(27),SE(30),SE(31),P1,PL,D1,DL,SE(1),SE(14))      MAT01680
            DL=D1                                                            MAT01690
            PL=P1                                                            MAT01700
            IF (ABS(P-PL).GT. 0.005*ABS(P+PL)) THEN                          MAT01710
                  PI=PL+(P-PL)*.01                                           MAT01720
                  DI=DL+(D-DL)*.01                                           MAT01730
            ELSE                                                             MAT01740
                  PI=P                                                       MAT01750
                  DI=D                                                       MAT01760
            ENDIF                                                            MAT01770
            IF (BUG) THEN                                                    MAT01780
              WRITE (6,*) 'C----- IS P GREATER THAN LIMITS ? '               MAT01790
              WRITE (6,*) 'PL,DL=',PL,DL                                     MAT01800
              WRITE (6,*) 'PI,DI=',PI,DI                                     MAT01810
            ENDIF                                                            MAT01820
            CALL HYST02(PI,DI,PL,DL,1.,1.,SE(1),SE(2),                       MAT01830
     &      SMAT(1),SMAT(2),SMAT(3),SMAT(4),SMAT(5),SMAT(6),                 MAT01840
     &      SMAT(7),SMAT(8),SMAT(9),SE(3),SE(4),SE(5),SE(6),SE(7),SE(8),     MAT01850
     &      SE(9),SE(10),SE(11),SE(12),SE(13),BUG,                           MAT01860
     &      SE(14),SE(27),SE(30),SE(31),SE(15),SE(18),SE(21),IDR )           MAT01870
            V=VL                                                             MAT01880
            GO TO 310                                                        MAT01890
C----- IS P LESS THAN LIMITS ?                                               MAT01900
      ELSE IF ((P.LT.PL .AND. PL.LE.A) .OR. (P.GT.PL .AND. PL.GE.A))THEN     MAT01910
            IF (ABS(P-PL).GT. 0.005*ABS(P+PL)) THEN                          MAT01920
                  PI=PL+(P-PL)*.01                                           MAT01930
                  DI=DL+(D-DL)*.01                                           MAT01940
            ELSE                                                             MAT01950
                  PI=P                                                       MAT01960
                  DI=D                                                       MAT01970
            ENDIF                                                            MAT01980
            IF (BUG) THEN                                                    MAT01990
              WRITE (6,*) 'C----- IS P LESS THAN LIMITS ? '                  MAT02000
              WRITE (6,*) 'PI,DI=',PI,DI                                     MAT02010
            ENDIF                                                            MAT02020
            CALL HYST02(PI,DI,PL,DL,1.,1.,SE(1),SE(2),                       MAT02030
     &      SMAT(1),SMAT(2),SMAT(3),SMAT(4),SMAT(5),SMAT(6),                 MAT02040
     &      SMAT(7),SMAT(8),SMAT(9),SE(3),SE(4),SE(5),SE(6),SE(7),SE(8),     MAT02050
     &      SE(9),SE(10),SE(11),SE(12),SE(13),BUG,                           MAT02060
     &      SE(14),SE(27),SE(30),SE(31),SE(15),SE(18),SE(21),IDR )           MAT02070
            V=VL                                                             MAT02080
            GO TO 310                                                        MAT02090
      ENDIF                                                                  MAT02100
C                                                                            MAT02110
C------ SAVE MAXIMUM PEAK POSITIVE DISPL                                     MAT02120
      IF (D.GT.SE(32)) THEN                                                  MAT02130
            SE(32)=D                                                         MAT02140
            SE(33)=P                                                         MAT02150
      ENDIF                                                                  MAT02160
C                                                                            MAT02170
      IF (BUG)  WRITE (6,*) 'STIFF=',STIFF,'   A=',SE(13)                    MAT02180
C                                                                            MAT02190
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                             MAT02200
  400 EESE=SE(27)                                                            MAT02210
      EPSE=SE(30)                                                            MAT02220
C                                                                            MAT02230
C------ TRANSFER DUCTILITIES AND EXCRUSION RATIOS FOR BOTH IOPT=3 AND 4.     MAT02240
      DO 410 I=1,3                                                           MAT02250
      DUCT(I)=MAX(SE(14+I),SE(17+I))                                         MAT02260
      EXCR(I)=SE(20+I)                                                       MAT02270
  410 EXCR(I+3)=SE(20+I+3)                                                   MAT02280
C                                                                            MAT02290
      RETURN                                                                 MAT02300
C                                                                            MAT02310
C---------------------- DAMAGE INDEX ------------------------------------    MAT02320
  500 CONTINUE                                                               MAT02330
      EESE=SE(27)                                                            MAT02340
      EPSE=SE(30)                                                            MAT02350
      BDAM=SMAT(11)                                                          MAT02360
      D =SE(32)                                                              MAT02370
      P =SE(33)                                                              MAT02380
      DAMAGE=ABS(D)/SMAT(10) + BDAM /( SMAT(4)*SMAT(10)) * EPSE              MAT02390
C                                                                            MAT02400
 9999 CONTINUE                                                               MAT02410
C                                                                            MAT02420
      RETURN                                                                 MAT02430
      END                                                                    MAT02440
C--------------------------------------------------------------------------- MAT00010
C         DEBUG UNIT(6),SUBCHK,INIT,SUBTRACE                                 MAT00020
C         END DEBUG                                                          MAT00030
C--------------------------------------------------------------------------- MAT00040
      SUBROUTINE MAT03                                                       MAT00050
     &  (IOPT ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG,                             MAT00060
     &   MAT  ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                               MAT00070
     &   SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)                                    MAT00080
C                                                                            MAT00090
      IMPLICIT REAL(A-H,O-Z)                                                 MAT00100
      LOGICAL ERR,FIRST,PRINT,BUG                                            MAT00110
      LOGICAL BTEST                                                          MAT00120
      DIMENSION SE(100),SMAT(30),DUCT(3),EXCR(6)                             MAT00130
      CHARACTER*80 TYPE                                                      MAT00140
C                                                                            MAT00150
      IF (IOPT.NE.1) LELEM=34+3*SMAT(2)+2                                    MAT00160
      IF (IOPT.EQ.0)  RETURN                                                 MAT00170
C                                                                            MAT00180
C--------------------------------------------------------------------------- MAT00190
C VARIABLES:                                                                 MAT00200
C--------------------------------------------------------------------------- MAT00210
C---------- GLOBAL VARIABLES                                                 MAT00220
C IOPT   = 1, INITIALIZE MATERIAL                                            MAT00230
C        = 2, GET STIFFNESS                                                  MAT00240
C SMAT   = INTERNAL STORAGE                                                  MAT00250
C SE     = OUTPUT STIFFNESS                                                  MAT00260
C--------------------------------------------------------------------------- MAT00270
C     ---- BEND1 DATA ----                                                   MAT00280
C        SMAT(1)= NSEGS,          NUMBER OF BACKBONE SEGMENTS                MAT00290
C        SMAT(2)= NI,             NUMBER OF SMALL AMPLITUDE LOOPS            MAT00300
C        SMAT(3)= CSE,            CONSTANT STRAIN ENERGY...                  MAT00310
C        SMAT(4)= DY,             YEILD POINT FOR DUCTILITIES                MAT00320
C        SMAT(5          )= BDAM  BETA FOR DAMAGE INDEX                      MAT00330
C        SMAT(5+I        )= PP,   BACKBONE MOMENT FOR POINT I                MAT00340
C        SMAT(5+I+NSEGS)= DP,     BACKBONE ROTATION FOR POINT I               MAT00350
C                                                                            MAT00360
      GO TO (100,200,300,400,500),IOPT                                       MAT00370
C                                                                            MAT00380
C                                                                            MAT00390
  100 CONTINUE                                                               MAT00400
C------------------ INPUT DATA FOR BEND1  HYSTERESIS MODEL ----------------- CMAT00410
C                                                                            CMAT00420
C  NSEG    = NUMBER OF BACKBONE SEGEMENTS                                    CMAT00430
C  NI      = NUMBER OF SMALL LOOP POINTS                                     CMAT00440
C  EI      = ELASTIC STIFFNESS                                               CMAT00450
C  G       = MOMENT GRADINET                                                 CMAT00460
C  PP      = BACKBONE LOAD                                                   CMAT00470
C  DP      = BACKBONE DISPLACEMENT                                           CMAT00480
C                                                                            CMAT00490
C------------------------------------------------------------------------------CMAT00500
      READ (5,*) TYPE ,NSEG,NI,DY,BDAM                                       MAT00510
      READ(5,*) ( SMAT(5+I      ),I=1,NSEG)                                  MAT00520
      READ(5,*) ( SMAT(5+I+NSEG),I=1,NSEG)                                   MAT00530
C                                                                            MAT00540
      LELEM = 34+3*NI +2                                                     MAT00550
      LMAT = 5+2*NSEG                                                        MAT00560
      SMAT(1)=NSEG                                                           MAT00570
      SMAT(2)=NI                                                             MAT00580
CCC   SMAT(3)=SMAT(6)/SMAT(6+NSEG)                                           MAT00590
      SMAT(4)=DY                                                             MAT00600
      SMAT(5)=BDAM                                                           MAT00610
C------------- CALC CONSTANT STRAIN ENERGY....                               MAT00620
      DL=0                                                                   MAT00630
      PL=0                                                                   MAT00640
      DO 10 I=1,NSEG                                                         MAT00650
            P=SMAT(5+I)                                                      MAT00660
            D=SMAT(5+I+NSEG)                                                 MAT00670
            IF (D.LE.DY) THEN                                                MAT00680
              CSE=CSE + 0.5*(P+PL)*(D-DL)                                    MAT00690
              PL=P                                                           MAT00700
              DL=D                                                           MAT00710
            ELSE IF (D.EQ.DL) THEN                                           MAT00720
              GO TO 20                                                       MAT00730
            ELSE                                                             MAT00740
              PY=PL+ (DY-DL)*(P-PL)/(D-DL)                                   MAT00750
              CSE=CSE + 0.5*(PY+PL)*(DY-DL)                                  MAT00760
              GO TO 20                                                       MAT00770
            ENDIF                                                            MAT00780
   10 CONTINUE                                                               MAT00790
   20 SMAT(3)=CSE                                                            MAT00800
C                                                                            MAT00810
      IF (FIRST .OR. BUG) WRITE (6,55)                                       MAT00820
      WRITE (6,57) MAT,NI,DY,BDAM,(SMAT(5+K),SMAT(5+K+NSEG),K=1,NSEG)        MAT00830
      WRITE (6,*) ' '                                                        MAT00840
C                                                                            MAT00850
   55   FORMAT('/'     BEND1 HYSTERESIS MODEL DATA - UNIT LENGTH MEMBER'     MAT00860
     &  /'                                                                   MAT00870
     &       /'     BACKBONE CURVE POINTS - MAT#  NI ',5X,'DY',9X,          MAT00880
     &  '  BETA          ',3X,' MOMENT',8X,'ROTATION')                      MAT00890
   57   FORMAT     ( (20X,2I4.2X,G15.6,3G15.6)/(68X,2G15.6))                MAT00900
C                                                                            MAT00910
      RETURN                                                                 MAT00920
C------------------------ GET STIFFNESS TERMS ------------------------------MAT00930
  200 CONTINUE                                                               MAT00940
C           ---- BEND1 MODEL DATA STORAGE FOR MEMBER ----                    MAT00950
C            SE( 1)= K                                                       MAT00960
C            SE( 2)= RULE                                                    MAT00970
C            SE( 3)= DIR                                                     MAT00980
C            SE( 4)= A                                                       MAT00990
C            SE( 5)= PMG                                                     MAT01000
C            SE( 6)= PMH                                                     MAT01010
C            SE( 7)= DMG                                                     MAT01020
C            SE( 8)= DMH                                                     MAT01030
C            SE( 9)= BACKB                                                   MAT01040
C            SE(10)= ALPHA1                                                  MAT01050
C            SE(11)= ALPHA2                                                  MAT01060
C            SE(12)= IR                                                      MAT01070
C            SE(13)= SR1                                                     MAT01080
C            SE(14)= SR2                                                     MAT01090
C            SE(15)= SR3                                                     MAT01100
C            SE(16)= SRM                                                     MAT01110
C            SE(17)= E_EQ_UNLOAD                                             MAT01120
C            SE(18)= UPOS(1)                                                 MAT01130
C            SE(19)= UPOS(2)                                                 MAT01140
C            SE(20)= UPOS(3)                                                 MAT01150
C            SE(21)= UNEG(1)                                                 MAT01160
C            SE(22)= UNEG(2)                                                 MAT01170
C            SE(23)= UNEG(3)                                                 MAT01180
C            SE(24)= EXCR(1)                                                 MAT01190
C            SE(25)= EXCR(2)                                                 MAT01200
C            SE(26)= EXCR(3)                                                 MAT01210
C            SE(27)= EXCR(4)                                                 MAT01220
C            SE(28)= EXCR(5)                                                 MAT01230
C            SE(29)= EXCR(6)                                                 MAT01240
C            SE(30)= EESE(1)                                                 MAT01250
C            SE(31)= EESE(2)                                                 MAT01260
C            SE(32)= EESE(3)                                                 MAT01270
C            SE(33)= EPSE                                                    MAT01280
C            SE(34)= PSE_OLD                                                 MAT01290
C            SE(34+1     )=PR( 1)                                            MAT01300
C            SE(34+    NI)=PR(NI)                                            MAT01310
C            SE(34+1+  NI)=DR( 1)                                            MAT01320
C            SE(34+  2*NI)=DR(NI)                                            MAT01330
C            SE(34+1+2*NI)=FR( 1)                                            MAT01340
C            SE(34+  3*NI)=FR(NI)                                            MAT01350
C            SE(35+  3*NI)=P_MAX                                             MAT01360
C            SE(36+  3*NI)=D_MAX                                             MAT01370
C                                                                            MAT01380
      NSEG=SMAT(1)                                                           MAT01390
      NI  =SMAT(2)                                                           MAT01400
      J1=  NSEG+6                                                            MAT01410
      J3=  NI+35                                                             MAT01420
      J4=2*NI+35                                                             MAT01430
      J5=3*NI+35-1 +2                                                        MAT01440
      SI=SMAT(6)/SMAT(6+NSEG)                                                MAT01450
C*    EI=  SMAT(4)                                                           MAT01460
      DO 210 I=1,J5                                                          MAT01470
  210 SE(I)=0                                                                MAT01480
      CALL HYST03(P,D,PL,DL,V ,NSEG,NI,                                      MAT01490
     & SMAT( 6),SMAT(J1),SMAT( 6),SMAT(J1),SI,SMAT(3),                       MAT01500
     & SE( 1),SE( 2),SE( 3),SE( 4),SE( 5),SE( 6),SE( 7),SE( 8),SE( 9),      MAT01510
     & SE(10),SE(11),SE(12),SE(13),SE(14),SE(15),SE(16),SE(35),SE(J3),       MAT01520
     & SE(J4),BUG,                                                           MAT01530
     & SE(17),SMAT(4),SE(30) ,SE(33),SE(34),SE(18),SE(21),SE(24))            MAT01540
      RETURN                                                                 MAT01550
C                                                                            MAT01560
C------------------------ GET STIFFNESS TERMS ------------------------------MAT01570
  300 CONTINUE                                                               MAT01580
C           SE( 4)= A                                                        MAT01590
      NSEG=SMAT(1)                                                           MAT01600
      NI  =SMAT(2)                                                           MAT01610
      J1=  NSEG+6                                                            MAT01620
      J3=  NI+35                                                             MAT01630
      J4=2*NI+35                                                             MAT01640
      SI=SMAT(6)/SMAT(6+NSEG)                                                MAT01650
C*    EI=  SMAT(4)                                                           MAT01660
C     SET VELOCITY FLAG TO 1 TO REMOVE INFL OF HIGH FREQ VELOC               MAT01670
C     DVEL=V*VL                                                              MAT01680
      DVEL = 1                                                               MAT01690
      ICYC=0                                                                 MAT01700
      IF (BUG) THEN                                                          MAT01710
        WRITE (6,*) '------------- IN MAT03 -------------------------'        MAT01720
        WRITE (6,*) 'P,D',P,D                                                MAT01730
        WRITE (6,*) 'PL,DL=',PL,DL                                           MAT01740
      ENDIF                                                                  MAT01750
  310 STIFF=SE(1)                                                            MAT01760
      P=PL+(D-DL)*STIFF                                                      MAT01770
      A=SE(4)                                                                MAT01780
      IF (A.EQ.0 .AND. SE(2).EQ.0) THEN                                      MAT01790
            A=SIGN(SMAT(6),P)                                                MAT01800
            IF (P.LT.PL .AND. PL.GT.0 .AND. P.GT.0) A=0                      MAT01810
            IF (P.GT.PL .AND. PL.LT.0 .AND. P.LT.0) A=0                      MAT01820
      ENDIF                                                                  MAT01830
      ICYC=ICYC+1                                                            MAT01840
      IF (BUG) THEN                                                          MAT01850
       WRITE (6,*) 'STIFF=',STIFF,'   A=',A                                  MAT01860
```

```
      WRITE (6,*) '     P=',P     ,'ICYC=',ICYC                                MAT01870
      ENDIF                                                                    MAT01880
C                                                                              MAT01890
C----- IS P WITHIN LIMITS ?                                                    MAT01900
      IF (ICYC.GT.10) THEN                                                     MAT01910
        WRITE (6,*) 'MORE THAN 10 CYCLES IN MAT03,  IELNO=',IELNO              MAT01920
      ELSE IF (((PL.LE.P .AND. P.LT.A) .OR. (PL.GE.P .AND. P.GT.A) .OR.        MAT01930
     &       (DL.EQ.D)) THEN                                                   MAT01940
        IF (BUG) WRITE (6,*) 'C----- IS P WITH IN LIMITS ? '                   MAT01950
        CALL STRENG (SE(30),SE(33),SE(34),P,PL,D,DL,SE(1),SE(17) )             MAT01960
        CALL HIST03(P,D,PL,DL,DVEL ,NSEG,NI,                                  MAT01970
     &       SMAT( 6),SMAT(J1),SMAT( 6),SMAT(J1),SI,SMAT(3),                   MAT01980
     &       SE( 1),SE( 2),SE( 3),SE( 4),SE( 5),SE( 6),                        MAT01990
     &       SE( 7),SE( 8),SE( 9),                                             MAT02000
     &       SE(10),SE(11),SE(12),SE(13),SE(14),SE(15),                        MAT02010
     &       SE(16),SE(35),SE(J3),                                             MAT02020
     &       SE(J4),BUG,                                                       MAT02030
     &       SE(17),SMAT(4),SE(30) ,SE(33),SE(34),SE(18),SE(21),SE(24))        MAT02040
        DVEL=1                                                                 MAT02050
C----- IS P GREATER THAN LIMITS ?                                              MAT02060
      ELSE IF (((PL.LE.A .AND. A.LE.P) .OR. (PL.GE.A .AND. A.GE.P)).AND.       MAT02070
     &     (STIFF.NE.0.00)) THEN                                               MAT02080
        DI=DL+(A-PL)/STIFF                                                     MAT02090
        PI=A                                                                   MAT02100
        CALL STRENG (SE(30),SE(33),SE(34),P1,PL,D1,DL,SE(1),SE(17) )           MAT02110
        DL=D1                                                                  MAT02120
        PL=P1                                                                  MAT02130
C       PI=PL+(P-PL)*.001                                                      MAT02140
C       DI=DL+(D-DL)*.001                                                      MAT02150
        IF (ABS(P-PL).GT. 0.0005*ABS(P+PL)) THEN                               MAT02160
             PI=PL+(P-PL)*.001                                                 MAT02170
             DI=DL+(D-DL)*.001                                                 MAT02180
        ELSE                                                                   MAT02190
             PI=P                                                              MAT02200
             DI=D                                                              MAT02210
        ENDIF                                                                  MAT02220
        IF (BUG) THEN                                                          MAT02230
          WRITE (6,*) 'C----- IS P GREATER THAN LIMITS ? '                     MAT02240
          WRITE (6,*) 'PL,DL=',PL,DL                                           MAT02250
          WRITE (6,*) 'PI,DI=',PI,DI                                           MAT02260
        ENDIF                                                                  MAT02270
        CALL HIST03(PI,DI,PL,DL,1.00 ,NSEG,NI,                                MAT02280
     &       SMAT( 6),SMAT(J1),SMAT( 6),SMAT(J1),SI,SMAT(3),                   MAT02290
     &       SE( 1),SE( 2),SE( 3),SE( 4),SE( 5),SE( 6),                        MAT02300
     &       SE( 7),SE( 8),SE( 9),                                             MAT02310
     &       SE(10),SE(11),SE(12),SE(13),SE(14),SE(15),                        MAT02320
     &       SE(16),SE(35),SE(J3),                                             MAT02330
     &       SE(J4),BUG,                                                       MAT02340
     &       SE(17),SMAT(4),SE(30) ,SE(33),SE(34),SE(18),SE(21),SE(24))        MAT02350
        GO TO 310                                                              MAT02360
C----- IS P LESS THAN LIMITS ?                                                 MAT02370
      ELSE IF ((P.LT.PL .AND. PL.LE.A) .OR. (P.GT.PL .AND. PL.GE.A))THEN       MAT02380
C       PI=PL+(P-PL)*.001                                                      MAT02390
C       DI=DL+(D-DL)*.001                                                      MAT02400
        IF (ABS(P-PL).GT. 0.0005*ABS(P+PL)) THEN                               MAT02410
             PI=PL+(P-PL)*.001                                                 MAT02420
             DI=DL+(D-DL)*.001                                                 MAT02430
        ELSE                                                                   MAT02440
             PI=P                                                              MAT02450
             DI=D                                                              MAT02460
        ENDIF                                                                  MAT02470
        IF (BUG) THEN                                                          MAT02480
          WRITE (6,*) 'C----- IS P LESS THAN LIMITS ? '                        MAT02490
          WRITE (6,*) 'PI,DI=',PI,DI                                           MAT02500
        ENDIF                                                                  MAT02510
        CALL HIST03(PI,DI,PL,DL,1.00 ,NSEG,NI,                                MAT02520
     &       SMAT( 6),SMAT(J1),SMAT( 6),SMAT(J1),SI,SMAT(3),                   MAT02530
     &       SE( 1),SE( 2),SE( 3),SE( 4),SE( 5),SE( 6),                        MAT02540
     &       SE( 7),SE( 8),SE( 9),                                             MAT02550
     &       SE(10),SE(11),SE(12),SE(13),SE(14),SE(15),                        MAT02560
     &       SE(16),SE(35),SE(J3),                                             MAT02570
     &       SE(J4),BUG,                                                       MAT02580
     &       SE(17),SMAT(4),SE(30) ,SE(33),SE(34),SE(18),SE(21),SE(24))        MAT02590
        GO TO 310                                                              MAT02600
      ENDIF                                                                    MAT02610
C                                                                              MAT02620
C-------- SAVE MAXIMUM DISPL AND IT'S LOAD                                     MAT02630
      IF (ABS(D).GT.ABS(SE(36+3*NI))) THEN                                     MAT02640
            SE(35+  3*NI)=P                                                    MAT02650
            SE(36+  3*NI)=D                                                    MAT02660
      ENDIF                                                                    MAT02670
C                                                                              MAT02680
      IF (BUG)  WRITE (6,*) 'STIFF=',SE(1),'   A=',SE(4)                       MAT02690
C                                                                              MAT02700
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                               MAT02710
  400 ESSE=SE(30)                                                              MAT02720
      SPSE=SE(33)                                                              MAT02730
C                                                                              MAT02740
C------ TRANSFER DUCTILITIES AND EXCRUSION RATIOS FOR BOTH IOPT=3 AND 4.       MAT02750
      DO 410 I=1,3                                                             MAT02760
      DUCT(I)=MAX(SE(17+I),SE(20+I))                                           MAT02770
      EXCR(I+3)=SE(23+I+3)                                                     MAT02780
  410 EXCR(I)=SE(23+I)                                                         MAT02790
C                                                                              MAT02800
      RETURN                                                                   MAT02810
C                                                                              MAT02820
C--------------------------- DAMAGE INDEX ------------------------------       MAT02830
  500 ESSE =SE(30)                                                             MAT02840
      SPSE =SE(33)                                                             MAT02850
      NSEGS=SMAT(1)                                                            MAT02860
      NI   =SMAT(2)                                                            MAT02870
      DMAX =SMAT(5+2*NSEGS)                                                    MAT02880
      DY   =SMAT(4)                                                            MAT02890
      BDAM =SMAT(5)                                                            MAT02900
      P    =SE(35+  3*NI)                                                      MAT02910
      D    =SE(36+  3*NI)                                                      MAT02920
                                                                               MAT02930
      D1   =SMAT(6+NSEGS)                                                      MAT02940
      IF (DY.LE.D1) THEN                                                       MAT02950
           PY=SMAT(6)*DY/D1                                                    MAT02960
      ELSE                                                                     MAT02970
           DO 510 I=2,NSEGS                                                    MAT02980
             P1=SMAT(4+I)                                                      MAT02990
             P2=SMAT(5+I)                                                      MAT03000
             D1=SMAT(4+I+NSEGS)                                                MAT03010
             D2=SMAT(5+I+NSEGS)                                                MAT03020
             IF (DY.GE.D1 .AND. DY.LE.D2)  THEN                                MAT03030
               PY=P1+(DY-D1)*(P2-P1)/(D2-D1)                                   MAT03040
               GO TO 520                                                       MAT03050
             ENDIF                                                             MAT03060
  510        CONTINUE                                                          MAT03070
      ENDIF                                                                    MAT03080
  520 DAMAGE=ABS(D)/DMAX + BDAM / (PY*DMAX) * EPSE                             MAT03090
C                                                                              MAT03100
      RETURN                                                                   MAT03110
      END                                                                      MAT03120
C-------------------------------------------------------------------------     MAT00010
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                             MAT00020
C         END DEBUG                                                            MAT00030
C-------------------------------------------------------------------------     MAT00040
      SUBROUTINE MAT04                                                         MAT00050
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG,                             MAT00060
     &   MAT    ,SE   ,P,PL,D,DL,V,VL,LELEM,LMAT,                             MAT00070
     &   SMAT,ESSE,EPSE,DUCT,EXCR,DAMAGE)                                      MAT00080
C                                                                              MAT00090
      IMPLICIT REAL(A-H,O-Z)                                                   MAT00100
      LOGICAL ERR,FIRST,PRINT,BUG                                              MAT00110
      LOGICAL BTEST                                                            MAT00120
      DIMENSION SE(100),SMAT(20),DUCT(3),EXCR(6)                               MAT00130
      CHARACTER*80 TYPE                                                        MAT00140
C                                                                              MAT00150
      IF (IOPT.NE.1) LELEM=32+3*SMAT(2)  +2                                    MAT00160
      IF (IOPT.EQ.0) RETURN                                                    MAT00170
C                                                                              MAT00180
C-------------------------------------------------------------------------     MAT00190
C VARIABLES:                                                                   MAT00200
C-------------------------------------------------------------------------     MAT00210
C---------- GLOBAL VARIABLES                                                    MAT00220
C IOPT   = 1, INITIALIZE MATERIAL                                              MAT00230
C        = 2, GET STIFFNESS                                                    MAT00240
C RINPUT = INPUT DATA                                                          MAT00250
C SMAT   = INTERNAL STORAGE                                                    MAT00260
C SE     = OUTPUT STIFFNESS                                                    MAT00270
C-------------------------------------------------------------------------     MAT00280
C           ---- SHEAR1 DATA ----                                              MAT00290
C             SMAT(1)= NSEGS.          NUMBER OF BACKBONE SEGMENTS             MAT00300
C             SMAT(2)= NI,             NUMBER OF SMALL AMPLITUDE LOOPS         MAT00310
C             SMAT(3)= CSE,            CONSTAINT STRAIN ENERGY...              MAT00320
C             SMAT(4)= DY,             YEILD DISPLACEMENT                      MAT00330
C             SMAT(4+I)= PP,           BACKBONE SHEAR FOR POINT I              MAT00340
C             SMAT(4+I+NSEG)= DP,      BACKBONE DISPLACEMENT FOR POINT I       MAT00350
C             SMAT(5+2*NSEG)= BDAM     BETA FOR DAMAGE INDEX                   MAT00360
C                                                                              MAT00370
      GO TO (100,200,300,400,500),IOPT                                         MAT00380
C                                                                              MAT00390
  100 CONTINUE                                                                 MAT00400
C                                                                              MAT00410
C------------------ INPUT DATA FOR SHEAR1 HYSTERESIS MODEL ------------       CMAT00420
C                                                                              CMAT00430
C  NSEG    = NUMBER OF BACKBONE SEGEMENTS                                      CMAT00440
C  NI      = NUMBER OF SMALL LOOP POINTS                                       CMAT00450
C  PP      = BACKBONE LOAD                                                     CMAT00460
C  DP      = BACKBONE DISPLACEMENT                                             CMAT00470
C-------------------------------------------------------------------------     CMAT00480
      READ (5,*) TYPE ,NSEG,NI,DY,BDAM                                         MAT00490
      READ (5,*) ( SMAT(4+I     ),I=1,NSEG)                                    MAT00500
      READ (5,*) ( SMAT(4+I+NSEG),I=1,NSEG)                                    MAT00510
      LELEM = 32+3*NI +2                                                       MAT00520
      LMAT = 5+2*NSEG                                                          MAT00530
      SMAT(1)=NSEG                                                             MAT00540
      SMAT(2)=NI                                                               MAT00550
      SMAT(4)=DY                                                               MAT00560
      SMAT(5+2*NSEG)=BDAM                                                      MAT00570
C------------- CALC CONSTANT STRAIN ENERGY....                                 MAT00580
      DL=0                                                                     MAT00590
      PL=0                                                                     MAT00600
      DO 10 I=1,NSEG                                                           MAT00610
            P=SMAT(4+I)                                                        MAT00620
            D=SMAT(4+I+NSEG)                                                   MAT00630
            IF (D.LE.DY) THEN                                                  MAT00640
              CSE=CSE + 0.5*(P+PL)*(D-DL)                                      MAT00650
              PL=P                                                             MAT00660
              DL=D                                                             MAT00670
            ELSE IF (D.EQ.DL) THEN                                             MAT00680
              GO TO 20                                                         MAT00690
            ELSE                                                               MAT00700
              PY=PL+ (DY-DL)*(P-PL)/(D-DL)                                     MAT00710
              CSE=CSE + 0.5*(PY+PL)*(DY-DL)                                    MAT00720
              GO TO 20                                                         MAT00730
            ENDIF                                                              MAT00740
   10 CONTINUE                                                                 MAT00750
   20 SMAT(3)=CSE                                                              MAT00760
C                                                                              MAT00770
C                                                                              MAT00780
      IF (FIRST .OR. BUG) WRITE (6,65)                                         MAT00790
      WRITE (6,67) MAT,NI,DY,BDAM,(SMAT(4+K),SMAT(4+K+NSEG),K=1,NSEG)          MAT00800
      WRITE (6,*)  '                                                           MAT00810
   65 FORMAT(///'     SHEAR1 HYSTERESIS MODEL DATA - UNIT LENGTH MEMBER'       MAT00820
     &  /'          ==========================================================',MAT00830
     &  /'               BACKBONE CURVE POINTS - MAT.  NI         DY      ',  MAT00840
     &  /  5X,'BETA',6X,    3X,' STRESS',8X,' STRAIN')                        MAT00850
   67 FORMAT      ( (20X,2I4,2X,4G15.6)/(68X,2G15.6))                          MAT00860
C                                                                              MAT00870
      RETURN                                                                   MAT00880
C------------------------- GET STIFFNESS TERMS ------------------------       MAT00890
  200 CONTINUE                                                                 MAT00900
C           ---- SHEAR1 MODEL DATA STORAGE FOR MEMBER ----                     MAT00910
C       SE( 1)= K                                                              MAT00920
C       SE( 2)= RULE                                                           MAT00930
C       SE( 3)= DIR                                                            MAT00940
C       SE( 4)= A                                                              MAT00950
C       SE( 5)= PMG                                                            MAT00960
C       SE( 6)= PMH                                                            MAT00970
C       SE( 7)= DMG                                                            MAT00980
C       SE( 8)= DMH                                                            MAT00990
C       SE( 9)= BACKB                                                          MAT01000
C       SE(10)= SR1                                                            MAT01010
C       SE(11)= SR2                                                            MAT01020
C       SE(12)= SR3                                                            MAT01030
C       SE(13)= SRM                                                            MAT01040
C       SE(14)= IR                                                             MAT01050
C       SE(15)= S_EQ_UNLOAD                                                    MAT01060
C       SE(16)= UPOS(1)                                                        MAT01070
C       SE(17)= UPOS(2)                                                        MAT01080
C       SE(18)= UPOS(3)                                                        MAT01090
C       SE(19)= UNEG(1)                                                        MAT01100
C       SE(20)= UNEG(2)                                                        MAT01110
C       SE(21)= UNEG(3)                                                        MAT01120
C       SE(22)= EXCR(1)                                                        MAT01130
C       SE(23)= EXCR(2)                                                        MAT01140
C       SE(24)= EXCR(3)                                                        MAT01150
C       SE(25)= EXCR(4)                                                        MAT01160
C       SE(26)= EXCR(5)                                                        MAT01170
C       SE(27)= EXCR(6)                                                        MAT01180
C       SE(28)= ESE                                                            MAT01190
C       SE(29)= ESE                                                            MAT01200
C       SE(30)= ESE                                                            MAT01210
C       SE(31)= PSE                                                            MAT01220
C       SE(32)= PSE_OLD                                                        MAT01230
C       SE(32+1    )=PR( 1)                                                    MAT01240
C       SE(32+   NI)=PR(NI)                                                    MAT01250
C       SE(32+1+  NI)=DR( 1)                                                   MAT01260
C       SE(32+  2*NI)=DR(NI)                                                   MAT01270
C       SE(32+1+2*NI)=FR( 1)                                                   MAT01280
C       SE(32+  3*NI)=FR(NI)                                                   MAT01290
C       SE(33+  3*NI)=P_MAX                                                    MAT01300
C       SE(34+  3*NI)=D_MAX                                                    MAT01310
C                                                                              MAT01320
      NSEG=SMAT(1)                                                             MAT01330
      NI   =SMAT(2)                                                            MAT01340
      SI   = SMAT(5)/SMAT(5+NSEG)                                              MAT01350
      J1=NSEG+5                                                                MAT01360
      J3=NI+33                                                                 MAT01370
      J4=2*NI+33                                                               MAT01380
      J5=3*NI+33-1 +2                                                          MAT01390
      DO 210 I=1,J5                                                            MAT01400
  210 SE(I)=0                                                                  MAT01410
          CALL HIST04( P, D,PL,DL, V  ,NSEG,NI,BUG,                            MAT01420
     &          SMAT( 5),SMAT(J1),SMAT( 5),SMAT(J1),SI,SMAT(3),SMAT(4),      MAT01430
     &          SE( 1)  ,SE( 2)  ,SE( 3)  ,SE( 4)  ,SE( 5) ,SE( 6) ,        MAT01440
     &          SE( 7)  ,SE( 8)  ,SE( 9)  ,SE(10)  ,SE(11) ,SE(12) ,        MAT01450
     &          SE(13)  ,SE(14)  ,SE(15)  ,SE(16)  ,SE(19) ,SE(22) ,        MAT01460
     &          SE(28)  ,SE(31)  ,SE(32)  ,SE(33)  ,SE(J3) ,SE(J4) )        MAT01470
      RETURN                                                                   MAT01480
C                                                                              MAT01490
C------------------------- GET STIFFNESS TERMS ------------------------       MAT01500
  300 CONTINUE                                                                 MAT01510
C           SE( 4)= A                                                          MAT01520
      NSEG=SMAT(1)                                                             MAT01530
      NI   =SMAT(2)                                                            MAT01540
      SI   = SMAT(5)/SMAT(5+NSEG)                                              MAT01550
      J1=NSEG+5                                                                MAT01560
      J3=NI+33                                                                 MAT01570
      J4=2*NI+33                                                               MAT01580
```

```
      DVEL=V*VL                                                         MAT01590
C     SET VELOCITY FLAG TO 1 TO REMOVE INFL OF HIGH FREQ VELOC          MAT01600
      DVEL = 1                                                          MAT01610
      ICYC=0                                                            MAT01620
      IF (BUG) THEN                                                     MAT01630
        WRITE (6,*) '------------- IN MAT04 -------------------------'  MAT01640
        WRITE (6,*) 'P,D',P,D                                           MAT01650
        WRITE (6,*) 'PL,DL=',PL,DL                                      MAT01660
      ENDIF                                                             MAT01670
  310 STIFF=SE(1)                                                       MAT01680
      P=PL+(D-DL)*STIFF                                                 MAT01690
      A=SE(4)                                                           MAT01700
      IF (A.EQ.0 .AND. SE(2).EQ.0) THEN                                 MAT01710
           A=SIGN(SMAT(5),P)                                            MAT01720
           IF (P.LT.PL .AND. PL.GT.0 .AND. P.GT.0) A=0                  MAT01730
           IF (P.GT.PL .AND. PL.LT.0 .AND. P.LT.0) A=0                  MAT01740
      ENDIF                                                             MAT01750
      ICYC=ICYC+1                                                       MAT01760
      IF (BUG) THEN                                                     MAT01770
       WRITE (6,*) 'STIFF=',STIFF,'   A=',A                             MAT01780
       WRITE (6,*) '      P=',P     ,'ICYC=',ICYC                       MAT01790
      ENDIF                                                             MAT01800
C                                                                       MAT01810
C----- IS P WITHIN LIMITS ?                                             MAT01820
      IF (ICYC.GT.10) THEN                                              MAT01830
           WRITE (6,*) 'MORE THAN 10 CYCLES IN MAT04,  IELNO=',IELNO    MAT01840
      ELSE IF ((PL.LE.P .AND. P.LT.A) .OR. (PL.GE.P .AND. P.GT.A) .OR.  MAT01850
     &       (DL.EQ.D)) THEN                                            MAT01860
           IF (BUG) WRITE (6,*) 'C----- IS P WITH IN LIMITS ? '         MAT01870
           CALL STRENG (SE(28),SE(31),SE(32),P,PL,D,DL,SE(1),SE(15) )   MAT01880
           CALL HYST04( P, D,PL,DL,DVEL,NSEG,NI,BUG,                    MAT01890
     &          SMAT( 5),SMAT(J1),SMAT( 5),SMAT(J1),SI,SMAT(3),SMAT(4), MAT01900
     &          SE( 1)  ,SE( 2)  ,SE( 3)  ,SE( 4)  ,SE( 5) ,SE( 6) ,    MAT01910
     &          SE( 7)  ,SE( 8)  ,SE( 9)  ,SE(10)  ,SE(11) ,SE(12) ,    MAT01920
     &          SE(13)  ,SE(14)  ,SE(15)  ,SE(16)  ,SE(19) ,SE(22) ,    MAT01930
     &          SE(28)  ,SE(31)  ,SE(32)  ,SE(33)  ,SE(J3) ,SE(J4) )    MAT01940
C----- IS P GREATER THAN LIMITS ?                                       MAT01950
      ELSE IF (((PL.LE.A .AND. A.LE.P) .OR. (PL.GE.A .AND. A.GE.P)).AND.MAT01960
     &     (STIFF.NE.0.00)) THEN                                        MAT01970
           D1=DL+(A-PL)/STIFF                                           MAT01980
           P1=A                                                         MAT01990
           CALL STRENG (SE(28),SE(31),SE(32),P1,PL,D1,DL,SE(1),SE(15) )MAT02000
           DL=D1                                                        MAT02010
           PL=P1                                                        MAT02020
C          PI=PL+(P-PL)*.001                                            MAT02030
C          DI=DL+(D-DL)*.001                                            MAT02040
           IF (ABS(P-PL).GT. 0.0005*ABS(P+PL)) THEN                     MAT02050
                PI=PL+(P-PL)*.001                                       MAT02060
                DI=DL+(D-DL)*.001                                       MAT02070
           ELSE                                                         MAT02080
                PI=P                                                    MAT02090
                DI=D                                                    MAT02100
           ENDIF                                                        MAT02110
           IF (BUG) THEN                                                MAT02120
             WRITE (6,*) 'C----- IS P GREATER THAN LIMITS ? '           MAT02130
             WRITE (6,*) 'PL,DL=',PL,DL                                 MAT02140
             WRITE (6,*) 'PI,DI=',PI,DI                                 MAT02150
           ENDIF                                                        MAT02160
           CALL HYST04(PI,DI,PL,DL,1.00,NSEG,NI,BUG,                    MAT02170
     &          SMAT( 5),SMAT(J1),SMAT( 5),SMAT(J1),SI,SMAT(3),SMAT(4), MAT02180
     &          SE( 1)  ,SE( 2)  ,SE( 3)  ,SE( 4)  ,SE( 5) ,SE( 6) ,    MAT02190
     &          SE( 7)  ,SE( 8)  ,SE( 9)  ,SE(10)  ,SE(11) ,SE(12) ,    MAT02200
     &          SE(13)  ,SE(14)  ,SE(15)  ,SE(16)  ,SE(19) ,SE(22) ,    MAT02210
     &          SE(28)  ,SE(31)  ,SE(32)  ,SE(33)  ,SE(J3) ,SE(J4) )    MAT02220
           GO TO 310                                                    MAT02230
C----- IS P LESS THAN LIMITS ?                                          MAT02240
      ELSE IF ((P.LT.PL .AND. PL.LE.A) .OR. (P.GT.PL .AND. PL.GE.A))THENMAT02250
C          PI=PL+(P-PL)*.001                                            MAT02260
C          DI=DL+(D-DL)*.001                                            MAT02270
           IF (ABS(P-PL).GT. 0.0005*ABS(P+PL)) THEN                     MAT02280
                PI=PL+(P-PL)*.001                                       MAT02290
                DI=DL+(D-DL)*.001                                       MAT02300
           ELSE                                                         MAT02310
                PI=P                                                    MAT02320
                DI=D                                                    MAT02330
           ENDIF                                                        MAT02340
           IF (BUG) THEN                                                MAT02350
             WRITE (6,*) 'C----- IS P LESS THAN LIMITS ? '              MAT02360
             WRITE (6,*) 'PI,DI=',PI,DI                                 MAT02370
           ENDIF                                                        MAT02380
           CALL HYST04(PI,DI,PL,DL,1.00,NSEG,NI,BUG,                    MAT02390
     &          SMAT( 5),SMAT(J1),SMAT( 5),SMAT(J1),SI,SMAT(3),SMAT(4), MAT02400
     &          SE( 1)  ,SE( 2)  ,SE( 3)  ,SE( 4)  ,SE( 5) ,SE( 6) ,    MAT02410
     &          SE( 7)  ,SE( 8)  ,SE( 9)  ,SE(10)  ,SE(11) ,SE(12) ,    MAT02420
     &          SE(13)  ,SE(14)  ,SE(15)  ,SE(16)  ,SE(19) ,SE(22) ,    MAT02430
     &          SE(28)  ,SE(31)  ,SE(32)  ,SE(33)  ,SE(J3) ,SE(J4) )    MAT02440
           GO TO 310                                                    MAT02450
      ENDIF                                                             MAT02460
C                                                                       MAT03470
      IF (BUG)  WRITE (6,*) 'STIFF=',STIFF,'   A=',SE(4)                MAT03480
C                                                                       MAT02490
C------ SAVE MAXIMUN DISPL AND LOAD                                     MAT02500
      IF (ABS(D).GT.ABS(SE(34+3*NI))) THEN                              MAT02510
           SE(33+  3*NI)=P                                              MAT02520
           SE(34+  3*NI)=D                                              MAT02530
      ENDIF                                                             MAT02540
C                                                                       MAT02550
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                        MAT02560
  400 EESE=SE(28)                                                       MAT02570
      EPSE=SE(31)                                                       MAT02580
C                                                                       MAT02590
C------ TRANSFER DUCTILITIES AND EXCRUSION RATIOS FOR BOTH IOPT=3 AND 4.MAT02600
      DO 410 I=1,3                                                      MAT02610
      DUCT(I)=MAX(SE(15+I),SE(18+I))                                    MAT02620
      EXCR(I)=SE(21+I)                                                  MAT02630
  410 EXCR(I+3)=SE(21+I+3)                                              MAT02640
      RETURN                                                            MAT02650
C                                                                       MAT02660
C--------------------------- DAMAGE INDEX ----------------------------- MAT02670
  500 EESE =SE(28)                                                      MAT02680
      EPSE =SE(31)                                                      MAT02690
      NSEGS=SMAT(1)                                                     MAT02700
      NI   =SMAT(2)                                                     MAT02710
      DMAX =SMAT(4+2*NSEGS)                                             MAT02720
      DY   =SMAT(4)                                                     MAT02730
      BDAM =SMAT(5+2*NSEG)                                              MAT02740
      P=SE(33+  3*NI)                                                   MAT02750
      D=SE(34+  3*NI)                                                   MAT02760
      IF (DY.LE.SMAT(5+NSEGS)) THEN                                     MAT02770
           PY=SMAT(5)*DY/SMAT(5+NSEGS)                                  MAT02780
      ELSE                                                              MAT02790
           DO 510 I=2,NSEGS                                             MAT02800
             P1=SMAT(3+I)                                               MAT02810
             P2=SMAT(4+I)                                               MAT02820
             D1=SMAT(3+I+NSEGS)                                         MAT02830
             D2=SMAT(4+I+NSEGS)                                         MAT02840
             IF (DY.GE.D1 .AND. DY.LE.D2)  THEN                         MAT02850
               PY=P1+(DY-D1)*(P2-P1)/(D2-D1)                            MAT02860
               GO TO 520                                                MAT02870
             ENDIF                                                      MAT02880
  510        CONTINUE                                                   MAT02890
      ENDIF                                                             MAT02900
  520 DAMAGE=ABS(D)/DMAX + BDAM / (PY*DMAX) * EPSE                      MAT02910
C                                                                       MAT02920
      RETURN                                                            MAT02930
C1111 FORMAT (1P,                                                       MAT02940
C     &' S1A- P=',G13.5,'  D=',G13.5,'  K=',G18.9,'  RULE=',G13.5,      MAT02950
C     & ' DIR=',F6.3,'  A=',G13.5,' IR=',I3)                            MAT02960
C1112 FORMAT (1P,                                                       MAT02970
C     &' S1B- P=',G13.5,'  D=',G13.5,'  K=',G18.9,'  RULE=',G13.5,      MAT02980
C     & ' DIR=',F6.3,'  A=',G13.5,' IR=',I3)                            MAT02990
C1113 FORMAT (1P,                                                       MAT03000
C     &' S1C- P=',G13.5,'  D=',G13.5,'  K=',G18.9,'  RULE=',G13.5,      MAT03010
C     & ' DIR=',F6.3,'  A=',G13.5,' IR=',I3)                            MAT03020
      END                                                               MAT03030
C-----------------------------------------------------------------------MAT03040
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    MAT03020
C           END DEBUG                                                   MAT03030
C-----------------------------------------------------------------------MAT03040
      SUBROUTINE MAT06                                                  MAT03050
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG,                      MAT03060
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                       MAT03070
     &   SMAT,ESSE,EPSE,DUCT,EXCR,DAMAGE)                               MAT03080
C                                                                       MAT03090
      IMPLICIT REAL(A-H,O-Z)                                            MAT03100
      LOGICAL ERR,FIRST,PRINT,BUG                                       MAT03110
      LOGICAL BTEST                                                     MAT03120
      DIMENSION SE(100),SMAT(13),DUCT(3),EXCR(8)                        MAT03130
C                                                                       MAT03140
      LELEM=49                                                          MAT03150
      IF (IOPT.EQ.0) RETURN                                             MAT03160
C                                                                       MAT03170
C-----------------------------------------------------------------------MAT03180
C VARIABLES:                                                            MAT03190
C-----------------------------------------------------------------------MAT03200
C---------- GLOBAL VARIABLES                                           MAT03210
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT03220
C        = 2, GET STIFFNESS                                             MAT03230
C RINPUT = INPUT DATA                                                   MAT03240
C SMAT   = INTERNAL STORAGE                                             MAT03250
C SE     = OUTPUT STIFFNESS                                             MAT03260
C-----------------------------------------------------------------------MAT03270
C           ---- TAKEDA DATA ----                                       MAT03280
C        SMAT( 1)= EI,            ELASTIC STIFFNESS                     MAT03290
C        SMAT( 2)= PC,            CRACKING LOAD                         MAT03300
C        SMAT( 3)= DC,            CRACKING DISPLACEMENT                 MAT03310
C        SMAT( 4)= PY,            YEILD LOAD                            MAT03320
C        SMAT( 5)= DY,            YEILD DISPLACEMENT                    MAT03330
C        SMAT( 6)= PU,            ULTIMATE LOAD                         MAT03340
C        SMAT( 7)= DU,            ULTIMATE DISPLACEMENT                 MAT03350
C        SMAT( 8)= OC,            INITIAL STIFFNESS                     MAT03360
C        SMAT( 9)= CI,            CRACKED STIFFNESS                     MAT03370
C        SMAT(10)= YU,            ULTIMATE STIFFNESS                    MAT03380
C        SMAT(11)= C'Y,           RELOADING STIFFNESS                   MAT03390
C        SMAT(12)= CSE            CONSTANT STRAIN ENERGY AT YEILD       MAT03400
C        SMAT(13)= BDAM           BETA FOR DAMAGE INDEX                 MAT03410
C                                                                       MAT03420
C------------------- INPUT DATA FOR TAKEDA HYSTERESIS MODEL ------------CMAT03430
C     THE FOLLOWING DATA IS INPUT ONCE FOR EACH MODEL                   CMAT03440
C                                                                       CMAT03450
C  EI       = MOMENT OF INERTIA                                         CMAT03460
C  G        = MOMENT GRADIENT                                           CMAT03470
C  PC,DC    = CRACKING LOAD AND DISPLACEMENT                            CMAT03480
C  PY,DY    = YEILD LOAD AND DISPLACEMENT                               CMAT03490
C  PU,DU    = ULTIMATE LOAD AND DISPLACEMENT                            CMAT03500
C                                                                       CMAT03510
C-----------------------------------------------------------------------CMAT03520
      GO TO (100,200,300,400,500),IOPT                                  MAT03530
C                                                                       MAT03540
  100 CONTINUE                                                          MAT03550
C     READ (5,*) TYPE ,SMAT(1),SMAT(12),(SMAT(I),I=2,7)                MAT03560
      READ (5,*) TYPE ,SMAT(1),          (SMAT(I),I=2,7),BDAM           MAT03570
C                                                                       MAT03580
      LMAT = 13                                                         MAT03590
      SMAT( 8)= SMAT(2)           / SMAT(3)                             MAT03600
      SMAT( 9)=(SMAT(4)-SMAT(2))/(SMAT(5)-SMAT(3))                      MAT03610
      SMAT(10)=(SMAT(6)-SMAT(4))/(SMAT(7)-SMAT(5))                      MAT03620
      SMAT(11)=(SMAT(2)+SMAT(4))/(SMAT(3)+SMAT(5))                      MAT03630
      SMAT( 1)=SMAT(1)                                                  MAT03640
      SMAT(12)=SMAT(2)*SMAT(3)/2 +.5*(SMAT(2)+SMAT(4))*(SMAT(5)-SMAT(3))MAT03650
      SMAT(13)=BDAM                                                     MAT03660
C                                                                       MAT03670
      IF (FIRST .OR. BUG) WRITE (6,5)                                   MAT03680
      WRITE (6,6) MAT,BDAM,SMAT(1),     SMAT(2),SMAT(4),SMAT(5),        MAT03690
     &                                  SMAT(3),SMAT(5),SMAT(7)         MAT03700
    5 FORMAT(///'     TAKEDA HYSTERESIS MODEL- FOR UNIT LENGTH MEMBER'/ MAT03710
     &       '     '  '**********************************************'//MAT03720
     &   ' MAT.     BETA          EI          ',                        MAT03730
     &   7X,'CRACK',10X,'YEILD', 7X,'ULTIMATE')                         MAT03740
    6 FORMAT (/1X,I5,2G15.6,1X, 2X,3G15.6,/39X,3G15.6/)                 MAT03750
C   WRITE (6,6) MAT,SMAT(1),SMAT(12),SMAT(2),SMAT(4),SMAT(6),          MAT03760
C    &                               SMAT(3),SMAT(5),SMAT(7)           MAT03770
C 5 FORMAT(///'     TAKEDA HYSTERESIS MODEL- FOR UNIT LENGTH MEMBER'/ MAT03780
C    &       '     '  '**********************************************'//MAT03790
C    &   ' MAT.  ELASTIC EI     MONT. GRADIENT  ',                      MAT03800
C    &   7X,'CRACK',10X,'YEILD',10X,'ULTIMATE',// )                     MAT03810
C 6 FORMAT (1X,I5,G15.6,1X,G15.6,2X,3G15.6,/39X,3G15.6/)                MAT03820
C                                                                       MAT03830
      RETURN                                                            MAT03840
C---------------------- GET STIFFNESS TERMS ----------------------------MAT03850
  200 CONTINUE                                                          MAT03860
C           ---- TAKEDA MODEL DATA STORAGE FOR MEMBER ----             MAT03870
C       SE( 1)= G                                                       MAT03880
C       SE( 2)= HRN                                                     MAT03890
C       SE( 3)= A                                                       MAT03900
C       SE( 4)= B                                                       MAT03910
C       SE( 5)= IDRO                                                    MAT03920
C       SE( 6)= RF                                                      MAT03930
C       SE( 7)= RNRL                                                    MAT03940
C       SE( 8)= UM1                                                     MAT03950
C       SE( 9)= UM2                                                     MAT03960
C       SE(10)= UM3                                                     MAT03970
C       SE(11)= UM4                                                     MAT03980
C       SE(12)= S1                                                      MAT03990
C       SE(13)= X0                                                      MAT01000
C       SE(14)= X0UM                                                    MAT01010
C       SE(15)= X0Y                                                     MAT01020
C       SE(16)= U0D                                                     MAT01030
C       SE(17)= U0                                                      MAT01040
C       SE(18)= X1UM                                                    MAT01050
C       SE(19)= U1                                                      MAT01060
C       SE(20)= U1D                                                     MAT01070
C       SE(21)= X2U0                                                    MAT01080
C       SE(22)= U2                                                      MAT01090
C       SE(23)= X3U1                                                    MAT01100
C       SE(24)= U3                                                      MAT01110
C       SE(25)= QY                                                      MAT01120
C       SE(26)= MAXFLG                                                  MAT01130
C       SE(27)= ACHNG                                                   MAT01140
C       SE(28)= S_EQ_UNLOAD                                             MAT01150
C       SE(29)= ESSE                                                    MAT01160
C       SE(30)= EESE                                                    MAT01170
C       SE(31)= EESE                                                    MAT01180
C       SE(32)= EPSE                                                    MAT01190
C       SE(33)= PSE_OLD                                                 MAT01200
C       SE(34)= UPOS(1)                                                 MAT01210
C       SE(35)= UPOS(2)                                                 MAT01220
C       SE(36)= UPOS(3)                                                 MAT01230
C       SE(37)= UNEG(1)                                                 MAT01240
C       SE(38)= UNEG(2)                                                 MAT01250
C       SE(39)= UNEG(3)                                                 MAT01260
C       SE(40)= EXCR(1)                                                 MAT01270
C       SE(41)= EXCR(2)                                                 MAT01280
C       SE(42)= EXCR(3)                                                 MAT01290
C       SE(43)= EXCR(4)                                                 MAT01300
C       SE(44)= EXCR(5)                                                 MAT01310
C       SE(45)= EXCR(6)                                                 MAT01320
C       SE(46)= P_MAX                                                   MAT01330
C       SE(47)= D_MAX                                                   MAT01340
C       SE(48)= SMAT(8)                                                 MAT01350
C       SE(49)= SMAT(1)                                                 MAT01360
C                                                                       MAT01370
      EI=SMAT(1)                                                        MAT01380
      GRAD=SMAT(12)                                                     MAT01390
```

```
      DO 210 I=1,47                                                     MAT01400
  210   SE(I)=0                                                         MAT01410
      CALL HYST06(P,D,PL,DL,V ,VL,                                      MAT01420
     & SE( 1),SE( 2),SE( 3),SE( 4),SE( 5),SE( 6),                       MAT01430
     & SMAT( 2),SMAT( 3),SMAT( 4),SMAT( 5),SMAT( 6),SMAT( 7),SMAT( 8),  MAT01440
     & SMAT( 9),SMAT(10),SMAT(11),SMAT(12),                             MAT01450
     & SE( 7),SE( 8),SE( 9),SE(10),SE(11),SE(12),SE(13),SE(14),SE(15),  MAT01460
     & SE(16),SE(17),SE(18),SE(19),SE(20),SE(21),SE(22),SE(23),SE(24),  MAT01470
     & SE(25),SE(26),SE(27),BUG,                                        MAT01480
     & SE(28),SE(29),SE(32),SE(33),SE(34),SE(37),SE(40))                MAT01490
C                                                                       MAT01500
      SE(48)=SMAT(8)                                                    MAT01510
      SE(49)=EI                                                         MAT01520
C                                                                       MAT01530
      RETURN                                                            MAT01540
C                                                                       MAT01550
C------------------------ GET STIFFNESS TERMS ------------------------- MAT01560
  300 CONTINUE                                                          MAT01570
C     SE( 1)= STIFF                                                     MAT01580
C      SE(27)= A                                                        MAT01590
C      SE(28)= S_EQ_UNLOAD                                              MAT01600
      ICYC=0                                                            MAT01610
      IF (BUG) THEN                                                     MAT01620
        WRITE (6,*) '------------- IN MAT06 -------------------------'   MAT01630
        WRITE (6,*) '   P,D=',P,D   ,'  V=',V                          MAT01640
        WRITE (6,*) 'PL,DL=',PL,DL,' VL=',VL                            MAT01650
      ENDIF                                                             MAT01660
  310 STIFF=SE(1)                                                       MAT01670
      P=PL+(D-DL)*STIFF                                                 MAT01680
      A=SE(27)                                                          MAT01690
      IF (A.EQ.0 .AND. SE(7).EQ.0) A=SMAT(2)                            MAT01700
      ICYC=ICYC+1                                                       MAT01710
      IF (BUG) THEN                                                     MAT01720
       WRITE (6,*) 'STIFF=',STIFF,'   A=',A                             MAT01730
       WRITE (6,*) '    P=',P    ,'ICYC=',ICYC                          MAT01740
      ENDIF                                                             MAT01750
C                                                                       MAT01760
C----- IS P WITHIN LIMITS ?                                             MAT01770
      IF (ICYC.GT.5) THEN                                               MAT01780
CC          WRITE (6,*) 'MORE THAN 5 CYCLES IN MAT06'                   MAT01790
            WRITE (6,*) 'MORE THAN 5 CYCLES IN MAT06,  IELNO=',IELNO    MAT01800
C--   ELSE IF ((PL.LE.P .AND. P.LT.A) .OR. (PL.GE.P .AND. P.GT.A) .OR.  MAT01810
      ELSE IF ((PL.LE.P .AND. P.LE.A) .OR. (PL.GE.P .AND. P.GE.A) .OR.  MAT01820
     &     (DL.EQ.D)) THEN                                              MAT01830
            IF (BUG) WRITE (6,*) 'C----- IS P WITH IN LIMITS ? '        MAT01840
            CALL STRENG(SE(29),SE(32),SE(33),P,PL,D,DL,SE(1),SE(28))    MAT01850
            CALL HYST06( P,D ,PL,DL,V ,VL,                              MAT01860
     &      SE( 1),SE( 2),SE( 3),SE( 4),SE( 5),SE( 6),                 MAT01870
     &      SMAT( 2),SMAT( 3),SMAT( 4),SMAT( 5),SMAT( 6),              MAT01880
     &      SMAT( 7),SMAT( 8),SMAT( 9),SMAT(10),SMAT(11),SMAT(12),     MAT01890
     &      SE( 7),SE( 8),SE( 9),SE(10),SE(11),SE(12),SE(13),          MAT01900
     &      SE(14),SE(15),SE(16),SE(17),SE(18),SE(19),SE(20),          MAT01910
     &      SE(21),SE(22),SE(23),SE(24),SE(25),SE(26),SE(27),BUG,      MAT01920
     &      SE(28),SE(29),SE(32),SE(33),SE(34),SE(37),SE(40))          MAT01930
C           V=VL                                                        MAT01940
C----- IS P GREATER THAN LIMITS ?                                       MAT01950
      ELSE IF (((PL.LE.A .AND. A.LT.P) .OR. (PL.GE.A .AND. A.GE.P)).AND.MAT01960
     &     (STIFF.NE.0.00)) THEN                                        MAT01970
            DI=DL+(A-PL)/STIFF                                          MAT01980
            PI=A                                                        MAT01990
            CALL STRENG(SE(29),SE(32),SE(33),PI,PL,DI,DL,SE(1),SE(28))  MAT02000
            IF (BUG) THEN                                               MAT02010
              WRITE (6,*) 'C----- IS P GREATER THAN LIMITS ? '          MAT02020
              WRITE (6,*) 'PL,DL=',PL,DL                                MAT02030
              WRITE (6,*) 'PI,DI=',PI,DI                                MAT02040
            ENDIF                                                       MAT02050
            CALL HYST06(PI,DI,PL,DL,1.,1.,                              MAT02060
     &      SE( 1),SE( 2),SE( 3),SE( 4),SE( 5),SE( 6),                 MAT02070
     &      SMAT( 2),SMAT( 3),SMAT( 4),SMAT( 5),SMAT( 6),              MAT02080
     &      SMAT( 7),SMAT( 8),SMAT( 9),SMAT(10),SMAT(11),SMAT(12),     MAT02090
     &      SE( 7),SE( 8),SE( 9),SE(10),SE(11),SE(12),SE(13),          MAT02100
     &      SE(14),SE(15),SE(16),SE(17),SE(18),SE(19),SE(20),          MAT02110
     &      SE(21),SE(22),SE(23),SE(24),SE(25),SE(26),SE(27),BUG,      MAT02120
     &      SE(28),SE(29),SE(32),SE(33),SE(34),SE(37),SE(40))          MAT02130
            DL=DI                                                       MAT02140
            PL=PI                                                       MAT02150
C                                                                       MAT02160
C           DL=DI                                                       MAT02170
C           PL=PI                                                       MAT02180
C           PI=PL+(P-PL)*.001                                           MAT02190
C           DI=DL+(D-DL)*.001                                           MAT02200
C           IF (BUG) THEN                                               MAT02210
C             WRITE (6,*) 'C----- IS P GREATER THAN LIMITS ? '          MAT02220
C             WRITE (6,*) 'PL,DL=',PL,DL                                MAT02230
C             WRITE (6,*) 'PI,DI=',PI,DI                                MAT02240
C           ENDIF                                                       MAT02250
C           CALL HYST06(PI,DI,PL,DL,1.,1.,                              MAT02260
C    &      SE( 1),SE( 2),SE( 3),SE( 4),SE( 5),SE( 6),                 MAT02270
C    &      SMAT( 2),SMAT( 3),SMAT( 4),SMAT( 5),SMAT( 6),              MAT02280
C    &      SMAT( 7),SMAT( 8),SMAT( 9),SMAT(10),SMAT(11),SMAT(12),     MAT02290
C    &      SE( 7),SE( 8),SE( 9),SE(10),SE(11),SE(12),SE(13),          MAT02300
C    &      SE(14),SE(15),SE(16),SE(17),SE(18),SE(19),SE(20),          MAT02310
C    &      SE(21),SE(22),SE(23),SE(24),SE(25),SE(26),SE(27),BUG,      MAT02320
C    &      SE(28),SE(29),SE(32),SE(33),SE(34),SE(37),SE(40))          MAT02330
            GO TO 310                                                   MAT02340
C----- IS P LESS THAN LIMITS ?                                          MAT02350
      ELSE IF ((P.LT.PL .AND. PL.LE.A) .OR. (P.GT.PL .AND. PL.GE.A)).THENMAT02360
            IF (ABS(P-PL).GT. 0.0005*ABS(P+PL)) THEN                    MAT02370
                 PI=PL+(P-PL)*.001                                      MAT02380
                 DI=DL+(D-DL)*.001                                      MAT02390
            ELSE                                                        MAT02400
                 PI=P                                                   MAT02410
                 DI=D                                                   MAT02420
            ENDIF                                                       MAT02430
C           PI=PL+(P-PL)*.001                                           MAT02440
C           DI=DL+(D-DL)*.001                                           MAT02450
            IF (BUG) THEN                                               MAT02460
              WRITE (6,*) 'C----- IS P LESS THAN LIMITS ? '             MAT02470
              WRITE (6,*) 'PI,DI=',PI,DI                                MAT02480
            ENDIF                                                       MAT02490
            CALL HYST06(PI,DI,PL,DL,1.,1.,                              MAT02500
     &      SE( 1),SE( 2),SE( 3),SE( 4),SE( 5),SE( 6),                 MAT02510
     &      SMAT( 2),SMAT( 3),SMAT( 4),SMAT( 5),SMAT( 6),              MAT02520
     &      SMAT( 7),SMAT( 8),SMAT( 9),SMAT(10),SMAT(11),SMAT(12),     MAT02530
     &      SE( 7),SE( 8),SE( 9),SE(10),SE(11),SE(12),SE(13),          MAT02540
     &      SE(14),SE(15),SE(16),SE(17),SE(18),SE(19),SE(20),          MAT02550
     &      SE(21),SE(22),SE(23),SE(24),SE(25),SE(26),SE(27),BUG,      MAT02560
     &      SE(28),SE(29),SE(32),SE(33),SE(34),SE(37),SE(40))          MAT02570
            GO TO 310                                                   MAT02580
      ENDIF                                                             MAT02590
C                                                                       MAT02600
      IF (BUG)  WRITE (6,*) 'STIFF=',STIFF,' A,B=',SE(3),SE(4),SE(27)   MAT02610
C                                                                       MAT02620
C------ MAXIMUM DISPLACEMENT                                            MAT02630
      IF (ABS(D).GT.ABS(SE(47))) THEN                                   MAT02640
            SE(46)= P                                                   MAT02650
            SE(47)= D                                                   MAT02660
      ENDIF                                                             MAT02670
C                                                                       MAT02680
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                        MAT02690
  400 EESE=SE(29)                                                       MAT02700
      EPSE=SE(32)                                                       MAT02710
C                                                                       MAT02720
C------ TRANSFER DUCTILITIES AND EXCRUSION RATIOS FOR BOTH IOPT=3 AND 4.MAT02730
      DO 410 I=1,3                                                      MAT02740
      DUCT(I)=MAX(SE(33+I),SE(36+I))                                    MAT02750
      EXCR(I)=SE(39+I)                                                  MAT02760
  410 EXCR(I+3)=SE(42+I)                                                MAT02770
      RETURN                                                            MAT02780
C                                                                       MAT02790
C------------------ DAMAGE INDEX ------------------------------------- MAT02800
  500 EESE=SE(29)                                                       MAT02810
      EPSE=SE(32)                                                       MAT02820
      PY=  SMAT( 4)                                                     MAT02830
      PU=  SMAT( 6)                                                     MAT02840
      DU=  SMAT( 7)                                                     MAT02850
      P =  SE(46)                                                       MAT02860
      D =  SE(47)                                                       MAT02870
      BDAM=SMAT(13)                                                     MAT02880
      DAMAGE=ABS(D)/DU + BDAM/(PY*DU) * EPSE                            MAT02890
C                                                                       MAT02900
      RETURN                                                            MAT02910
      END                                                               MAT02920
C-----------------------------------------------------------------------MAT00010
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    MAT00020
C           END DEBUG                                                   MAT00030
C-----------------------------------------------------------------------MAT00040
      SUBROUTINE MAT07                                                  MAT00050
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG,                      MAT00060
     &   MAT    ,SE   ,P,PL,D,DL,V,VL,LELEM,LMAT,                       MAT00070
     &   SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)                              MAT00080
C                                                                       MAT00090
      IMPLICIT REAL(A-H,O-Z)                                            MAT00100
      LOGICAL ERR,FIRST,PRINT,BUG                                       MAT00110
      LOGICAL BTEST                                                     MAT00120
      DIMENSION SE(100),SMAT(7),DUCT(3),EXCR(6)                         MAT00130
      CHARACTER*80 TYPE                                                 MAT00140
C                                                                       MAT00150
C------- RESERVE STORAGE FOR ELEMENT                                    MAT00160
      LELEM=24                                                          MAT00170
      IF (IOPT.EQ.0) RETURN                                             MAT00180
C                                                                       MAT00190
C-----------------------------------------------------------------------MAT00200
C VARIABLES:                                                            MAT00210
C-----------------------------------------------------------------------MAT00220
C---------- GLOBAL VARIABLES                                           MAT00230
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT00240
C        = 2, GET STIFFNESS                                             MAT00250
C RINPUT = INPUT DATA                                                   MAT00260
C SMAT   = INTERNAL STORAGE                                             MAT00270
C SE     = OUTPUT STIFFNESS                                             MAT00280
C-----------------------------------------------------------------------MAT00290
      GO TO (100,200,300,400,500),IOPT                                  MAT00300
C                                                                       MAT00310
  100 CONTINUE                                                          MAT00320
C                                                                       MAT00330
C----- INTERPERTATE INPUT DATA                                          MAT00340
C     KE    = SMAT( 1)                                                  MAT00350
C     PY    = SMAT( 2)                                                  MAT00360
C     KIE   = SMAT( 3)                                                  MAT00370
C     DY    = SMAT( 4)                                                  MAT00380
C     CSE   = SMAT( 5)                                                  MAT00390
C     DMAX  = SMAT( 6)                                                  MAT00400
C     BDAM  = SMAT( 7)                                                  MAT00410
C                                                                       MAT00420
      READ (5,*) TYPE ,(SMAT(I),I=1,3),DUCMAX,BDAM                      MAT00430
C                                                                       MAT00440
      LMAT = 7                                                          MAT00450
      SMAT(4)=SMAT(2)/SMAT(1)                                           MAT00460
      SMAT(5)=SMAT(4)*SMAT(2)/2                                         MAT00470
      SMAT(6)=SMAT(4)*DUCMAX                                            MAT00480
      SMAT(7)=BDAM                                                      MAT00490
C                                                                       MAT00500
      IF (FIRST .OR. BUG) WRITE (6,191)                                 MAT00510
      WRITE (6,192) MAT,(SMAT(K),K=1,3),DUCMAX,BDAM                     MAT00520
C                                                                       MAT00530
  191 FORMAT (///'     BILINEAR HYSTERESIS MODEL PARAMETERS'/           MAT00540
     &        '     ------------------------------------'//            MAT00550
     &  ' MAT.',6X,'KE',10X,'PY',10X,'KIE', 7X,'MAX DUCT',             MAT00560
     &  10X,' BDAM')                                                    MAT00570
  192 FORMAT (1X,I5,7G15.6/)                                            MAT00580
C                                                                       MAT00590
      RETURN                                                            MAT00600
C                                                                       MAT00610
C------------------------- INITILIZE STIFFNESS ------------------------ MAT00620
  200 CONTINUE                                                          MAT00630
C         ---- ELASTO-PLASTIC MODEL DATA STORAGE FOR MEMBER ----       MAT00640
C     SE( 1)= K                                                         MAT00650
C     SE( 2)= RULE                                                      MAT00660
C     SE( 3)= EPSE_OLD                                                  MAT00670
C     SE( 4)= A, UPPER LIMIT BEFORE RULE CHANGE                         MAT00680
C     SE( 5)= B, LOWER LIMIT BEFORE RULE CHANGE                         MAT00690
C     SE( 6)=EESE(1)                                                    MAT00700
C     SE( 7)=EESE(2)                                                    MAT00710
C     SE( 8)=EESE(3)                                                    MAT00720
C     SE( 9)=EPSE                                                       MAT00730
C     SE(10)=KEY                                                        MAT00740
C     SE(11)=UPOS(1)                                                    MAT00750
C     SE(12)=UPOS(2)                                                    MAT00760
C     SE(13)=UPOS(3)                                                    MAT00770
C     SE(14)=UNEG(1)                                                    MAT00780
C     SE(15)=UNEG(2)                                                    MAT00790
C     SE(16)=UNEG(3)                                                    MAT00800
C     SE(17)=EXCR(1)                                                    MAT00810
C     SE(18)=EXCR(2)                                                    MAT00820
C     SE(19)=EXCR(3)                                                    MAT00830
C     SE(20)=EXCR(4)                                                    MAT00840
C     SE(21)=EXCR(5)                                                    MAT00850
C     SE(22)=EXCR(6)                                                    MAT00860
C     SE(23)=P_MAX                                                      MAT00870
C     SE(24)=D_MAX                                                      MAT00880
C                                                                       MAT00890
C     INITILIZE RULE #1 - ELASTIC BEHAVIOR                              MAT00900
C                                                                       MAT00910
      DO 210 I=1,22                                                     MAT00920
  210   SE(I)=0                                                         MAT00930
C                                                                       MAT00940
      CALL HYST07(0.,0.,0.,0.,SE(1),SE(2),SE(3),SE(4),SE(5),V,VL,      MAT00950
     &     SMAT(1),SMAT(2),SMAT(3),SMAT(4),SMAT(5),BUG,                 MAT00960
     &     SE(6),SE(9),SE(10),SE(11),SE(14),SE(17))                     MAT00970
C                                                                       MAT00980
      RETURN                                                            MAT00990
C                                                                       MAT01000
C------------- GET STIFFNESS TERMS AND ACTUAL FORCE -------------------MAT01010
  300 CONTINUE                                                          MAT01020
C                                                                       MAT01030
C--------- CHECK TO SEE IF DD = 0 ?                                     MAT01040
      P=P                                                               MAT01050
      D=D                                                               MAT01060
      IF (D.EQ.DL)  RETURN                                              MAT01070
C                                                                       MAT01080
C--------- CALL HIST07 TO CALC NEW STIFFNESS AND GET LOAD P             MAT01090
      CALL HYST07(P,D,PL,DL,SE(1),SE(2),SE(3),SE(4),SE(5),V,VL,         MAT01100
     &     SMAT(1),SMAT(2),SMAT(3),SMAT(4),SMAT(5),BUG,                 MAT01110
     &     SE(6),SE(9),SE(10),SE(11),SE(14),SE(17))                     MAT01120
C                                                                       MAT01130
      SE1=SE(1)                                                         MAT01140
C                                                                       MAT01150
C------- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                       MAT01160
      IF (ABS(D).GT.ABS(SE(24)))THEN                                    MAT01170
          SE(23)=P                                                      MAT01180
          SE(24)=D                                                      MAT01190
      ENDIF                                                             MAT01200
C                                                                       MAT01210
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                        MAT01220
  400 EESE=SE(6)                                                        MAT01230
      EPSE=SE(9)                                                        MAT01240
C                                                                       MAT01250
C------ TRANSFER DUCTILITIES AND EXCRUSION RATIOS FOR BOTH IOPT=3 AND 4.MAT01260
      DO 410 I=1,3                                                      MAT01270
      DUCT(I)=MAX(SE(10+I),SE(13+I))                                    MAT01280
      EXCR(I+3)=SE(16+I+3)                                              MAT01290
  410 EXCR(I)=SE(16+I)                                                  MAT01300
      RETURN                                                            MAT01310
```

```
C                                                                       MAT01320
C*------------------------------ DAMAGE INDEX ---------------------------MAT01330
  500 EESE=SE(6)                                                        MAT01340
      EPSE=SE(9)                                                        MAT01350
      PY    = SMAT( 2)                                                  MAT01360
      DY    = SMAT( 4)                                                  MAT01370
      DMAX  = SMAT( 6)                                                  MAT01380
      BDAM  = SMAT( 7)                                                  MAT01390
      P=   SE(23)                                                       MAT01400
      D=   SE(24)                                                       MAT01410
      DAMAGE=ABS(D)/DMAX + BDAM / (PY*DMAX) * EPSE                      MAT01420
C                                                                       MAT01430
      RETURN                                                            MAT01440
      END                                                               MAT01450
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP                                  MAT00010
C*---------------------------------------------------------------------MAT00020
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    MAT00030
C           END DEBUG                                                   MAT00040
C*---------------------------------------------------------------------MAT00050
      SUBROUTINE MAT08                                                  MAT00060
     &  (IOPT    ,IELNO    ,ERR ,FIRST  ,PRINT ,BUG,                    MAT00070
     &   MAT     ,SE   ,P,PL,D,DL,V,VL,LELEM,LMAT,                      MAT00080
     &   SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)                              MAT00090
C                                                                       MAT00100
      IMPLICIT REAL(A-H,O-Z)                                            MAT00110
      LOGICAL ERR,FIRST,PRINT,BUG                                       MAT00120
      LOGICAL BTEST                                                     MAT00130
      DIMENSION SE(100),SMAT(9),DUCT(3),EXCR(6)                         MAT00140
      CHARACTER*80 TYPE                                                 MAT00150
C                                                                       MAT00160
C------- RESERVE STORAGE FOR ELEMENT                                    MAT00170
      LELEM=39                                                          MAT00180
      IF (IOPT.EQ.0) RETURN                                             MAT00190
C                                                                       MAT00200
C---------------------------------------------------------------------- MAT00210
C VARIABLES:                                                            MAT00220
C---------------------------------------------------------------------- MAT00230
C---------- GLOBAL VARIABLES                                           MAT00240
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT00250
C        = 2, GET STIFFNESS                                             MAT00260
C RINPUT = INPUT DATA                                                   MAT00270
C SMAT   = INTERNAL STORAGE                                             MAT00280
C SE     = OUTPUT STIFFNESS                                             MAT00290
C---------------------------------------------------------------------- MAT00300
      GO TO (100,200,300,400,500),IOPT                                  MAT00310
C                                                                       MAT00320
  100 CONTINUE                                                          MAT00330
C                                                                       MAT00340
C----- INTERPERTATE INPUT DATA                                          MAT00350
C       E    = SMAT( 1)                                                 MAT00360
C       A    = SMAT( 2)                                                 MAT00370
C       R    = SMAT( 3)                                                 MAT00380
C     TS     = SMAT( 4)                                                 MAT00390
C     PY     = SMAT( 5)                                                 MAT00400
C     CC     = SMAT( 6)                                                 MAT00410
C     DMAX   = SMAT( 7)   : ULTIMATE DISPLACEMENT = DUCMAX*DELY         MAT00420
C     BDAM   = SMAT( 8)   : BATA PARAMETER                              MAT00430
C     SHAPE  = SMAT( 9)   : CROSS SECTION SHAPE                         MAT00440
C                           1: FOR BOX OR ANGLE SECTION                 MAT00450
C                           2: FOR I SHAPE SECTION                      MAT00460
C                           3: FOR BOX SECTION CONSIDERING              MAT00470
C                              LOCAL BUCKLING. THIS IS ONLY FOR MEMBER  MAT00480
C                              WITH KL/R LESS THAN 40 CASE              MAT00490
C---------------------------------------------------------------------- MAT00500
C                                                                       MAT00510
      READ (5,*) TYPE ,(SMAT(I),I=1,4), SHAPE,DUCMAX, BDAM             MAT00520
C                                                                       MAT00530
      LMAT = 9                                                          MAT00540
      PI=3.1415926                                                      MAT00550
      SMAT(5)=SMAT(2)*SMAT(4)                                           MAT00560
      SMAT(6)=SQRT(2.*PI*PI*SMAT(1)/SMAT(4) )                           MAT00570
      SMAT(7)=DUCMAX                                                    MAT00580
      SMAT(8)=BDAM                                                      MAT00590
      SMAT(9)=SHAPE                                                     MAT00600
C                                                                       MAT00610
      IF (FIRST .OR. BUG) WRITE (6,191)                                 MAT00620
      WRITE (6,192) MAT,(SMAT(K),K=1,4),SHAPE,DUCMAX,BDAM               MAT00630
C                                                                       MAT00640
 191  FORMAT (///'     GOEL HYSTERESIS MODEL PARAMETERS'/                MAT00650
     &        '     ------------------------------'//                   MAT00660
     &  '  MAT.',3X,'    E',6X,'    A',6X,'    R',6X,'   TS',          MAT00670
     &       6X,' SHAPE',6X, 'MAX DUC', 6X,'  BETA')                    MAT00680
 192  FORMAT (1X,I5,7G12.6/)                                            MAT00690
C                                                                       MAT00700
      RETURN                                                            MAT00710
C                                                                       MAT00720
C*------------------------ INITIALIZE STIFFNESS TERMS --------------------MAT00730
  200 CONTINUE                                                          MAT00740
C         ---- GOEL HYSTERESIS MODEL DATA STORAGE FOR MEMBER ----       MAT00750
C     SE( 1)=          STIFMEMBER AXIAL STIFFNESS                       MAT00760
C     SE( 2)= K        EFFECT SLENDER COEFFICIENT (TRANSFERD FROM ELE08) MAT00770
C     SE( 3)= NOCY     NO. OF CYCLE                                     MAT00780
C     SE( 4)= NRULE    RULE NUMBER                                      MAT00790
C     SE( 5)= EL       MEMBER LENGTH (TRANSFERED FROM ELE08)            MAT00800
C     SE( 6)= IRV      0:REVRSC=.FALSE.    1:REVRSC=.TRUE.              MAT00810
C     SE( 7)= IDUC     0: HC=-12           1: HC .LT. -12               MAT00820
C     SE( 8)= PBOT                                                      MAT00830
C     SE( 9)= DBOT                                                      MAT00840
C     SE(10)= PTOP                                                      MAT00850
C     SE(11)= DTOP                                                      MAT00860
C     SE(12)= KEQ      UNLOADING EQUIVALENT STIFFNESS                   MAT00870
C     SE(13)= DUCT(1)                                                   MAT00880
C     SE(14)= DUCT(2)                                                   MAT00890
C     SE(15)= DUCT(3)                                                   MAT00900
C     SE(16)= EXCR(1)                                                   MAT00910
C     SE(17)= EXCR(2)                                                   MAT00920
C     SE(18)= EXCR(3)                                                   MAT00930
C     SE(19)= EXCR(4)                                                   MAT00940
C     SE(20)= EXCR(5)                                                   MAT00950
C     SE(21)= EXCR(6)                                                   MAT00960
C     SE(22)= DELY     YIELDING DISPLACEMENT                            MAT00970
C     SE(23)= EESE(1)                                                   MAT00980
C     SE(24)= EESE(2)                                                   MAT00990
C     SE(25)= EESE(2)                                                   MAT01000
C     SE(26)= EPSE                                                      MAT01010
C     SE(27)= PSEOLD                                                    MAT01020
C     SE(28)= CSE                                                       MAT01030
C     SE(29)= DMAX     ULTIMATE DISPLACEMENT                            MAT01040
C     SE(30)= D_MAX    MAXIMUM ELEMENT DISPLACEMENT FOR DAMAGE INDEX    MAT01050
C     SE(31)= P_MAX    ELEMENT LOAD CORRESPONDING TO D_MAX              MAT01060
C     SE(32)= HA       CURRENT CONTROL POINT HA                         MAT01070
C     SE(33)= VA       CURRENT CONTROL POINT VA                         MAT01080
C     SE(34)= HC       CURRENT CONTROL POINT HC                         MAT01090
C     SE(35)= VC       CURRENT CONTROL POINT VC                         MAT01100
C     SE(36)= RSN      CURRENT RESIDUAL STRAIN VALUE                    MAT01110
C     SE(37)= DEMAX    CURRENT MAXIMUM AXIAL DISPLACEMENT FOR DUCTILITIES MAT01120
C     SE(38)= REGIN    CURRENT REGINE ZONE NUMBER                       MAT01130
C     SE(39)= PREGIN PREVIOUT REGIN ZONE NUMBER                         MAT01140
C ------------------------------------------------------------          MAT01150
      SE(1)=1                                                           MAT01160
      SE(3)=1                                                           MAT01170
      SE(4)=1                                                           MAT01180
      DO 210 I=6,20                                                     MAT01190
  210   SE(I)=0                                                         MAT01200
      SE(37)=0                                                          MAT01210
      SE(38)=1                                                          MAT01220
      SE(39)=1                                                          MAT01230
C                                                                       MAT01240
      CALL HIST08(0.,0.,0.,0.,V,VL,SE(5),SE(2),                        MAT01250
     &     SMAT(1),SMAT(2),SMAT(3),SMAT(4),SMAT(5),SMAT(9),             MAT01260
     &     SMAT(6),SE(3),SE(4),SE(1),SE(6),SE(7),                       MAT01270
     &     SE(32),SE(33),SE(34),SE(35),SE(36),SE(37),SE(38),SE(39),     MAT01280
     &     SE(8),SE(9),SE(10),SE(11),BUG,                               MAT01290
     &     SE(12),SE(13),SE(16),SE(22),SE(23),SE(26),SE(27),SE(28) )    MAT01300
      SE(28)=SE(22)*SMAT(5)/2                                           MAT01310
      SE(29)=SE(22)*SMAT(7)                                             MAT01320
      SE(30)=0                                                          MAT01330
      SE(31)=0                                                          MAT01340
C                                                                       MAT01350
      RETURN                                                            MAT01360
C*------------------------ GET STIFFNESS TERMS ---------------------------MAT01370
  300 CONTINUE                                                          MAT01380
C                                                                       MAT01390
C-------- CALL HYST07 TO CALC NEW STIFFNESS STIF                        MAT01400
      CALL HYST08(P,PL,D,DL,V,VL,SE(5),SE(2),                           MAT01410
     &     SMAT(1),SMAT(2),SMAT(3),SMAT(4),SMAT(5),SMAT(9),             MAT01420
     &     SMAT(6),SE(3),SE(4),SE(1),SE(6),SE(7),                       MAT01430
     &     SE(32),SE(33),SE(34),SE(35),SE(36),SE(37),SE(38),SE(39),     MAT01440
     &     SE(8),SE(9),SE(10),SE(11),BUG,                               MAT01450
     &     SE(12),SE(13),SE(16),SE(22),SE(23),SE(26),SE(27),SE(28) )    MAT01460
C                                                                       MAT01470
C --- STORE MAXIMUM DISPLACEMENT AND CORRESPONDING LOAD                 MAT01480
      IF(ABS(D) .GT. ABS(SE(30))) THEN                                  MAT01490
              SE(30)=D                                                  MAT01500
              SE(31)=P                                                  MAT01510
      ENDIF                                                             MAT01520
C                                                                       MAT01530
C=== GET ENERGIES, DUCTILITY AND EXCURSION FOR BOTH IOPT=3 AND 4 ====   MAT01540
  400 CONTINUE                                                          MAT01550
C                                                                       MAT01560
C --- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                          MAT01570
      EESE=SE(23)                                                       MAT01580
      EPSE=SE(26)                                                       MAT01590
C                                                                       MAT01600
C --- TRANSFER DUCTILITIES AND EXCURSION RATIOS FOR BOTH IOPT=3 AND 4.  MAT01610
      DO 410 I=1,3                                                      MAT01620
        DUCT(I)=SE(12+I)                                                MAT01630
        EXCR(I+3)=SE(15+I+3)                                            MAT01640
  410   EXCR(I)=SE(15+I)                                                MAT01650
C                                                                       MAT01660
      RETURN                                                            MAT01670
C*------------------ DAMAGE INDEX ----------------------------------    MAT01680
  500 CONTINUE                                                          MAT01690
      EESE=SE(23)                                                       MAT01700
      EPSE=SE(26)                                                       MAT01710
      DMAX=SE(29)                                                       MAT01720
      D=SE(30)                                                          MAT01730
      P=SE(31)                                                          MAT01740
      PY=SMAT(5)                                                        MAT01750
      BDAM=SMAT(8)                                                      MAT01760
      DAMAGE=ABS(D)/DMAX + BDAM / (PY*DMAX) * EPSE                      MAT01770
C                                                                       MAT01780
      RETURN                                                            MAT01790
      END                                                               MAT01800
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP                                  MAT00010
C*---------------------------------------------------------------------MAT00020
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    MAT00030
C           END DEBUG                                                   MAT00040
C*---------------------------------------------------------------------MAT00050
      SUBROUTINE MAT09                                                  MAT00060
     &  (IOPT    ,IELNO    ,ERR ,FIRST  ,PRINT ,BUG,                    MAT00070
     &   MAT     ,SE   ,P,PL,D,DL,V,VL,LELEM,LMAT,                      MAT00080
     &   SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)                              MAT00090
C                                                                       MAT00100
      IMPLICIT REAL(A-H,O-Z)                                            MAT00110
      LOGICAL ERR,FIRST,PRINT,BUG                                       MAT00120
      LOGICAL BTEST                                                     MAT00130
      DIMENSION SE(100),SMAT(4),DUCT(3),EXCR(6)                         MAT00140
      CHARACTER*80 TYPE                                                 MAT00150
C                                                                       MAT00160
C------- RESERVE STORAGE FOR ELEMENT                                    MAT00170
      LELEM=66                                                          MAT00180
      IF (IOPT.EQ.0) RETURN                                             MAT00190
C                                                                       MAT00200
C---------------------------------------------------------------------- MAT00210
C VARIABLES:                                                            MAT00220
C---------------------------------------------------------------------- MAT00230
C---------- GLOBAL VARIABLES                                           MAT00240
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT00250
C        = 2, INITIALIZE STIFFNESS                                      MAT00260
C        = 3, GET STIFFNESS TERM                                        MAT00270
C        = 4, GET ENERGIES, DUCTILITY AND EXCURSION                     MAT00280
C        = 5, CALCULATE DAMAGE INDEX                                    MAT00290
C RINPUT = INPUT DATA                                                   MAT00300
C SMAT   = INTERNAL STORAGE                                             MAT00310
C SE     = OUTPUT STIFFNESS                                             MAT00320
C---------------------------------------------------------------------- MAT00330
      GO TO (100,200,300,400,500),IOPT                                  MAT00340
C                                                                       MAT00350
  100 CONTINUE                                                          MAT00360
C                                                                       MAT00370
C----- INTERPERTATE INPUT DATA                                          MAT00380
C     HA    = SMAT( 1)                                                  MAT00390
C     VA    = SMAT( 2)                                                  MAT00400
C     DUCMAX= SMAT( 3)    : FAILURE DUCTILITY                         MAT00410
C     BDAM  = SMAT( 4)    : BATA PARAMETER                            MAT00420
C                                                                       MAT00430
      READ (5,*) TYPE ,(SMAT(I),I=1,2), DUCMAX, BDAM                    MAT00440
C                                                                       MAT00450
      LMAT = 4                                                          MAT00460
      SMAT(3)=DUCMAX                                                    MAT00470
      SMAT(4)=BDAM                                                      MAT00480
C                                                                       MAT00490
      IF (FIRST .OR. BUG) WRITE (6,191)                                 MAT00500
      WRITE (6,192) MAT,(SMAT(K),K=1,2),DUCMAX,BDAM                     MAT00510
C                                                                       MAT00520
 191  FORMAT (////' PINO-SUAREZ TOWER LONG DIR. HYST. MODEL PARAMETERS'/ MAT00530
     &        '     ------------------------------------------------'// MAT00540
     &  '  MAT.',3X,'    HA',6X,'    VA',                             MAT00550
     &       6X,'MAX DUC', 6X,'  BETA')                                 MAT00560
 192  FORMAT (1X,I3,2X,6G12.6/)                                         MAT00570
C                                                                       MAT00580
      RETURN                                                            MAT00590
C                                                                       MAT00600
C*------------------------ INITIALIZE STIFFNESS TERMS --------------------MAT00610
  200 CONTINUE                                                          MAT00620
C     ---- PINO-SUAREZ TOWER LONG GIRDER MODEL DATA STORAGE FOR MEMBER  MAT00630
C     SE( 1)= STIF   MEMBER MOMENT-ROTATION STIFFNESS                   MAT00640
C     SE( 2)= DP     LOAD INCREMENT                                     MAT00650
C     SE( 3)= DX     DISPLACEMENT INCREMENT                             MAT00660
C     SE( 4)= RV     INDEX FOR TOP OR BOTTOM ENVELOPE                   MAT00670
C                    RV=0 THEN HIGH=.TRUE.                              MAT00680
C                    RV=1 THEN HIGH=.FALSE.                             MAT00690
C     SE( 5)= RULE   RULE NUMBER                                        MAT00700
C     SE( 6)= PTOP                                                      MAT00710
C     SE( 7)= DTOP                                                      MAT00720
C     SE( 8)= PBOT                                                      MAT00730
C     SE( 9)= DBOT                                                      MAT00740
C     SE(10)= SPTOP                                                     MAT00750
C     SE(11)= SDTOP                                                     MAT00760
C     SE(12)= SPBOT                                                     MAT00770
C     SE(13)= SDBOT                                                     MAT00780
C     SE(14)= SSPTOP                                                    MAT00790
C     SE(15)= SSDTOP                                                    MAT00800
C     SE(16)= SSPBOT                                                    MAT00810
C     SE(17)= SSDBOT                                                    MAT00820
C     SE(18)= HB                                                        MAT00830
C     SE(19)= VB                                                        MAT00840
C     SE(20)= HD                                                        MAT00850
C     SE(21)= VD                                                        MAT00860
C     SE(22)= HE                                                        MAT00870
C     SE(23)= VE                                                        MAT00880
C     SE(24)= HAP                                                       MAT00890
C     SE(25)= VAP                                                       MAT00900
```

```
C     SE(26)= HBP                                                       MAT00910
C     SE(27)= VBP                                                       MAT00920
C     SE(28)= SEHD                                                      MAT00930
C     SE(29)= SEVD                                                      MAT00940
C     SE(30)= SEHE                                                      MAT00950
C     SE(31)= SEVE                                                      MAT00960
C     SE(32)= SSSPTP                                                    MAT00970
C     SE(33)= SSSCTP                                                    MAT00980
C     SE(34)= SSSPBT                                                    MAT00990
C     SE(35)= SSSCBT                                                    MAT01000
C     SE(36)= SHD                                                       MAT01010
C     SE(37)= SVD                                                       MAT01020
C     SE(38)= SHE                                                       MAT01030
C     SE(39)= SVE                                                       MAT01040
C     SE(40)= KEQ     UNLOADING EQUIVALENT STIFFNESS                    MAT01050
C     SE(41)= DUCT(1)                                                   MAT01060
C     SE(42)= DUCT(2)                                                   MAT01070
C     SE(43)= DUCT(3)                                                   MAT01080
C     SE(44)= EXCR(1)                                                   MAT01090
C     SE(45)= EXCR(2)                                                   MAT01100
C     SE(46)= EXCR(3)                                                   MAT01110
C     SE(47)= EXCR(4)                                                   MAT01120
C     SE(48)= EXCR(5)                                                   MAT01130
C     SE(49)= EXCR(6)                                                   MAT01140
C     SE(50)= DELY    YIELDING DISPLACEMENT                             MAT01150
C     SE(51)= EESE(1)                                                   MAT01160
C     SE(52)= EESE(2)                                                   MAT01170
C     SE(53)= EESE(2)                                                   MAT01180
C     SE(54)= EPSE                                                      MAT01190
C     SE(55)= PSEOLD                                                    MAT01200
C     SE(56)= CSE                                                       MAT01210
C     SE(57)= DMAX    ULTIMATE DISPLACEMENT                             MAT01220
C     SE(58)= D_MAX  MAXIMUM ELEMENT DISPLACEMENT FOR DAMAGE INDEX      MAT01230
C     SE(59)= P_MAX  ELEMENT LOAD CORRESPONDING TO D_MAX                MAT01240
C     SE(60)= DEMAX  CURRENT MAXIMUM AXIAL DISPLACEMENT FOR DUCTILITIES MAT01250
C     SE(61)= INITIAL STIFFNESS                                         MAT01260
C     SE(62)= DISP FACTOR FACH                                          MAT01270
C     SE(63)= MOMENT FACTOR FACV                                        MAT01280
C     SE(64)= ROTATION STIFFNESS FACTOR=FACV/FACH                       MAT01290
C     SE(65)= RA                                                        MAT01300
C     SE(66)= VA                                                        MAT01310
C ------------------------------------------------------------------    MAT01320
      DO 210 I=1,61                                                     MAT01330
  210   SE(I)=0                                                         MAT01340
      SE(50)=SMAT(1)                                                    MAT01350
      SE(5)=1                                                           MAT01360
      SE(65)= SMAT(1)                                                   MAT01370
      SE(66)= SMAT(2)                                                   MAT01380
C                                                                       MAT01390
      MOPT=IOPT                                                         MAT01400
      CALL HYST09(MOPT,SE(1),0,0,0,0,SE(2),SE(3),                      MAT01410
     &     SE(4),SE(5),SE(6),SE(7),SE(8),SE(9),                        MAT01420
     &     SE(10),SE(11),SE(12),SE(13),                                MAT01430
     &     SE(14),SE(15),SE(16),SE(17),                                MAT01440
     &     SE(65),SE(66),SE(18),SE(19),SE(20),SE(21),SE(22),SE(23),    MAT01450
     &     SE(24),SE(25),SE(26),SE(27),                                MAT01460
     &     SE(28),SE(29),SE(30),SE(31),                                MAT01470
     &     SE(32),SE(33),SE(34),SE(35),                                MAT01480
     &     SE(36),SE(37),SE(38),SE(39),BUG,                            MAT01490
     &     SE(40),SE(41),SE(44),SE(50),SE(51),SE(54),SE(55),SE(56),    MAT01500
     &     SE(60),SE(62),SE(63),SE(64),SMAT(3) )                       MAT01510
      SE(56)=SE(50)*SMAT(2)/2                                           MAT01520
      SE(57)=SE(22)*SMAT(3)                                             MAT01530
      SE(58)=0                                                          MAT01540
      SE(59)=0                                                          MAT01550
      SE(61)=SE(1)                                                      MAT01560
C                                                                       MAT01570
      RETURN                                                            MAT01580
C------------------------ GET STIFFNESS TERMS ------------------------  MAT01590
  300 CONTINUE                                                          MAT01600
      SE(2)=P-PL                                                        MAT01610
      SE(3)=D-DL                                                        MAT01620
C                                                                       MAT01630
C-- CALL HYST09 TO CALC NEW STIFFNESS STIF                             MAT01640
      MOPT=IOPT                                                         MAT01650
      CALL HYST09(MOPT,SE(1),P,D,PL,DL,SE(2),SE(3),                    MAT01660
     &     SE(4),SE(5),SE(6),SE(7),SE(8),SE(9),                        MAT01670
     &     SE(10),SE(11),SE(12),SE(13),                                MAT01680
     &     SE(14),SE(15),SE(16),SE(17),                                MAT01690
     &     SE(65),SE(66),SE(18),SE(19),SE(20),SE(21),SE(22),SE(23),    MAT01700
     &     SE(24),SE(25),SE(26),SE(27),                                MAT01710
     &     SE(28),SE(29),SE(30),SE(31),                                MAT01720
     &     SE(32),SE(33),SE(34),SE(35),                                MAT01730
     &     SE(36),SE(37),SE(38),SE(39),BUG,                            MAT01740
     &     SE(40),SE(41),SE(44),SE(50),SE(51),SE(54),SE(55),SE(56),    MAT01750
     &     SE(60),SE(62),SE(63),SE(64),SMAT(3) )                       MAT01760
C                                                                       MAT01770
C --- STORE MAXIMUM DISPLACEMENT AND CORRESPONDING LOAD                MAT01780
      IF(ABS(D) .GT. ABS(SE(58))) THEN                                  MAT01790
              SE(58)=D                                                  MAT01800
              SE(59)=P                                                  MAT01810
      ENDIF                                                             MAT01820
C                                                                       MAT01830
C--- GET ENERGIES, DUCTILITY AND EXCURSION FOR BOTH IOPT=3 AND 4 ----   MAT01840
  400 CONTINUE                                                          MAT01850
C                                                                       MAT01860
C --- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                          MAT01870
      EESE=SE(51)                                                       MAT01880
      EPSE=SE(54)                                                       MAT01890
C                                                                       MAT01900
C --- TRANSFER DUCTILITIES AND EXCURSION RATIOS FOR BOTH IOPT=3 AND 4.  MAT01910
      DO 410 I=1,3                                                      MAT01920
        DUCT(I)=SE(40+I)                                                MAT01930
        EXCR(I+3)=SE(43+I+3)                                            MAT01940
  410   EXCR(I)=SE(43+I)                                                MAT01950
C                                                                       MAT01960
      RETURN                                                            MAT01970
C----------------- DAMAGE INDEX ------------------------------------    MAT01980
  500 CONTINUE                                                          MAT01990
C...DAMAGE INDEX IS NOT AVAILABLE                                       MAT02000
C     EESE=SE(51)                                                       MAT02010
C     EPSE=SE(54)                                                       MAT02020
C     DMAX=SE(57)                                                       MAT02030
C     D=SE(58)                                                          MAT02040
C     P=SE(59)                                                          MAT02050
C     PY=SMAT(2)                                                        MAT02060
C     BDAM=SMAT(4)                                                      MAT02070
C     DAMAGE=ABS(D)/DMAX + BDAM / (PY*DMAX) * EPSE                      MAT02080
      DAMAGE=0                                                          MAT02090
C                                                                       MAT02100
      RETURN                                                            MAT02110
      END                                                               MAT02120
C@PROCESS SDUMP GOSTMT XREF MAP                                         MAT00010
C**********************************************************************MAT00020
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    MAT00030
C           END DEBUG                                                   MAT00040
C**********************************************************************MAT00050
      SUBROUTINE MAT10                                                  MAT00060
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG,                      MAT00070
     &   MAT    ,SE   ,P,PP,D,DL,V,VL,LELEM,LMAT,                       MAT00080
     &   SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)                              MAT00090
C                                                                       MAT00100
      IMPLICIT REAL(A-H,O-Z)                                            MAT00110
      LOGICAL ERR,FIRST,PRINT,BUG                                       MAT00120
      LOGICAL BTEST                                                     MAT00130
      DIMENSION SE(100),SMAT(9),DUCT(3),EXCR(6)                         MAT00140
      CHARACTER*80 TYPE                                                 MAT00150
C                                                                       MAT00160
C------- RESERVE STORAGE FOR ELEMENT                                    MAT00170
      LELEM=15                                                          MAT00180
      IF (IOPT.EQ.0) RETURN                                             MAT00190
C                                                                       MAT00200
C----------------------------------------------------------------------MAT00210
C VARIABLES:                                                            MAT00220
C----------------------------------------------------------------------MAT00230
C---------- GLOBAL VARIABLES                                           MAT00240
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT00250
C        = 2, INITIALIZE STIFFNESS                                      MAT00260
C        = 3, GET STIFFNESS TERM                                        MAT00270
C        = 4, GET ENERGIES, DUCTILITY AND EXCURSION                     MAT00280
C        = 5, CALCULATE DAMAGE INDEX                                    MAT00290
C RINPUT = INPUT DATA                                                   MAT00300
C SMAT   = INTERNAL STORAGE                                             MAT00310
C SE     = OUTPUT STIFFNESS                                             MAT00320
C----------------------------------------------------------------------MAT00330
      GO TO (100,200,300,400,500),IOPT                                  MAT00340
C                                                                       MAT00350
  100 CONTINUE                                                          MAT00360
C                                                                       MAT00370
C----- INTERPERTATE INPUT DATA                                          MAT00380
C     ELAS  = SMAT( 1)   : 0 FOR ELASTIC ANALYSIS                       MAT00390
C                        : 1 FOR INELASTIC ANALYSIS                     MAT00400
C                        : 2 FOR SIMULATING PINO-SUAREZ TOWER COLUMN    MAT00410
C                          LOCAL BUCKLING                               MAT00420
C                        : 3 FOR INELASTIC ANALYSIS AND CHECK BOX       MAT00430
C                          COLUMN INTERACTIVE STRENGH CRITERIA          MAT00440
C     SP    = SMAT( 2)   : RATIO FOR INELASTIC STIFFNESS TO INITIAL     MAT00450
C                          STIFFNESS                                    MAT00460
C     E     = SMAT( 3)   : ELASTIC MODULUS                              MAT00470
C     TI    = SMAT( 4)   : SECTION MOMENT OF INERTIA                    MAT00480
C     TMP   = SMAT( 5)   : PLASTIC MOMENT CAPACITY                      MAT00490
C     DUCMAX= SMAT( 6)   : FAILURE DUCTILITY                            MAT00500
C                          .GE. 1 FAILURE DUCTILITY CONSIDERED          MAT00510
C                          .LT. 1 FAILURE DUCTILITY NOT CONSIDERED      MAT00520
C     BDAM  = SMAT( 7)   : PARAMETER FOR ANG'S DAMAGE INDEX             MAT00530
C     REDUCE= SMAT( 8)   : RESIDUAL STRENGTH FOR ELAS=2                 MAT00540
C           = SMAT( 9)   : DUMMY VARIABLE                               MAT00550
C**********************************************************************MAT00560
C                                                                       MAT00570
      READ (5,*) TYPE ,(SMAT(I),I=1,5),DUCMAX,BDAM                      MAT00580
      SMAT(6)=DUCMAX                                                    MAT00590
      SMAT(7)=BDAM                                                      MAT00600
C                                                                       MAT00610
      IF( SMAT(1) .EQ. 2 ) THEN                                         MAT00620
         READ(5,*)REDUCE                                                MAT00630
         SMAT(8)=REDUCE                                                 MAT00640
      ENDIF                                                             MAT00650
C                                                                       MAT00660
      LMAT = 9                                                          MAT00670
C                                                                       MAT00680
      IF (FIRST .OR. BUG) WRITE (6,191)                                 MAT00690
      WRITE (6,192) MAT,(SMAT(K),K=1,5),SMAT(6),SMAT(7),REDUCE          MAT00700
C                                                                       MAT00710
 191  FORMAT (///' BILINEAR INTERACTIVE MATERIAL PROPERTIES'/           MAT00720
     &        ' ================================='//                    MAT00730
     &  '  MAT.',3X,' ELAS',6X,'  SP',                                  MAT00740
     &          6X,'   E ',6X,'   TI',6X,'   MP',                      MAT00750
     &          5X,'MAX DUC', 6X,'  BETA',6X,'REDU F',/)                MAT00760
 192  FORMAT (1X,I5,8G12.6/)                                            MAT00770
C                                                                       MAT00780
      RETURN                                                            MAT00790
C                                                                       MAT00800
C------------------------ INITIALIZE STIFFNESS TERMS ------------------MAT00810
  200 CONTINUE                                                          MAT00820
C     ---- BILINEAR INTERACTIVE MATERIAL DATA STORAGE FOR MEMBER ----   MAT00830
C     SE( 1)= EI      SECTION RIGIDITY                                  MAT00840
C     SE( 2)= P1      CURRENT YIELDING LOAD FOR PASITIVE LOAD           MAT00850
C     SE( 3)= P2      CURRENT YIELDING LOAD FOR NEGATIVE LOAD           MAT00860
C     SE( 4)= YIELD  0: ELASTIC,  1: INELASTIC                          MAT00870
C     SE( 5)= ELAS                                                      MAT00880
C     SE( 6)= SP                                                        MAT00890
C     SE( 7)= FAILURE DUCTILITY                                         MAT00900
C     SE( 8)= PARAMETER FOR ANG'S DAMAGE INDEX                          MAT00910
C     SE( 9)= RESIDUAL STRENGTH FOR ELAS=2                              MAT00920
C     SE(10)= CURRENT DUCTILITY                                         MAT00930
C     SE(11)= SMAT(4)                                                   MAT00940
C     SE(12)= TMP=SMAT(5)                                               MAT00950
C     SE(13)= DELY  YIELDING DISPLACEMENT                               MAT00960
C             FOR ROTATION SE(13)=TMP*LENGTH/(6EI)                      MAT00970
C             FOR AXIAL    SE(13)=TMP*LENGTH/EI                         MAT00980
C             FOR TORSION  SE(13)=TMP*LENGTH/EI                         MAT00990
C     SE(14)= DEMAX CURRENT MAXIMUM DISPLACEMENT FOR DUCTILITY          MAT01000
C     SE(15)= YIELDING DISPLACEMENT INDEX:                              MAT01010
C             0: DELY HAS NOT BEEN DETERMINED                           MAT01020
C             1: DELY HAS BEEN DETERMINED                               MAT01030
C ------------------------------------------------------------          MAT01040
      DO 210 I=1,15                                                     MAT01050
  210   SE(I)=0                                                         MAT01060
      SE(1)= SMAT(3)*SMAT(4)                                            MAT01070
      SE(2)= SMAT(5)                                                    MAT01080
      SE(3)= -SMAT(5)                                                   MAT01090
      SE(4)= 0                                                          MAT01100
      SE(5)= SMAT(1)                                                    MAT01110
      SE(6)= SMAT(2)                                                    MAT01120
      SE(7)= SMAT(6)                                                    MAT01130
      SE(8)= SMAT(7)                                                    MAT01140
      SE(9)= SMAT(8)                                                    MAT01150
      SE(11)= SMAT(4)                                                   MAT01160
      SE(12)= SMAT(5)                                                   MAT01170
C                                                                       MAT01180
C                                                                       MAT01190
      RETURN                                                            MAT01200
C------------------------ GET STIFFNESS TERMS ------------------------  MAT01210
  300 CONTINUE                                                          MAT01220
C                                                                       MAT01230
C..CHECK FAILURE DUCTILITY                                              MAT01240
      DUCT(1)=SE(10)                                                    MAT01250
      IF( DUCT(1) .GE. SMAT(6) .AND. SMAT(6) .GE. 1) THEN               MAT01260
        P=0                                                             MAT01270
        YIELD=1                                                         MAT01280
        SE(4)=YIELD                                                     MAT01290
        RETURN                                                          MAT01300
      ENDIF                                                             MAT01310
C                                                                       MAT01320
      ELAS  = SMAT( 1)                                                  MAT01330
      SP    = SMAT( 2)                                                  MAT01340
      E     = SMAT( 3)                                                  MAT01350
      TI    = SMAT( 4)                                                  MAT01360
      TMP   = SMAT( 5)                                                  MAT01370
      P1    = SE(2)                                                     MAT01380
      P2    = SE(3)                                                     MAT01390
      YIELD = SE(4)                                                     MAT01400
C                                                                       MAT01410
      IF( ELAS .EQ. 0 ) THEN                                            MAT01420
         YIELD = 0                                                      MAT01430
         SE(4)=YIELD                                                    MAT01440
         SE(14)=0                                                       MAT01450
         DUCT(1)=0                                                      MAT01460
         RETURN                                                         MAT01470
      ENDIF                                                             MAT01480
C     PPPP=P                                                            MAT01490
C                                                                       MAT01500
      IF( ELAS .EQ. 2 ) THEN                                            MAT01510
C ** FOR ELASTIC REGION                                                 MAT01520
         IF( YIELD .EQ. 0 ) THEN                                        MAT01530
            IF( P .GE. P1 .OR. P .LE. P2 ) THEN                         MAT01540
               YIELD=1                                                  MAT01550
               IF( P .GE. P1 ) P=P1                                     MAT01560
               IF( P .LE. P2 ) P=P2                                     MAT01570
C ------------ DETERMINE ELEMENT DUCTILITY UNIT                        MAT01580
C             NOTE: ROTATION DUCTILITY UNIT IS DEFINED WHEN             MAT01590
C             SE(15)=1, IF (P .GE. TMP .AND. D .GE. TMP*LENGTH/(6EI))   MAT01600
C             ELSE SE(15)=0                                             MAT01610
C                                                                       MAT01620
```

```
              IF( SE(15) .EQ. 0 ) THEN
                  IF( ABS(D) .GE. ABS(SE(13)) ) THEN
C                   SE(13)=ABS(D)
                    SE(15)=1
                  ENDIF
              ENDIF
            ELSE IF( P .LT. P1 .AND. P .GT. P2 ) THEN
              YIELD=0
            ENDIF

C ** FOR INELASTIC REGION
          ELSE IF( YIELD .EQ. 1 ) THEN
            IF( P .GE. P1 .OR. P .LE. P2 ) THEN
              YIELD=1
              IF( P .GE. P1 ) P=P1
              IF( P .LE. P2 ) P=P2
C ------------ DETERMINE ELEMENT DUCTILITY UNIT
C             NOTE: ROTATION DUCTILITY UNIT IS DEFINED WHEN
C             SE(15)=1, IF (P .GE. TMP .AND. D .GE. TMP*LENGTH/(6EI))
C             ELSE SE(15)=0
C
              IF( SE(15) .EQ. 0 ) THEN
                  IF( ABS(D) .GE. ABS(SE(13)) ) THEN
C                   SE(13)=ABS(D)
                    SE(15)=1
                  ENDIF
              ENDIF
            ELSE IF( P .LT. P1 .AND. PP .GE. 0 ) THEN
              YIELD=0
C             *** UNBALANCED FORCE ***
              IF( (D-DL) .NE. 0 ) THEN
                SM2=(P-PP)/(D-DL)
                SM1=SM2/SP
                UD=D-DL
                UP=SM1*UD
                P=PP+UP
                IF( P .LE. P2 ) THEN
                    P=P2
                    YIELD=1
                ENDIF
              ENDIF
            ELSE IF( P .GT. P2 .AND. PP .LE. 0 ) THEN
              YIELD=0
C             *** UNBALANCED FORCE ***
              IF( (D-DL) .NE. 0 ) THEN
                SM2=(P-PP)/(D-DL)
                SM1=SM2/SP
                UD=D-DL
                UP=SM1*UD
                P=PP+UP
                IF( P .GE. P1 ) THEN
                    P=P1
                    YIELD=1
                ENDIF
              ENDIF
            ENDIF
          ENDIF
C
          GO TO 399
        ENDIF
C
C...  FOR ELAS=1 AND 3 CASES:
C --- ELASTIC REGION
        IF(YIELD .EQ. 0 ) THEN
          IF( P .GE. P1 .OR. P .LE. P2 ) THEN
            YIELD=1
C           *** UNBALANCED FORCE ***
            IF( P .GE. P1 ) THEN
              IF( (D-DL) .NE. 0 .AND. (P-PP) .NE. 0) THEN
                SM1=(P-PP)/(D-DL)
C               /* SM2 IS ESTIMATED VALUE */
                SM2=SM1*SP
                UP1=P-P1
                UD=UP1/SM1
                UP2=-UD*SM2
                P=P-(UP1+UP2)
              ENDIF
C             IF( D .EQ. 0 .AND. DL .EQ. 0 ) P=P
C             P=P1
            ELSE IF( P .LE. P2 ) THEN
              IF( (D-DL) .NE. 0 .AND. (P-PP) .NE. 0) THEN
                SM1=(P-PP)/(D-DL)
C               /* SM2 IS ESTIMATED VALUE */
                SM2=SM1*SP
                UP1=P-P2
                UD=UP1/SM1
                UP2=-UD*SM2
                P=P-(UP1+UP2)
              ENDIF
C             IF( D .EQ. 0 .AND. DL .EQ. 0 ) P=P2
C             P=P2
            ENDIF
C --------- DETERMINE ELEMENT DUCTILITY UNIT
C           NOTE: ROTATION DUCTILITY UNIT IS DEFINED WHEN
C           SE(15)=1, IF (P .GE. TMP .AND. D .GE. TMP*LENGTH/(6EI))
C           ELSE SE(15)=0
C
            IF( SE(15) .EQ. 0 ) THEN
                IF( ABS(D) .GE. ABS(SE(13)) ) THEN
C                 SE(13)=ABS(D)
                  SE(15)=1
                ENDIF
            ENDIF
C
C
          ELSE
            YIELD=0
          ENDIF
          GO TO 399
        ENDIF
C
C --- INELASTIC REGION
        IF(YIELD .EQ. 1 ) THEN
          IF( P .GT. P1 ) THEN
              IF( P.GE. PP) THEN
                  YIELD=1
C
                  IF( SE(15) .EQ. 0 ) THEN
                      IF( ABS(D) .GE. ABS(SE(13)) ) THEN
                        SE(13)=ABS(D)
                        SE(15)=1
                      ENDIF
                  ENDIF
              ELSE IF( P.LT. PP) THEN
                  YIELD=0
                  P1=PP
                  P2=PP-2*TMP
C                 *** UNBALANCED FORCE ***
                  IF( (D-DL) .NE. 0 ) THEN
                    SM2=(P-PP)/(D-DL)
                    SM1=SM2/SP
                    UD=D-DL
                    UP=SM1*UD
                    P=PP+UP
                  ENDIF
              ENDIF
C
          ELSE IF( P .LT. P2 ) THEN
              IF( P.LE. PP) THEN
                  YIELD=1
C
                  IF( SE(15) .EQ. 0 ) THEN
                      IF( ABS(D) .GE. ABS(SE(13)) ) THEN
                        SE(13)=ABS(D)
                        SE(15)=1
                      ENDIF
                  ENDIF
              ELSE IF(P.GT. PP) THEN
                  YIELD=0
                  P2=PP
                  P1=PP+2*TMP
C                 *** UNBALANCED FORCE ***
                  IF( (D-DL) .NE. 0 ) THEN
                    SM2=(P-PP)/(D-DL)
                    SM1=SM2/SP
                    UD=D-DL
                    UP=SM1*UD
                    P=PP+UP
                  ENDIF
              ENDIF
C
          ENDIF
        ENDIF
C
  399 SE(2)=P1
      SE(3)=P2
      SE(4)=YIELD
      IF( SE(15) .EQ. 1 ) THEN
          IF( SE(14) .LE. ABS(D)) THEN
              SE(14)=ABS(D)
              DUCT(1)=SE(14)/SE(13)
              SE(10)=DUCT(1)
          ENDIF
      ELSE IF( SE(15) .EQ. 0 ) THEN
          DUCT(1)=0
          SE(10)=DUCT(1)
      ENDIF
C
C     IF( PPPP .NE. P ) THEN
C       DDDD=P-PPPP
C     WRITE(62,879) P,PPPP,DDDD
C 879 FORMAT(1X,4G15.6)
C     ENDIF
C
C============== GET DUCTILITY IOPT=3 AND 4 ==========================
  400 CONTINUE
C
C --- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.
      EESE=0
      EPSE=0
      ELAS = SMAT(1)
C
C --- TRANSFER DUCTILITIES AND EXCURSION RATIOS FOR BOTH IOPT=3 AND 4.
C     DO 410 I=1,3
C       DUCT(I)=0
C       EXCR(I+3)=0
C 410   EXCR(I)=0
C
      IF( ELAS .EQ. 0 ) THEN
          DUCT(1)=0
          SE(10)=DUCT(1)
      ELSE IF( ELAS .NE. 0 ) THEN
              IF( SE(15) .EQ. 0 ) THEN
                  DUCT(1)=0
                  SE(10)=DUCT(1)
              ELSE IF( SE(15) .EQ. 1 ) THEN
                  DUCT(1)=SE(14)/SE(13)
                  SE(10)=DUCT(1)
              ENDIF
      ENDIF
C
C     WRITE(40,777)SE(13),SE(14),DUCT(1),SE(15)
C 777 FORMAT(1X,4G15.6)
      RETURN
C================== DAMAGE INDEX ====================================
  500 CONTINUE
C..DAMAGE INDEX IS NOT AVAILABLE
      DAMAGE=0
C
      RETURN
      END
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP
C*********************************************************************
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE
C           END DEBUG
C*********************************************************************
      SUBROUTINE MAT11
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG,
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,
     &   SMAT,ESSE,EPSE,DUCT,EXCR,DAMAGE)
C
      IMPLICIT REAL(A-H,O-Z)
      LOGICAL ERR,FIRST,PRINT,BUG
      LOGICAL BTEST
      DIMENSION SE(100),SMAT(5),DUCT(3),EXCR(6)
      CHARACTER*80 TYPE
C
C------- RESERVE STORAGE FOR ELEMENT
      LELEM=31
      IF (IOPT.EQ.0) RETURN
C
C*********************************************************************
C VARIABLES:
C*********************************************************************
C---------- GLOBAL VARIABLES
C IOPT   = 1, INITIALIZE MATERIAL
C        = 2, INITIALIZE STIFFNESS
C        = 3, GET STIFFNESS TERM
C        = 4, GET ENERGIES, DUCTILITY AND EXCURSION
C        = 5, CALCULATE DAMAGE INDEX
C RINPUT = INPUT DATA
C SMAT   = INTERNAL STORAGE
C SE     = OUTPUT STIFFNESS
C*********************************************************************
      GO TO (100,200,300,400,500),IOPT
C
  100 CONTINUE
C
C-----  INTERPERTATE INPUT DATA
C     HA    = SMAT( 1)
C     VA    = SMAT( 2)
C     RATIO = SMAT( 3)   : STRAIN HARTEN RATIO TO ELASTIC STIFFNESS
C     DUCMAX= SMAT( 4)   : FAILURE DUCTILITY
C     BDAM  = SMAT( 5)   : BATA PARAMETER
C
      READ (5,*) TYPE ,(SMAT(I),I=1,3), DUCMAX, BDAM
C
      LMAT = 5
      SMAT(4)=DUCMAX
      SMAT(5)=BDAM
C
      IF (FIRST .OR. BUG) WRITE (6,191)
      WRITE (6,192) MAT,(SMAT(K),K=1,3),DUCMAX,BDAM
C
  191  FORMAT (////' PINO-SUAREZ TOWER SHORT DIR. HYST. MODEL PARAMETERS'/
     &'  ================================================='//
     &'  MAT.',3X,'   HA',6X,'   VA',6X,' RATIO',
     &'          6X,'MAX DUC', 6X,'  BETA')
  192  FORMAT (1X,I3,2X,6G12.6/)
C
      RETURN
C
C========================= INITIALIZE STIFFNESS TERMS ================
```

```
  200 CONTINUE                                                          MAT00630
C     ---- PINO-SUAREZ TOWER SHORT GIRDER MODEL DATA STORAGE FOR MEMBER MAT00640
C     SE( 1)= STIF   MEMBER MOMENT-ROTATION STIFFNESS                  MAT00650
C     SE( 2)= DP     LOAD INCREMENT                                    MAT00660
C     SE( 3)= DX     DISPLACEMENT INCREMENT                            MAT00670
C     SE( 4)= RV     INDEX FOR TOP OR BOTTOM ENVELOPE                  MAT00680
C                    RV=0 THEN HIGH=.TRUE.                             MAT00690
C                    RV=1 THEN HIGH=.FALSE.                            MAT00700
C     SE( 5)= RULE   RULE NUMBER                                       MAT00710
C     SE( 6)= PTOP                                                     MAT00720
C     SE( 7)= DTOP                                                     MAT00730
C     SE( 8)= PBOT                                                     MAT00740
C     SE( 9)= DBOT                                                     MAT00750
C     SE(10)= KEQ      UNLOADING EQUIVALENT STIFFNESS                  MAT00760
C     SE(11)= DUCT(1)                                                  MAT00770
C     SE(12)= DUCT(2)                                                  MAT00780
C     SE(13)= DUCT(3)                                                  MAT00790
C     SE(14)= EXCR(1)                                                  MAT00800
C     SE(15)= EXCR(2)                                                  MAT00810
C     SE(16)= EXCR(3)                                                  MAT00820
C     SE(17)= EXCR(4)                                                  MAT00830
C     SE(18)= EXCR(5)                                                  MAT00840
C     SE(19)= EXCR(6)                                                  MAT00850
C     SE(20)= DELY     YIELDING DISPLACEMENT                           MAT00860
C     SE(21)= EESE(1)                                                  MAT00870
C     SE(22)= EESE(2)                                                  MAT00880
C     SE(23)= EESE(2)                                                  MAT00890
C     SE(24)= EPSE                                                     MAT00900
C     SE(25)= PSEOLD                                                   MAT00910
C     SE(26)= CSE                                                      MAT00920
C     SE(27)= DMAX   ULTIMATE DISPLACEMENT                             MAT00930
C     SE(28)= D_MAX  MAXIMUM ELEMENT DISPLACEMENT FOR DAMAGE INDEX     MAT00940
C     SE(29)= P_MAX  ELEMENT LOAD CORRESPONDING TO D_MAX               MAT00950
C     SE(30)= DEMAX  CURRENT MAXIMUM AXIAL DISPLACEMENT FOR DUCTILITIES MAT00960
C     SE(31)= INITIAL STIFFNESS                                        MAT00970
C ---------------------------------------------------------------      MAT00980
      DO 210 I=1,31                                                    MAT00990
  210   SE(I)=0                                                        MAT01000
      SE(20)=SMAT(1)                                                   MAT01010
      SE(5)=1                                                          MAT01020
C                                                                      MAT01030
      MOPT=IOPT                                                        MAT01040
      CALL HYST11(MOPT,SE(1),0,0,0,0,SE(2),SE(3),                     MAT01050
     &    SE(4),SE(5),SE(6),SE(7),SE(8),SE(9),                         MAT01060
     &    SMAT(1),SMAT(2),SMAT(3),BUG,                                 MAT01070
     &  SE(10),SE(11),SE(14),SE(20),SE(21),SE(24),SE(25),SE(26),SE(30), MAT01080
     &  SMAT(4) )                                                      MAT01090
      SE(26)=SE(20)*SMAT(2)/2                                          MAT01100
      SE(27)=SMAT(1)*SMAT(4)                                           MAT01110
      SE(28)=0                                                         MAT01120
      SE(29)=0                                                         MAT01130
      SE(31)=SE(1)                                                     MAT01140
C                                                                      MAT01150
      RETURN                                                           MAT01160
C------------------------ GET STIFFNESS TERMS ------------------------MAT01170
  300 CONTINUE                                                         MAT01180
      SE(2)=P-PL                                                       MAT01190
      SE(3)=D-DL                                                       MAT01200
C                                                                      MAT01210
C-- CALL HYST11 TO CALC NEW STIFFNESS STIF                             MAT01220
      MOPT=IOPT                                                        MAT01230
      CALL HYST11(MOPT,SE(1),P,D,PL,DL,SE(2),SE(3),                    MAT01240
     &    SE(4),SE(5),SE(6),SE(7),SE(8),SE(9),                         MAT01250
     &    SMAT(1),SMAT(2),SMAT(3),BUG,                                 MAT01260
     & SE(10),SE(11),SE(14),SE(20),SE(21),SE(24),SE(25),SE(26),SE(30), MAT01270
     & SMAT(4) )                                                       MAT01280
C                                                                      MAT01290
C --- STORE MAXIMUM DISPLACEMENT AND CORRESPONDING LOAD               MAT01300
      IF(ABS(D) .GT. ABS(SE(28))) THEN                                 MAT01310
              SE(28)=D                                                 MAT01320
              SE(29)=P                                                 MAT01330
      ENDIF                                                            MAT01340
C                                                                      MAT01350
C--- GET ENERGIES, DUCTILITY AND EXCURSION FOR BOTH IOPT=3 AND 4 ----  MAT01360
  400 CONTINUE                                                         MAT01370
C                                                                      MAT01380
C --- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                         MAT01390
      EESE=SE(21)                                                      MAT01400
      EPSE=SE(24)                                                      MAT01410
C                                                                      MAT01420
C --- TRANSFER DUCTILITIES AND EXCURSION RATIOS FOR BOTH IOPT=3 AND 4. MAT01430
      DO 410 I=1,3                                                     MAT01440
        DUCT(I)=SE(10+I)                                               MAT01450
        EXCR(I+3)=SE(13+I+3)                                           MAT01460
  410   EXCR(I)=SE(13+I)                                               MAT01470
C                                                                      MAT01480
      RETURN                                                           MAT01490
C------------------ DAMAGE INDEX ------------------------------------ MAT01500
  500 CONTINUE                                                         MAT01510
C..DAMAGE INDEX IS NOT AVAILABLE                                       MAT01520
C     EESE=SE(21)                                                      MAT01530
C     EPSE=SE(24)                                                      MAT01540
C     DMAX=SE(27)                                                      MAT01550
C     D=SE(28)                                                         MAT01560
C     P=SE(29)                                                         MAT01570
C     PT=SMAT(2)                                                       MAT01580
C     BDAM=SMAT(4)                                                     MAT01590
C     DAMAGE=ABS(D)/DMAX + BDAM / (PT*DMAX) * EPSE                     MAT01600
      DAMAGE=0                                                         MAT01610
C                                                                      MAT01620
      RETURN                                                           MAT01630
      END                                                              MAT01640
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP                                 MAT00010
C--------------------------------------------------------------------- MAT00020
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                   MAT00030
C           END DEBUG                                                  MAT00040
C--------------------------------------------------------------------- MAT00050
      SUBROUTINE MAT12                                                 MAT00060
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG,                     MAT00070
     &   MAT    ,SE  ,P,PL,D,DL,V,VL,LELEM,LMAT,                      MAT00080
     &   SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)                             MAT00090
C                                                                      MAT00100
      IMPLICIT REAL(A-H,O-Z)                                           MAT00110
C /8-15-92/                                                            MAT00120
C     COMMON /IDSTEP/ISTEP                                             MAT00130
C                                                                      MAT00140
      LOGICAL ERR,FIRST,PRINT,BUG                                      MAT00150
      LOGICAL BTEST                                                    MAT00160
      REAL INEB, INEH                                                  MAT00170
      DIMENSION SE(100),SMAT(LMAT),DUCT(3),EXCR(6),EEASAT(12,12)       MAT00180
      DIMENSION A(12,12),SAT(12,12)                                    MAT00190
      DIMENSION P(12),PL(12),D(12),DL(12)                              MAT00200
      CHARACTER*80 TYPE                                                MAT00210
C                                                                      MAT00220
C /8-15-92/                                                            MAT00230
C     IF(ISTEP .LT. 2 .OR. ISTEP .EQ. 57 .OR. ISTEP .EQ. 58 ) THEN     MAT00240
C       WRITE(6,*)'1######## ISTEP = ',ISTEP                           MAT00250
C       BUG= .TRUE.                                                    MAT00260
C     ELSE IF(ISTEP .GE. 65 .AND. ISTEP .LE. 67 ) THEN                 MAT00270
C       WRITE(6,*)'1######## ISTEP = ',ISTEP                           MAT00280
C       BUG= .TRUE.                                                    MAT00290
C     ELSE IF(ISTEP .GE. 100 .AND. ISTEP .LE. 102 ) THEN               MAT00300
C       WRITE(6,*)'1######## ISTEP = ',ISTEP                           MAT00310
C       BUG= .TRUE.                                                    MAT00320
C     ELSE                                                             MAT00330
C       BUG= .FALSE.                                                   MAT00340
C     ENDIF                                                            MAT00350
C                                                                      MAT00360
C                                                                      MAT00370
C-------- RESERVE STORAGE FOR ELEMENT                                  MAT00380
      LELEM= 99                                                        MAT00390
      IF (IOPT.EQ.0) RETURN                                            MAT00400
C                                                                      MAT00410
C---------------------------------------------------------------------MAT00420
C VARIABLES:                                                           MAT00430
C---------------------------------------------------------------------MAT00440
C---------- GLOBAL VARIABLES                                          MAT00450
C IOPT   = 1, INITIALIZE MATERIAL                                      MAT00460
C        = 2, INITIALIZE STIFFNESS                                     MAT00470
C        = 3, GET STIFFNESS TERM                                       MAT00480
C        = 4, GET ENERGIES, DUCTILITY AND EXCURSION                    MAT00490
C        = 5, CALCULATE DAMAGE INDEX                                   MAT00500
C RINPUT = INPUT DATA                                                  MAT00510
C SMAT   = INTERNAL STORAGE                                            MAT00520
C SE     = OUTPUT STIFFNESS                                            MAT00530
C---------------------------------------------------------------------MAT00540
      GO TO (100,200,300,400,500),IOPT                                 MAT00550
C                                                                      MAT00560
  100 CONTINUE                                                         MAT00570
C                                                                      MAT00580
C----- INTERPERTATE INPUT DATA                                         MAT00590
C     NSEG  = SMAT( 1)                                                 MAT00600
C     YS    = SMAT( 2)                                                 MAT00610
C     EM    = SMAT( 3)                                                 MAT00620
C     LIBN  = SMAT( 4)                                                 MAT00630
C     HH    = SMAT( 5)                                                 MAT00640
C     UU    = SMAT( 6)                                                 MAT00650
C     WW    = SMAT( 7)                                                 MAT00660
C     ZZ    = SMAT( 8)                                                 MAT00670
C     INEB  = SMAT( 9)                                                 MAT00680
C     INEH  = SMAT(10)                                                 MAT00690
C     ST    = SMAT(11)                                                 MAT00700
C     IREV1 = SMAT(12)                                                 MAT00710
C     IREV2 = SMAT(13)                                                 MAT00720
C     IREV3 = SMAT(14)                                                 MAT00730
C     IREV4 = SMAT(15)                                                 MAT00740
C     IECOP = SMAT(16)                                                 MAT00750
C     NUMP  = SMAT(17)                                                 MAT00760
C     SMALL = SMAT(18)                                                 MAT00770
C     RATIO1= SMAT(19)                                                 MAT00780
C     RATIO= SMAT(20)                                                  MAT00790
C     TOTA  = SMAT(21)                                                 MAT00800
C     IAUTO = SMAT(22)                                                 MAT00810
C     IMATER= SMAT(23)                                                 MAT00820
C     RATIO3= SMAT(24)                                                 MAT00830
C     IR    = SMAT(25)                                                 MAT00840
C     G     = SMAT(26)                                                 MAT00850
C     QANEE = SMAT(27)                                                 MAT00860
C     ISTIF = SMAT(28)                                                 MAT00870
C     LMAT  = SMAT(29)                                                 MAT00880
C                                                                      MAT00890
      READ (5,*) TYPE ,NSEG,SMAT(2),SMAT(3),LIBN,(SMAT(I),I=5,8),     MAT00900
     &           INEB,INEH,SMAT(11),IREV1,IREV2,IREV3,IREV4,IECOP,     MAT00910
     &           (SMAT(I),I=18,21),IAUTO,IMATER,SMAT(24),IR,           MAT00920
     &           SMAT(26),SMAT(27),ISTIF                               MAT00930
                                                                       MAT00940
      SMAT(1)=NSEG                                                     MAT00950
      SMAT(4)=LIBN                                                     MAT00960
      SMAT(9)=INEB                                                     MAT00970
      SMAT(10)=INEH                                                    MAT00980
      SMAT(12)=IREV1                                                   MAT00990
      SMAT(13)=IREV2                                                   MAT01000
      SMAT(14)=IREV3                                                   MAT01010
      SMAT(15)=IREV4                                                   MAT01020
      SMAT(16)=IECOP                                                   MAT01030
      SMAT(22)=IAUTO                                                   MAT01040
      SMAT(23)=IMATER                                                  MAT01050
      SMAT(25)=IR                                                      MAT01060
      SMAT(28)=ISTIF                                                   MAT01070
C                                                                      MAT01080
C                                                                      MAT01090
      IF( FIRST .OR. BUG ) WRITE(6,181)                                MAT01100
        IF( LIBN .EQ. 1 ) THEN                                         MAT01110
            NUMP=4*(INEB+INEH)                                         MAT01120
            SMAT(17)=NUMP                                              MAT01130
            WRITE(6,191)                                               MAT01140
     &           MAT,  NSEG,SMAT(2),SMAT(3),LIBN,(SMAT(I),I=5,8),      MAT01150
     &           INEB,INEH,SMAT(11),IREV1,IREV2,IREV3,IREV4,IECOP,     MAT01160
     &           NUMP,(SMAT(I),I=18,21),IAUTO,IMATER,SMAT(24),IR,      MAT01170
     &           SMAT(26),SMAT(27),ISTIF                               MAT01180
        ELSE IF( LIBN .EQ. 2 ) THEN                                    MAT01190
            NUMP=4*INEB                                                MAT01200
            SMAT(17)=NUMP                                              MAT01210
            WRITE(6,192)                                               MAT01220
     &           MAT,  NSEG,SMAT(2),SMAT(3),LIBN,(SMAT(I),I=5,8),      MAT01230
     &           INEB,INEH,SMAT(11),IREV1,IREV2,IREV3,IREV4,IECOP,     MAT01240
     &           NUMP,(SMAT(I),I=18,21),IAUTO,IMATER,SMAT(24),IR,      MAT01250
     &           SMAT(26),SMAT(27),ISTIF                               MAT01260
        ELSE IF( LIBN .EQ. 3 ) THEN                                    MAT01270
            NUMP=4*INEB+INEH                                           MAT01280
            SMAT(17)=NUMP                                              MAT01290
            WRITE(6,194)                                               MAT01300
     &           MAT,  NSEG,SMAT(2),SMAT(3),LIBN,(SMAT(I),I=5,8),      MAT01310
     &           INEB,INEH,SMAT(11),IREV1,IREV2,IREV3,IREV4,IECOP,     MAT01320
     &           NUMP,(SMAT(I),I=18,21),IAUTO,IMATER,SMAT(24),IR,      MAT01330
     &           SMAT(26),SMAT(27),ISTIF                               MAT01340
        ELSE IF( LIBN .EQ. 4 ) THEN                                    MAT01350
            NUMP=IREV1*IREV2*2 + IREV3*IREV4                           MAT01360
            SMAT(17)=NUMP                                              MAT01370
            WRITE(6,196)                                               MAT01380
     &           MAT,  NSEG,SMAT(2),SMAT(3),LIBN,(SMAT(I),I=5,8),      MAT01390
     &           INEB,INEH,SMAT(11),IREV1,IREV2,IREV3,IREV4,IECOP,     MAT01400
     &           NUMP,(SMAT(I),I=18,21),IAUTO,IMATER,SMAT(24),IR,      MAT01410
     &           SMAT(26),SMAT(27),ISTIF                               MAT01420
        ELSE IF( LIBN .EQ. 6 ) THEN                                    MAT01430
            NUMP=8*INEB                                                MAT01440
            SMAT(17)=NUMP                                              MAT01450
            WRITE(6,192)                                               MAT01460
     &           MAT,  NSEG,SMAT(2),SMAT(3),LIBN,(SMAT(I),I=5,8),      MAT01470
     &           INEB,INEH,SMAT(11),IREV1,IREV2,IREV3,IREV4,IECOP,     MAT01480
     &           NUMP,(SMAT(I),I=18,21),IAUTO,IMATER,SMAT(24),IR,      MAT01490
     &           SMAT(26),SMAT(27),ISTIF                               MAT01500
        ELSE IF( LIBN .EQ. 7 ) THEN                                    MAT01510
            NUMP=IREV1*IREV2 + IREV3*IREV4                             MAT01520
            SMAT(17)=NUMP                                              MAT01530
            WRITE(6,193)                                               MAT01540
     &           MAT,  NSEG,SMAT(2),SMAT(3),LIBN,(SMAT(I),I=5,8),      MAT01550
     &           INEB,INEH,SMAT(11),IREV1,IREV2,IREV3,IREV4,IECOP,     MAT01560
     &           NUMP,(SMAT(I),I=18,21),IAUTO,IMATER,SMAT(24),IR,      MAT01570
     &           SMAT(26),SMAT(27),ISTIF                               MAT01580
        ELSE IF( LIBN .EQ. 8 ) THEN                                    MAT01590
            NUMP=IREV1*IREV2 + IREV3*IREV4                             MAT01600
            SMAT(17)=NUMP                                              MAT01610
            WRITE(6,193)                                               MAT01620
     &           MAT,  NSEG,SMAT(2),SMAT(3),LIBN,(SMAT(I),I=5,8),      MAT01630
     &           INEB,INEH,SMAT(11),IREV1,IREV2,IREV3,IREV4,IECOP,     MAT01640
     &           NUMP,(SMAT(I),I=18,21),IAUTO,IMATER,SMAT(24),IR,      MAT01650
     &           SMAT(26),SMAT(27),ISTIF                               MAT01660
        ELSE IF( LIBN .EQ. 9 ) THEN                                    MAT01670
            NUMP=IREV1*IREV2 + IREV3*IREV4 + INEB*INEH                 MAT01680
            SMAT(17)=NUMP                                              MAT01690
            WRITE(6,195)                                               MAT01700
     &           MAT,  NSEG,SMAT(2),SMAT(3),LIBN,(SMAT(I),I=5,8),      MAT01710
     &           INEB,INEH,SMAT(11),IREV1,IREV2,IREV3,IREV4,IECOP,     MAT01720
     &           NUMP,(SMAT(I),I=18,21),IAUTO,IMATER,SMAT(24),IR,      MAT01730
     &           SMAT(26),SMAT(27),ISTIF                               MAT01740
        ELSE                                                           MAT01750
            WRITE(6,199)MAT,SMAT(4)                                    MAT01760
        ENDIF                                                          MAT01770
C                                                                      MAT01780
  181 FORMAT (///' STABILITY ELEMENT'/                                 MAT01790
     &        ' ------------------'//)                                 MAT01800
  191 FORMAT (1X,'MAT. NO.     =',I3,2X,' BOX SECTION'/                MAT01810
     &        1X,'NSEG        =',1P,G10.3,9X,'YS          =',1P,G10.3/ MAT01820
```

```
     &        1X,'EM              =',1P,G10.3,9X,'LIBN          =',1P,G10.3/
     &        1X,'BOX HEIGTH      =',1P,G10.3,9X,'BOX WILTH     =',1P,G10.3/
     &        1X,'ECCXO           =',1P,G10.3,9X,'ECCYO         =',1P,G10.3/
     &        1X,'H.W.SEC.NO.     =',1P,G10.3,9X,'H.W.SEC.NO.   =',1P,G10.3/
     &        1X,'THICKNESS       =',1P,G10.3,9X,'L.B. FLAG     =',1P,G10.3/
     &        1X,'IREV2           =',1P,G10.3,9X,'IREV3         =',1P,G10.3/
     &        1X,'IREV4           =',1P,G10.3,9X,'IECOP         =',1P,G10.3/
     &        1X,'NUMP            =',1P,G10.3,9X,'SMALL         =',1P,G10.3/
     &        1X,'RATIXO          =',1P,G10.3,9X,'RATITO        =',1P,G10.3/
     &        1X,'TOTA            =',1P,G10.3,9X,'IAUTO         =',1P,G10.3/
     &        1X,'IMATER          =',1P,G10.3,9X,'RATIO3        =',1P,G10.3/
     &        1X,'IR              =',1P,G10.3,9X,'G             =',1P,G10.3/
     &      1X,'QRNEE       =',1P,G10.3,9X,'ISTIF       =',1P,G10.3//)
C
  192  FORMAT (1X,'MAT. NO.     =',I3,2X,' TUBE SECTION'/
     &        1X,'NSEG            =',1P,G10.3,9X,'YS            =',1P,G10.3/
     &        1X,'EM              =',1P,G10.3,9X,'LIBN          =',1P,G10.3/
     &        1X,'RADIUS          =',1P,G10.3,9X,'DUMMY VAR.    =',1P,G10.3/
     &        1X,'ECCXO           =',1P,G10.3,9X,'ECCYO         =',1P,G10.3/
     &        1X,'1/4.SEC.NO.     =',1P,G10.3,9X,'DUMMY VAR.    =',1P,G10.3/
     &        1X,'THICKNESS       =',1P,G10.3,9X,'IREV1         =',1P,G10.3/
     &        1X,'IREV2           =',1P,G10.3,9X,'IREV3         =',1P,G10.3/
     &        1X,'IREV4           =',1P,G10.3,9X,'IECOP         =',1P,G10.3/
     &        1X,'NUMP            =',1P,G10.3,9X,'SMALL         =',1P,G10.3/
     &        1X,'RATIXO          =',1P,G10.3,9X,'RATITO        =',1P,G10.3/
     &        1X,'TOTA            =',1P,G10.3,9X,'IAUTO         =',1P,G10.3/
     &        1X,'IMATER          =',1P,G10.3,9X,'RATIO3        =',1P,G10.3/
     &        1X,'IR              =',1P,G10.3,9X,'G             =',1P,G10.3/
     &      1X,'QRNEE       =',1P,G10.3,9X,'ISTIF       =',1P,G10.3//)
C
  193  FORMAT (1X,'MAT. NO.     =',I3,2X,' ANGLE SECTION'/
     &        1X,'NSEG            =',1P,G10.3,9X,'YS            =',1P,G10.3/
     &        1X,'EM              =',1P,G10.3,9X,'LIBN          =',1P,G10.3/
     &        1X,'LAG LENGTH      =',1P,G10.3,9X,'DUMMY VAR.    =',1P,G10.3/
     &        1X,'ECCXO           =',1P,G10.3,9X,'ECCYO         =',1P,G10.3/
     &        1X,'LAG.SEC.NO.     =',1P,G10.3,9X,'DUMMY VAR.    =',1P,G10.3/
     &        1X,'THICKNESS       =',1P,G10.3,9X,'N.R.IN U DIR=',1P,G10.3/
     &        1X,'N.C.IN U DIR=',1P,G10.3,9X,'N.R.IN V DIR=',1P,G10.3/
     &        1X,'N.C.IN V DIR=',1P,G10.3,9X,'IECOP         =',1P,G10.3/
     &        1X,'NUMP            =',1P,G10.3,9X,'SMALL         =',1P,G10.3/
     &        1X,'RATIXO          =',1P,G10.3,9X,'RATITO        =',1P,G10.3/
     &        1X,'TOTA            =',1P,G10.3,9X,'IAUTO         =',1P,G10.3/
     &        1X,'IMATER          =',1P,G10.3,9X,'RATIO3        =',1P,G10.3/
     &        1X,'IR              =',1P,G10.3,9X,'G             =',1P,G10.3/
     &      1X,'QRNEE       =',1P,G10.3,9X,'ISTIF       =',1P,G10.3//)
C
  194  FORMAT (1X,'MAT. NO.     =',I3,2X,' RECTANGULAR SECTION'/
     &        1X,'NSEG            =',1P,G10.3,9X,'YS            =',1P,G10.3/
     &        1X,'EM              =',1P,G10.3,9X,'LIBN          =',1P,G10.3/
     &        1X,'SEC HEIGTH      =',1P,G10.3,9X,'SEC WILTH     =',1P,G10.3/
     &        1X,'ECCXO           =',1P,G10.3,9X,'ECCYO         =',1P,G10.3/
     &        1X,'UU SEC.NO.      =',1P,G10.3,9X,'HH SEC.NO.    =',1P,G10.3/
     &        1X,'DUMMY VAR.      =',1P,G10.3,9X,'IREV1         =',1P,G10.3/
     &        1X,'IREV2           =',1P,G10.3,9X,'IREV3         =',1P,G10.3/
     &        1X,'IREV4           =',1P,G10.3,9X,'IECOP         =',1P,G10.3/
     &        1X,'NUMP            =',1P,G10.3,9X,'SMALL         =',1P,G10.3/
     &        1X,'RATIXO          =',1P,G10.3,9X,'RATITO        =',1P,G10.3/
     &        1X,'TOTA            =',1P,G10.3,9X,'IAUTO         =',1P,G10.3/
     &        1X,'IMATER          =',1P,G10.3,9X,'RATIO3        =',1P,G10.3/
     &        1X,'IR              =',1P,G10.3,9X,'G             =',1P,G10.3/
     &      1X,'QRNEE       =',1P,G10.3,9X,'ISTIF       =',1P,G10.3//)
C
  195  FORMAT (1X,'MAT. NO.     =',I3,2X,' ANGLE WITH COVER PL.'/
     &        1X,'NSEG            =',1P,G10.3,9X,'YS            =',1P,G10.3/
     &        1X,'EM              =',1P,G10.3,9X,'LIBN          =',1P,G10.3/
     &        1X,'LAG LENGTH      =',1P,G10.3,9X,'COV.WIDTH     =',1P,G10.3/
     &        1X,'COV.THICKNES=',1P,G10.3,9X,'DUMMY VAR.    =',1P,G10.3/
     &        1X,'N.R. COV.       =',1P,G10.3,9X,'N.C. COV.     =',1P,G10.3/
     &        1X,'THICKNESS       =',1P,G10.3,9X,'N.R.IN U DIR=',1P,G10.3/
     &        1X,'N.C.IN U DIR=',1P,G10.3,9X,'N.R.IN V DIR=',1P,G10.3/
     &        1X,'N.C.IN V DIR=',1P,G10.3,9X,'IECOP         =',1P,G10.3/
     &        1X,'NUMP            =',1P,G10.3,9X,'SMALL         =',1P,G10.3/
     &        1X,'RATIXO          =',1P,G10.3,9X,'RATITO        =',1P,G10.3/
     &        1X,'TOTA            =',1P,G10.3,9X,'IAUTO         =',1P,G10.3/
     &        1X,'IMATER          =',1P,G10.3,9X,'RATIO3        =',1P,G10.3/
     &        1X,'IR              =',1P,G10.3,9X,'G             =',1P,G10.3/
     &      1X,'QRNEE       =',1P,G10.3,9X,'ISTIF       =',1P,G10.3//)
C
  196  FORMAT (1X,'MAT. NO.     =',I3,2X,' WIDE-FLANGE SECTION'/
     &        1X,'NSEG            =',1P,G10.3,9X,'YS            =',1P,G10.3/
     &        1X,'EM              =',1P,G10.3,9X,'LIBN          =',1P,G10.3/
     &        1X,'BOX HEIGTH      =',1P,G10.3,9X,'BOX WILTH     =',1P,G10.3/
     &        1X,'ECCXO           =',1P,G10.3,9X,'ECCYO         =',1P,G10.3/
     &        1X,'WEB THICK.      =',1P,G10.3,9X,'DUMMY VAR.    =',1P,G10.3/
     &        1X,'FLA.THICK.      =',1P,G10.3,9X,'IREV1         =',1P,G10.3/
     &        1X,'IREV2           =',1P,G10.3,9X,'IREV3         =',1P,G10.3/
     &        1X,'IREV4           =',1P,G10.3,9X,'IECOP         =',1P,G10.3/
     &        1X,'NUMP            =',1P,G10.3,9X,'SMALL         =',1P,G10.3/
     &        1X,'RATIXO          =',1P,G10.3,9X,'RATITO        =',1P,G10.3/
     &        1X,'TOTA            =',1P,G10.3,9X,'IAUTO         =',1P,G10.3/
     &        1X,'IMATER          =',1P,G10.3,9X,'RATIO3        =',1P,G10.3/
     &        1X,'IR              =',1P,G10.3,9X,'G             =',1P,G10.3/
     &      1X,'QRNEE       =',1P,G10.3,9X,'ISTIF       =',1P,G10.3//)
C
  199  FORMAT (1X,'MAT. NO.     =',I3,' IS NOT AVAILABLE',
     &        ' LIBN= ',G10.3,' CHECK LIBN '/)
C
      I1=29+1
      I2=I1+NSEG*12
      I3=I2+(NSEG+1)*6
      I4=I3+NSEG
      I5=I4+2
      I6=I5+2
      I7=I6+NSEG*NUMP
      I8=I7+NSEG*NUMP
      I9=I8+NUMP
      I10=I9+NUMP
      I11=I10+NSEG*2*3
      I12=I11+NSEG
      I13=I12+(NSEG+1)*6
      I14=I13+(NSEG+1)*6
      I15=I14+(NSEG+1)*6
      I16=I15+NSEG
      I17=I16+NSEG*12
      I18=I17+NSEG*12
      I19=I18+NSEG*12
      I20=I19+NSEG*12
      I21=I20+NSEG*NUMP
      I22=I21+NSEG*NUMP
      I23=I22+NSEG*NUMP
      I24=I23+NUMP
      I25=I24+NSEG*9
      I26=I25+NSEG*9
      I27=I26+((NSEG+1)*6)**2
      I28=I27+((NSEG+1)*6*((NSEG+1)*6+1))/2
      I29=I28+(NSEG+1)*6 + 1
      I30=I29+144
      I31=I30+144
      I32=I31+(NSEG+1)*6
      I33=I32+((NSEG+1)*6)**2
      I34=I33+NSEG*144
      I35=I34+2
      I36=I35+2
      I37=I36+(NSEG+1)*6
      I38=I37+(NSEG+1)*6
      I39=I38+(NSEG+1)*6
      I40=I39+((NSEG+1)*6)**2
      I41=I40+NUMP
      I42=I41+NSEG*NUMP
      I43=I42+NUMP
      I44=I43+NUMP
      I45=I44+NSEG*NUMP
      I46=I45+NSEG*NUMP
      I47=I46+NSEG*NUMP
      I48=I47+NSEG*NUMP
      I49=I48+NSEG*NUMP
      I50=I49+NSEG*NUMP
      I51=I50+NSEG*NUMP
C
      LMAT= I51 + NUMP - 1
      SMAT(29)=LMAT
C
      RETURN
C
C------------------------- INITIALIZE STIFFNESS TERMS ------------------
  200 CONTINUE
C     ---- STABILITY ELEMENT MATERIAL DATA STORAGE FOR MEMBER ----
C     SE(1- 78)= EEASAT
C     SE(   79)=ELS
C     SE(   80)=IMD
C     SE(   81)=ISPE
C     SE(   82)=DELTP2
C     SE(   83)=S2
C     SE(   84)=SP
C     SE(   85)=ISTART
C     SE(   86)=RNEEE
C
C     SE(   87)=EESE
C     SE(   88)=EPSE
C     SE(   89)=DUCT(1)
C     SE(   90)=DUCT(2)
C     SE(   91)=DUCT(3)
C     SE(   92)=EXCR(1)
C     SE(   93)=EXCR(2)
C     SE(   94)=EXCR(3)
C     SE(   95)=EXCR(4)
C     SE(   96)=EXCR(5)
C     SE(   97)=EXCR(6)
C     SE(   98)=FLP FLAG FOR JUDGING LOCAL BUCKLING
C               0  :  NO LOCAL BUCKLING OCCURRED
C               1-4:  LOCAL BUCKLING OCCURRED
C     SE(   99)=FLAG FOR JUDGING LOCAL BUCKLING HAS OCCURRED OR NOT
C               0  :  NO LOCAL BUCKLING HAS OCCURRED
C               1  :  LOCAL BUCKLING HAS OCCURRED
C                     SET COMPRESSION STIFFNESS=0.001
C-----------------------------------------------------
C
      NSEG  = SMAT( 1)
      YS    = SMAT( 2)
      EM    = SMAT( 3)
      LIBN  = SMAT( 4)
      HH    = SMAT( 5)
      UU    = SMAT( 6)
      WW    = SMAT( 7)
      ZZ    = SMAT( 8)
      INEB  = SMAT( 9)
      INEH  = SMAT(10)
      ST    = SMAT(11)
      IREV1 = SMAT(12)
      IREV2 = SMAT(13)
      IREV3 = SMAT(14)
      IREV4 = SMAT(15)
      IECOP = SMAT(16)
      NUMP  = SMAT(17)
      SMALL = SMAT(18)
      RATIXO= SMAT(19)
      RATITO= SMAT(20)
      TOTA  = SMAT(21)
      IAUTO = SMAT(22)
      IMATER= SMAT(23)
      RATIO3= SMAT(24)
      IR    = SMAT(25)
      G     = SMAT(26)
      QRNEE = SMAT(27)
      ISTIF = SMAT(28)
      ISPE=0
      ISTART=0
      ELS=SE(79)
      I1=29+1
      I2=I1+NSEG*12
      I3=I2+(NSEG+1)*6
      I4=I3+NSEG
      I5=I4+2
      I6=I5+2
      I7=I6+NSEG*NUMP
      I8=I7+NSEG*NUMP
      I9=I8+NUMP
      I10=I9+NUMP
      I11=I10+NSEG*2*3
      I12=I11+NSEG
      I13=I12+(NSEG+1)*6
      I14=I13+(NSEG+1)*6
      I15=I14+(NSEG+1)*6
      I16=I15+NSEG
      I17=I16+NSEG*12
      I18=I17+NSEG*12
      I19=I18+NSEG*12
      I20=I19+NSEG*12
      I21=I20+NSEG*NUMP
      I22=I21+NSEG*NUMP
      I23=I22+NSEG*NUMP
      I24=I23+NUMP
      I25=I24+NSEG*9
      I26=I25+NSEG*9
      I27=I26+((NSEG+1)*6)**2
      I28=I27+((NSEG+1)*6*((NSEG+1)*6+1))/2
      I29=I28+(NSEG+1)*6 + 1
      I30=I29+144
      I31=I30+144
      I32=I31+(NSEG+1)*6
      I33=I32+((NSEG+1)*6)**2
      I34=I33+NSEG*144
      I35=I34+2
      I36=I35+2
      I37=I36+(NSEG+1)*6
      I38=I37+(NSEG+1)*6
      I39=I38+(NSEG+1)*6
      I40=I39+((NSEG+1)*6)**2
      I41=I40+NUMP
      I42=I41+NSEG*NUMP
      I43=I42+NUMP
      I44=I43+NUMP
      I45=I44+NSEG*NUMP
      I46=I45+NSEG*NUMP
      I47=I46+NSEG*NUMP
      I48=I47+NSEG*NUMP
      I49=I48+NSEG*NUMP
      I50=I49+NSEG*NUMP
      I51=I50+NSEG*NUMP
C
      DO 210 I=1,78
  210   SE(I)=0
      ELS=SE( 79)
      SE( 99)=0
C
C-----CALIBRATE TWO-END DISPLACEMENT INCREMENT BEFORE CALL HYST12
      DO 66 I=1,10,3
        SMAT(I13+I-1)=D(2+I-1)-DL(2+I-1)
        SMAT(I13+I  )=D(3+I-1)-DL(3+I-1)
        SMAT(I13+I+1)=D(1+I-1)-DL(1+I-1)
   66 CONTINUE
C     DO 66 I=1,12
C  66   SMAT(I13+I-1)=D(I)-DL(I)
C
```

```
      MOPT=IOPT                                                         MAT04670
            CALL HYST12( MOPT ,   NSEG,   YS,   EM,   LIBN,    HH,    UU,MAT04680
     &                   WW   ,   ZZ  , INEB, INEH,   ST  , IREV1,IREV2,MAT04690
     &                   IREV3,  IREV4,IECOP,ECCXO,  ECCYO,  NUMP,      MAT04700
     &                   ELS  ,  SMALL,RATIXO,RATIYO,  TOTA,  IMD,      MAT04710
     &                   ISPE , DELTP2,S2    ,SP   ,ISTART, BUG  ,      MAT04720
     &                   IELNO, ISTIF, IAUTO,IMATER,                    MAT04730
     &                   RATIO3, IR, G, QRNEE, RNEEE,                   MAT04740
C                                                                       MAT04750
     & SMAT(I1),SMAT(I2),SMAT(I3),SMAT(I4),SMAT(I5),SMAT(I6),SMAT(I7),  MAT04760
     & SMAT(I8),SMAT(I9),SMAT(I10),SMAT(I11),SMAT(I12),SMAT(I13),       MAT04770
     & SMAT(I14),SMAT(I15),SMAT(I16),SMAT(I17),SMAT(I18),SMAT(I19),     MAT04780
     & SMAT(I20),SMAT(I21),SMAT(I22),SMAT(I23),SMAT(I24),SMAT(I25),     MAT04790
     & SMAT(I26),SMAT(I27),SMAT(I28),SMAT(I29),SMAT(I30),SMAT(I31),     MAT04800
     & SMAT(I32),SMAT(I33),SMAT(I34),SMAT(I35),SMAT(I36),SMAT(I37),     MAT04810
     & SMAT(I38),SMAT(I39),SMAT(I40),SMAT(I41),SMAT(I42),SMAT(I43),     MAT04820
     & SMAT(I44),SMAT(I45),SMAT(I46),SMAT(I47),SMAT(I48),SMAT(I49),     MAT04830
     & SMAT(I50),SMAT(I51),SE(98) )                                     MAT04840
C                                                                       MAT04850
       K=0                                                              MAT04860
       DO 60 I=1,12                                                     MAT04870
       DO 60 J=1,12                                                     MAT04880
        K=K+1                                                           MAT04890
   60   EEASAT(I,J)=SMAT(I29+K-1)                                       MAT04900
C                                                                       MAT04910
C-----CONVERT STABILITY ELEMENT TWO-END STIFFNESS INTO STANDARD DOF     MAT04920
C     DIRECTION                                                         MAT04930
      DO 68 I=1,12                                                      MAT04940
      DO 68 J=1,12                                                      MAT04950
   68  A(I,J)=0                                                         MAT04960
C                                                                       MAT04970
      DO 69 I=1,4                                                       MAT04980
        A(1+(I-1)*3,3+(I-1)*3)=1                                        MAT04990
        A(2+(I-1)*3,1+(I-1)*3)=1                                        MAT05000
        A(3+(I-1)*3,2+(I-1)*3)=1                                        MAT05010
   69 CONTINUE                                                          MAT05020
C                                                                       MAT05030
      DO 90 I=1,12                                                      MAT05040
      DO 90 J=1,12                                                      MAT05050
         SAT(I,J)=0                                                     MAT05060
         DO 90 K=1,12                                                   MAT05070
   90       SAT(I,J)=SAT(I,J)+EEASAT(I,K)*A(J,K)                       MAT05080
C                                                                       MAT05090
      DO 91 I=1,12                                                      MAT05100
      DO 91 J=1,12                                                      MAT05110
         EEASAT(I,J)=0                                                  MAT05120
         DO 91 K=1,12                                                   MAT05130
   91       EEASAT(I,J)=EEASAT(I,J)+A(I,K)*SAT(K,J)                    MAT05140
C                                                                       MAT05150
       K=0                                                              MAT05160
       DO 80 I=1,12                                                     MAT05170
       DO 80 J=I,12                                                     MAT05180
       K=K+1                                                            MAT05190
   80    SE(K)=EEASAT(I,J)                                              MAT05200
      SE(   79)=ELS                                                     MAT05210
      SE(   80)=IMD                                                     MAT05220
      SE(   81)=ISPE                                                    MAT05230
      SE(   82)=DELTP2                                                  MAT05240
      SE(   83)=S2                                                      MAT05250
      SE(   84)=SP                                                      MAT05260
      SE(   85)=ISTART                                                  MAT05270
      SE(   86)=RNEEE                                                   MAT05280
C                                                                       MAT05290
      RETURN                                                            MAT05300
C------------------------- GET STIFFNESS TERMS ------------------------MAT05310
  300 CONTINUE                                                          MAT05320
C                                                                       MAT05330
      ELS   = SE(   79)                                                 MAT05340
      IMD   = SE(   80)                                                 MAT05350
      ISPE  = SE(   81)                                                 MAT05360
      DELTP2= SE(   82)                                                 MAT05370
      S2    = SE(   83)                                                 MAT05380
      SP    = SE(   84)                                                 MAT05390
      ISTART= SE(   85)                                                 MAT05400
      RNEEE = SE(   86)                                                 MAT05410
C                                                                       MAT05420
C-- CALL HYST12 TO CALC NEW STIFFNESS STIF                              MAT05430
      NSEG  = SMAT( 1)                                                  MAT05440
      YS    = SMAT( 2)                                                  MAT05450
      EM    = SMAT( 3)                                                  MAT05460
      LIBN  = SMAT( 4)                                                  MAT05470
      HH    = SMAT( 5)                                                  MAT05480
      UU    = SMAT( 6)                                                  MAT05490
      WW    = SMAT( 7)                                                  MAT05500
      ZZ    = SMAT( 8)                                                  MAT05510
      INEB  = SMAT( 9)                                                  MAT05520
      INEH  = SMAT(10)                                                  MAT05530
      ST    = SMAT(11)                                                  MAT05540
      IREV1 = SMAT(12)                                                  MAT05550
      IREV2 = SMAT(13)                                                  MAT05560
      IREV3 = SMAT(14)                                                  MAT05570
      IREV4 = SMAT(15)                                                  MAT05580
      IECOP = SMAT(16)                                                  MAT05590
      NUMP  = SMAT(17)                                                  MAT05600
      SMALL = SMAT(18)                                                  MAT05610
      RATIXO= SMAT(19)                                                  MAT05620
      RATIYO= SMAT(20)                                                  MAT05630
      TOTA  = SMAT(21)                                                  MAT05640
      IAUTO = SMAT(22)                                                  MAT05650
      IMATER= SMAT(23)                                                  MAT05660
      RATIO3= SMAT(24)                                                  MAT05670
      IR    = SMAT(25)                                                  MAT05680
      G     = SMAT(26)                                                  MAT05690
      QRNEE = SMAT(27)                                                  MAT05700
      ISTIF = SMAT(28)                                                  MAT05710
      ISPE=0                                                            MAT05720
      ELS=SE(79)                                                        MAT05730
      I1=29+1                                                           MAT05740
      I2=I1+NSEG*12                                                     MAT05750
      I3=I2+(NSEG+1)*6                                                  MAT05760
      I4=I3+NSEG                                                        MAT05770
      I5=I4+2                                                           MAT05780
      I6=I5+2                                                           MAT05790
      I7=I6+NSEG*NUMP                                                   MAT05800
      I8=I7+NSEG*NUMP                                                   MAT05810
      I9=I8+NUMP                                                        MAT05820
      I10=I9+NUMP                                                       MAT05830
      I11=I10+NSEG*2*3                                                  MAT05840
      I12=I11+NSEG                                                      MAT05850
      I13=I12+(NSEG+1)*6                                                MAT05860
      I14=I13+(NSEG+1)*6                                                MAT05870
      I15=I14+(NSEG+1)*6                                                MAT05880
      I16=I15+NSEG                                                      MAT05890
      I17=I16+NSEG*12                                                   MAT05900
      I18=I17+NSEG*12                                                   MAT05910
      I19=I18+NSEG*12                                                   MAT05920
      I20=I19+NSEG*12                                                   MAT05930
      I21=I20+NSEG*NUMP                                                 MAT05940
      I22=I21+NSEG*NUMP                                                 MAT05950
      I23=I22+NSEG*NUMP                                                 MAT05960
      I24=I23+NUMP                                                      MAT05970
      I25=I24+NSEG*9                                                    MAT05980
      I26=I25+NSEG*9                                                    MAT05990
      I27=I26+((NSEG+1)*6)**2                                           MAT06000
      I28=I27+((NSEG+1)*6*((NSEG+1)*6+1))/2                             MAT06010
      I29=I28+(NSEG+1)*6 + 1                                            MAT06020
      I30=I29+144                                                       MAT06030
      I31=I30+144                                                       MAT06040
      I32=I31+(NSEG+1)*6                                                MAT06050
      I33=I32+((NSEG+1)*6)**2                                           MAT06060
      I34=I33+NSEG*144                                                  MAT06070
      I35=I34+2                                                         MAT06080
      I36=I35+2                                                         MAT06090
      I37=I36+(NSEG+1)*6                                                MAT06100
      I38=I37+(NSEG+1)*6                                                MAT06110
      I39=I38+(NSEG+1)*6                                                MAT06120
      I40=I39+((NSEG+1)*6)**2                                           MAT06130
      I41=I40+NUMP                                                      MAT06140
      I42=I41+NSEG*NUMP                                                 MAT06150
      I43=I42+NUMP                                                      MAT06160
      I44=I43+NUMP                                                      MAT06170
      I45=I44+NSEG*NUMP                                                 MAT06180
      I46=I45+NSEG*NUMP                                                 MAT06190
      I47=I46+NSEG*NUMP                                                 MAT06200
      I48=I47+NSEG*NUMP                                                 MAT06210
      I49=I48+NSEG*NUMP                                                 MAT06220
      I50=I49+NSEG*NUMP                                                 MAT06230
      I51=I50+NSEG*NUMP                                                 MAT06240
C                                                                       MAT06250
C----CALIBRATE TWO-END DISPLACEMENT INCREMENT BEFORE CALL HYST12         MAT06260
      DO 46 I=1,10,3                                                    MAT06270
        SMAT(I13+I-1)=D(2+I-1)-DL(2+I-1)                                MAT06280
        SMAT(I13+I  )=D(3+I-1)-DL(3+I-1)                                MAT06290
        SMAT(I13+I+1)=D(1+I-1)-DL(1+I-1)                                MAT06300
   46 CONTINUE                                                          MAT06310
C     DO 18 I=1,12                                                      MAT06320
C  18   SMAT(I13+I-1)=D(I)-DL(I)                                        MAT06330
C                                                                       MAT06340
      MOPT=IOPT                                                         MAT06350
            CALL HYST12( MOPT ,   NSEG,   YS,   EM,   LIBN,    HH,    UU,MAT06360
     &                   WW   ,   ZZ  , INEB, INEH,   ST  , IREV1,IREV2,MAT06370
     &                   IREV3,  IREV4,IECOP,ECCXO,  ECCYO,  NUMP,      MAT06380
     &                   ELS  ,  SMALL,RATIXO,RATIYO,  TOTA,  IMD,      MAT06390
     &                   ISPE , DELTP2,S2    ,SP   ,ISTART, BUG  ,      MAT06400
     &                   IELNO, ISTIF, IAUTO,IMATER,                    MAT06410
     &                   RATIO3, IR, G, QRNEE, RNEEE,                   MAT06420
C                                                                       MAT06430
     & SMAT(I1),SMAT(I2),SMAT(I3),SMAT(I4),SMAT(I5),SMAT(I6),SMAT(I7),  MAT06440
     & SMAT(I8),SMAT(I9),SMAT(I10),SMAT(I11),SMAT(I12),SMAT(I13),       MAT06450
     & SMAT(I14),SMAT(I15),SMAT(I16),SMAT(I17),SMAT(I18),SMAT(I19),     MAT06460
     & SMAT(I20),SMAT(I21),SMAT(I22),SMAT(I23),SMAT(I24),SMAT(I25),     MAT06470
     & SMAT(I26),SMAT(I27),SMAT(I28),SMAT(I29),SMAT(I30),SMAT(I31),     MAT06480
     & SMAT(I32),SMAT(I33),SMAT(I34),SMAT(I35),SMAT(I36),SMAT(I37),     MAT06490
     & SMAT(I38),SMAT(I39),SMAT(I40),SMAT(I41),SMAT(I42),SMAT(I43),     MAT06500
     & SMAT(I44),SMAT(I45),SMAT(I46),SMAT(I47),SMAT(I48),SMAT(I49),     MAT06510
     & SMAT(I50),SMAT(I51),SE(98) )                                     MAT06520
C                                                                       MAT06530
       K=0                                                              MAT06540
       DO 65 I=1,12                                                     MAT06550
       DO 65 J=1,12                                                     MAT06560
        K=K+1                                                           MAT06570
   65   EEASAT(I,J)=SMAT(I29+K-1)                                       MAT06580
C                                                                       MAT06590
C-----CONVERT STABILITY ELEMENT TWO-END STIFFNESS INTO STANDARD DOF     MAT06600
C     DIRECTION                                                         MAT06610
      DO 38 I=1,12                                                      MAT06620
      DO 38 J=1,12                                                      MAT06630
   38  A(I,J)=0                                                         MAT06640
C                                                                       MAT06650
      DO 39 I=1,4                                                       MAT06660
        A(1+(I-1)*3,3+(I-1)*3)=1                                        MAT06670
        A(2+(I-1)*3,1+(I-1)*3)=1                                        MAT06680
        A(3+(I-1)*3,2+(I-1)*3)=1                                        MAT06690
   39 CONTINUE                                                          MAT06700
C                                                                       MAT06710
      DO 40 I=1,12                                                      MAT06720
      DO 40 J=1,12                                                      MAT06730
         SAT(I,J)=0                                                     MAT06740
         DO 40 K=1,12                                                   MAT06750
   40       SAT(I,J)=SAT(I,J)+EEASAT(I,K)*A(J,K)                       MAT06760
C                                                                       MAT06770
      DO 41 I=1,12                                                      MAT06780
      DO 41 J=1,12                                                      MAT06790
         EEASAT(I,J)=0                                                  MAT06800
         DO 41 K=1,12                                                   MAT06810
   41       EEASAT(I,J)=EEASAT(I,J)+A(I,K)*SAT(K,J)                    MAT06820
C                                                                       MAT06830
      IF( SE(99) .EQ. 1 ) THEN                                          MAT06840
         IF( P(7) .LE. 0 ) THEN                                         MAT06850
            XLL=DL(7)-DL(1)                                             MAT06860
            XCC=D(7)-D(1)                                               MAT06870
            XDD=XCC-XLL                                                 MAT06880
            IF( XDD .LT. 0 ) THEN                                       MAT06890
              DO 339 J=1,12                                             MAT06900
                EEASAT(1,J)=0                                           MAT06910
                EEASAT(7,J)=0                                           MAT06920
                EEASAT(J,1)=0                                           MAT06930
                EEASAT(J,7)=0                                           MAT06940
  339         CONTINUE                                                  MAT06950
              EEASAT(1,1)=0.001                                         MAT06960
              EEASAT(7,7)=0.001                                         MAT06970
              EEASAT(1,7)=-0.001                                        MAT06980
              EEASAT(7,1)=-0.001                                        MAT06990
            ENDIF                                                       MAT07000
         ENDIF                                                          MAT07010
      ENDIF                                                             MAT07020
      IF( SE(99) .EQ. 0 ) THEN                                          MAT07030
         IF( SE(98) .GT. 0 ) SE(99)=1                                   MAT07040
      ENDIF                                                             MAT07050
C                                                                       MAT07060
       K=0                                                              MAT07070
       DO 85 I=1,12                                                     MAT07080
       DO 85 J=I,12                                                     MAT07090
       K=K+1                                                            MAT07100
   85    SE(K)=EEASAT(I,J)                                              MAT07110
      SE(   79)=ELS                                                     MAT07120
      SE(   80)=IMD                                                     MAT07130
      SE(   81)=ISPE                                                    MAT07140
      SE(   82)=DELTP2                                                  MAT07150
      SE(   83)=S2                                                      MAT07160
      SE(   84)=SP                                                      MAT07170
      SE(   85)=ISTART                                                  MAT07180
      SE(   86)=RNEEE                                                   MAT07190
C                                                                       MAT07200
      RETURN                                                            MAT07210
C--- GET ENERGIES, DUCTILITY AND EXCURSION FOR BOTH IOPT=3 AND 4 ----   MAT07220
  400 CONTINUE                                                          MAT07230
C                                                                       MAT07240
C --- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                          MAT07250
      EESE=0                                                            MAT07260
      EPSE=0                                                            MAT07270
C                                                                       MAT07280
C --- TRANSFER DUCTILITIES AND EXCURSION RATIOS FOR BOTH IOPT=3 AND 4.  MAT07290
      DO 410 I=1,3                                                      MAT07300
        DUCT(I)=0                                                       MAT07310
        EXCR(I+3)=0                                                     MAT07320
  410   EXCR(I)=0                                                       MAT07330
C                                                                       MAT07340
       RETURN                                                           MAT07350
C------------------ DAMAGE INDEX -------------------------------------MAT07360
  500 CONTINUE                                                          MAT07370
      EESE=0                                                            MAT07380
      EPSE=0                                                            MAT07390
      DAMAGE=0                                                          MAT07400
C                                                                       MAT07410
      RETURN                                                            MAT07420
      END                                                               MAT07430
C$PROCESS SDUMP OPT(0) GOSTMT XREF MAP                                  MAT07440
C-----------------------------------------------------------------------MAT07450
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    MAT07460
C           END DEBUG                                                   MAT07470
C-----------------------------------------------------------------------MAT07480
      SUBROUTINE MAT13                                                  MAT07490
```

```
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG,                      MAT00070
     &   MAT    ,SE   ,P,PL,D,DL,V,VL,LELEM,LMAT,                       MAT00080
     &   SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)                               MAT00090
C                                                                       MAT00100
      IMPLICIT REAL(A-H,O-Z)                                            MAT00110
      LOGICAL ERR,FIRST,PRINT,BUG                                       MAT00120
      LOGICAL BTEST                                                     MAT00130
      DIMENSION SE(100),SMAT(8),DUCT(3),EXCR(6)                         MAT00140
      CHARACTER*80 TYPE                                                 MAT00150
C                                                                       MAT00160
C------- RESERVE STORAGE FOR ELEMENT                                    MAT00170
      LELEM=39                                                          MAT00180
      IF (IOPT.EQ.0) RETURN                                             MAT00190
C                                                                       MAT00200
C-----------------------------------------------------------------------MAT00210
C VARIABLES:                                                            MAT00220
C-----------------------------------------------------------------------MAT00230
C---------- GLOBAL VARIABLES                                           MAT00240
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT00250
C        = 2, GET STIFFNESS                                             MAT00260
C RINPUT = INPUT DATA                                                   MAT00270
C SMAT   = INTERNAL STORAGE                                             MAT00280
C SE     = OUTPUT STIFFNESS                                             MAT00290
C-----------------------------------------------------------------------MAT00300
      GO TO (100,200,300,400,500),IOPT                                  MAT00310
C                                                                       MAT00320
  100 CONTINUE                                                          MAT00330
C                                                                       MAT00340
C----- INTERPERTATE INPUT DATA                                          MAT00350
C       E    = SMAT( 1)                                                 MAT00360
C       A    = SMAT( 2)                                                 MAT00370
C       R    = SMAT( 3)                                                 MAT00380
C       YS   = SMAT( 4)                                                 MAT00390
C       PY   = SMAT( 5)                                                 MAT00400
C       CC   = SMAT( 6)                                                 MAT00410
C       DMAX = SMAT( 7)   : ULTIMATE DISPLACEMENT = DUCMAX*DELY         MAT00420
C       BDAM = SMAT( 8)   : BATA PARAMETER                              MAT00430
C                                                                       MAT00440
      READ (5,*) TYPE ,(SMAT(I),I=1,4), DUCMAX, BDAM                    MAT00450
C                                                                       MAT00460
      LMAT = 8                                                          MAT00470
      PI=3.1415926                                                      MAT00480
      SMAT(5)=SMAT(2)*SMAT(4)                                           MAT00490
      SMAT(6)=SQRT(2.*PI*PI*SMAT(1)/SMAT(4) )                           MAT00500
      SMAT(7)=DUCMAX                                                    MAT00510
      SMAT(8)=BDAM                                                      MAT00520
C                                                                       MAT00530
      IF (FIRST .OR. BUG) WRITE (6,191)                                 MAT00540
      WRITE (6,192) MAT,(SMAT(K),K=1,4),DUCMAX,BDAM                     MAT00550
C                                                                       MAT00560
  191 FORMAT (///'     GOEL HYSTERESIS MODEL FOR BOX LOCAL BUCKLING'/    MAT00570
     &        '     ------------------------------------------'//       MAT00580
     &  '  MAT.',3X,'     E',5X,'     A',6X,'     R',6X,'    YS',       MAT00590
     &           5X,'MAX DUC',5X,'  BETA')                              MAT00600
  192 FORMAT (1X,I5,6G12.6/)                                            MAT00610
C                                                                       MAT00620
      RETURN                                                            MAT00630
C                                                                       MAT00640
C------------------------ INITIALIZE STIFFNESS TERMS ------------------MAT00650
  200 CONTINUE                                                          MAT00660
C       ---- GOEL HYSTERESIS MODEL DATA STORAGE FOR MEMBER ----         MAT00670
C     SE( 1)=        STIFMEMBER AXIAL STIFFNESS                       MAT00680
C     SE( 2)= K=1    EFFECT SLENDER COEFFICIENT                       MAT00690
C     SE( 3)= NOCY   NO. OF CYCLE                                     MAT00700
C     SE( 4)= NRULE  RULE NUMBER                                      MAT00710
C     SE( 5)= EL     MEMBER LENGTH (TRANSFERED FROM ELE08)            MAT00720
C     SE( 6)= IRV    0:REVRSC=.FALSE.    1:REVRSC=.TRUE.              MAT00730
C     SE( 7)= IDUC   0: HC=-12           1: HC .LT. -12               MAT00740
C     SE( 8)= PBOT                                                    MAT00750
C     SE( 9)= DBOT                                                    MAT00760
C     SE(10)= PTOP                                                    MAT00770
C     SE(11)= DTOP                                                    MAT00780
C     SE(12)= KEQ    UNLOADING EQUIVALENT STIFFNESS                   MAT00790
C     SE(13)= DUCT(1)                                                 MAT00800
C     SE(14)= DUCT(2)                                                 MAT00810
C     SE(15)= DUCT(3)                                                 MAT00820
C     SE(16)= EXCR(1)                                                 MAT00830
C     SE(17)= EXCR(2)                                                 MAT00840
C     SE(18)= EXCR(3)                                                 MAT00850
C     SE(19)= EXCR(4)                                                 MAT00860
C     SE(20)= EXCR(5)                                                 MAT00870
C     SE(21)= EXCR(6)                                                 MAT00880
C     SE(22)= DELY   YIELDING DISPLACEMENT                            MAT00890
C     SE(23)= EESE(1)                                                 MAT00900
C     SE(24)= EESE(2)                                                 MAT00910
C     SE(25)= EESE(2)                                                 MAT00920
C     SE(26)= EPSE                                                    MAT00930
C     SE(27)= PSEOLD                                                  MAT00940
C     SE(28)= CSE                                                     MAT00950
C     SE(29)= DMAX   ULTIMATE DISPLACEMENT                            MAT00960
C     SE(30)= D_MAX  MAXIMUM ELEMENT DISPLACEMENT FOR DAMAGE INDEX    MAT00970
C     SE(31)= P_MAX  ELEMENT LOAD CORRESPONDING TO D_MAX              MAT00980
C     SE(32)= HA     CURRENT CONTROL POINT HA                         MAT00990
C     SE(33)= VA     CURRENT CONTROL POINT VA                         MAT01000
C     SE(34)= HC     CURRENT CONTROL POINT HC                         MAT01010
C     SE(35)= VC     CURRENT CONTROL POINT VC                         MAT01020
C     SE(36)= RSN    CURRENT RESIDUAL STRAIN VALUE                    MAT01030
C     SE(37)= DEMAX  CURRENT MAXIMUM AXIAL DISPLACEMENT FOR DUCTILITIES MAT01040
C     SE(38)= REGIN  CURRENT REGINE ZONE NUMBER                       MAT01050
C     SE(39)= PREGIN PREVIOUT REGIN ZONE NUMBER                       MAT01060
C ---------------------------------------------------------           MAT01070
      SE(1)=1                                                           MAT01080
      SE(2)=1                                                           MAT01090
      SE(3)=1                                                           MAT01100
      SE(4)=1                                                           MAT01110
      DO 210 I=6,28                                                     MAT01120
  210   SE(I)=0                                                         MAT01130
      SE(37)=0                                                          MAT01140
      SE(38)=1                                                          MAT01150
      SE(39)=1                                                          MAT01160
C                                                                       MAT01170
      CALL HYST13(0.,0.,0.,0.,V,VL,SE(5),SE(2),                        MAT01180
     &    SMAT(1),SMAT(2),SMAT(3),SMAT(4),SMAT(5),                      MAT01190
     &    SMAT(6),SE(3),SE(4),SE(1),SE(6),SE(7),                        MAT01200
     &    SE(32),SE(33),SE(34),SE(35),SE(36),SE(37),SE(38),SE(39),      MAT01210
     &    SE(8),SE(9),SE(10),SE(11),BUG,                                MAT01220
     &    SE(12),SE(13),SE(16),SE(22),SE(23),SE(26),SE(27),SE(28) )     MAT01230
      SE(28)=SE(22)*SMAT(5)/2                                           MAT01240
      SE(29)=SE(22)*SMAT(7)                                             MAT01250
      SE(30)=0                                                          MAT01260
      SE(31)=0                                                          MAT01270
C                                                                       MAT01280
      RETURN                                                            MAT01290
C------------------------- GET STIFFNESS TERMS -------------------------MAT01300
  300 CONTINUE                                                          MAT01310
C                                                                       MAT01320
C-------- CALL HYST13 TO CALC NEW STIFFNESS STIF                        MAT01330
      CALL HYST13(P,PL,D,DL,V,VL,SE(5),SE(2),                           MAT01340
     &    SMAT(1),SMAT(2),SMAT(3),SMAT(4),SMAT(5),                      MAT01350
     &    SMAT(6),SE(3),SE(4),SE(1),SE(6),SE(7),                        MAT01360
     &    SE(32),SE(33),SE(34),SE(35),SE(36),SE(37),SE(38),SE(39),      MAT01370
     &    SE(8),SE(9),SE(10),SE(11),BUG,                                MAT01380
     &    SE(12),SE(13),SE(16),SE(22),SE(23),SE(26),SE(27),SE(28) )     MAT01390
C                                                                       MAT01400
C --- STORE MAXIMUM DISPLACEMENT AND CORRESPONDING LOAD                 MAT01410
      IF(ABS(D) .GT. ABS(SE(30))) THEN                                  MAT01420
            SE(30)=D                                                    MAT01430
            SE(31)=P                                                    MAT01440
      ENDIF                                                             MAT01450
C                                                                       MAT01460
C--- GET ENERGIES, DUCTILITY AND EXCURSION FOR BOTH IOPT=3 AND 4 ----   MAT01470
  400 CONTINUE                                                          MAT01480
C                                                                       MAT01490
C --- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                          MAT01500
      EESE=SE(23)                                                       MAT01510
      EPSE=SE(26)                                                       MAT01520
C                                                                       MAT01530
C --- TRANSFER DUCTILITIES AND EXCURSION RATIOS FOR BOTH IOPT=3 AND 4.  MAT01540
      DO 410 I=1,3                                                      MAT01550
        DUCT(I)=SE(12+I)                                                MAT01560
        EXCR(I+3)=SE(15+I+3)                                            MAT01570
  410   EXCR(I)=SE(15+I)                                                MAT01580
C                                                                       MAT01590
      RETURN                                                            MAT01600
C----------------- DAMAGE INDEX ---------------------------------------MAT01610
  500 CONTINUE                                                          MAT01620
      EESE=SE(23)                                                       MAT01630
      EPSE=SE(26)                                                       MAT01640
      DMAX=SE(29)                                                       MAT01650
      D=SE(30)                                                          MAT01660
      P=SE(31)                                                          MAT01670
      PY=SMAT(5)                                                        MAT01680
      BDAM=SMAT(8)                                                      MAT01690
      DAMAGE=ABS(D)/DMAX + BDAM / (PY*DMAX) * EPSE                      MAT01700
C                                                                       MAT01710
      RETURN                                                            MAT01720
      END                                                               MAT01730
C-----------------------------------------------------------------------MAT00010
C       DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                        MAT00020
C       END DEBUG                                                       MAT00030
C-----------------------------------------------------------------------MAT00040
      SUBROUTINE MAT14                                                  MAT00050
     &  (IOPT   ,IELNO   ,ERR ,FIRST  ,PRINT ,BUG,                      MAT00060
     &   MAT    ,SE   ,P,PL,D,DL,V,VL,LELEM,LMAT,                       MAT00070
     &   SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)                               MAT00080
C                                                                       MAT00090
      LOGICAL ERR,FIRST,PRINT,BUG                                       MAT00100
      LOGICAL BTEST                                                     MAT00110
      DIMENSION SE(100),SMAT(40),DUCT(3),EXCR(6)                        MAT00120
      CHARACTER*80 TYPE                                                 MAT00130
      REAL KETD62,MQD,LWP                                               MAT00140
      INTEGER SECNO,CYCLEN                                              MAT00150
      REAL K,KP,DU0(4),FU0(4),KLAST                                     MAT00160
      LOGICAL TNPTFG, SRFLAG,POSTFG                                     MAT00170
C                                                                       MAT00180
C                                                                       MAT00190
      IF (IOPT.NE.1) LELEM=80                                           MAT00200
      IF (IOPT.EQ.0) RETURN                                             MAT00210
C                                                                       MAT00220
C-----------------------------------------------------------------------MAT00230
C VARIABLES:                                                            MAT00240
C-----------------------------------------------------------------------MAT00250
C---------- GLOBAL VARIABLES                                           MAT00260
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT00270
C        = 2, GET STIFFNESS                                             MAT00280
C RINPUT = INPUT DATA                                                   MAT00290
C SMAT   = INTERNAL STORAGE                                             MAT00300
C SE     = OUTPUT STIFFNESS                                             MAT00310
C-----------------------------------------------------------------------MAT00320
C           ---- SHEAR1 DATA ----                                       MAT00330
C             SMAT(1)= NSEGS,        NUMBER OF BACKBONE SEGMENTS        MAT00340
C             SMAT(2)= CSE,          CONSTAINT STRAIN ENERGY...         MAT00350
C             SMAT(3)= H,            HEIGHT OF SHEAR WALL               MAT00360
C             SMAT(4)= WID,          WIDTH OF SHEAR WALL                MAT00370
C             SMAT(4+I)= PP,         BACKBONE SHEAR FOR POINT I         MAT00380
C             SMAT(4+I+NSEG)= DP,    BACKBONE DISPLACEMENT FOR POINT I  MAT00390
C             SMAT(5+2*NSEG)= BDAM   BETA FOR DAMAGE INDEX              MAT00400
C             SMAT(6+2*NSEG)= HO,    HEIGHT OF OPENING                  MAT00410
C             SMAT(7+2*NSEG)= HS,    SHEAR SPAN LENGTH                  MAT00420
C             SMAT(8+2*NSEG)= LWP,                                      MAT00430
C                                                    LWP=L'/W1          MAT00440
C                                                                       MAT00450
C                                                    L'                 MAT00460
C                                                                       MAT00470
C                                                    W1                 MAT00480
C                                                                       MAT00490
C                                                                       MAT00500
C                                                                       MAT00510
C             SMAT(9+2*NSEG)= AD,    CROSS SECTION AREA OF DIAGONAL BAR MAT00520
C             SMAT(10+2*NSEG)=AV,    CROSS SECTION AREA OF VERTICAL BAR MAT00530
C             SMAT(11+2*NSEG)=MQD,   SHEAR SPAN LENGTH RATIO            MAT00540
C             SMAT(12+2*NSEG)=SOC=OSOC , ORIGINAL STIFFNESS             MAT00550
C             SMAT(13+2*NSEG)=SCY=OSCY  ,                               MAT00560
C             SMAT(14+2*NSEG)=SOY(=OSOY),                               MAT00570
C             SMAT(15+2*NSEG)=SYU=OSYU,                                 MAT00580
C             SMAT(16+2*NSEG)=SUF=OSUF,                                 MAT00590
C             SMAT(17+2*NSEG)=SCU(=OSCU),                               MAT00600
C             SMAT(18+2*NSEG)=SOB(=OSOB),                               MAT00610
C             SMAT(19+2*NSEG)=SOE(=OSOB),                               MAT00620
C             SMAT(20+2*NSEG)=0.                                        MAT00630
C                                                                       MAT00640
      GO TO (100,200,300,400,500),IOPT                                  MAT00650
C                                                                       MAT00660
  100 CONTINUE                                                          MAT00670
C                                                                       MAT00680
C------------------ INPUT DATA FOR SHEAR1 HYSTERESIS MODEL ------------CMAT00690
C                                                                       CMAT00700
C  NSEG    = NUMBER OF BACKBONE SEGEMENTS                               CMAT00710
C  PP      = BACKBONE LOAD                                              CMAT00720
C  DP      = BACKBONE DISPLACEMENT                                      CMAT00730
C-----------------------------------------------------------------------CMAT00740
C FOR THIS MATERIAL, NSEG SHOULD BE 4.                                  MAT00750
      READ (5,*) TYPE ,NSEG,BDAM                                        MAT00760
      READ (5,*) ( SMAT(4+I     ),I=1,NSEG)                             MAT00770
      READ (5,*) ( SMAT(4+I+NSEG),I=1,NSEG)                             MAT00780
      READ (5,*) HO,HS,LWP,AD,AV                                        MAT00790
      LELEM = 80                                                        MAT00800
      LMAT =28                                                          MAT00810
      SMAT(1)=NSEG                                                      MAT00820
C     SMAT(2)=CSE; CSE= CONSTANT STRAIN ENERGY                          MAT00830
      SMAT(2)=SMAT(5)*SMAT(9)/2+(SMAT(5)+SMAT(6))*(SMAT(10)-SMAT(9))/2  MAT00840
C TO CORRECT FAILURE DISPLACEMENT                                       MAT00850
CC    IF(SMAT(8).NE.0.) THEN                                            MAT00860
CC      STI=(SMAT(8)-SMAT(7))/(SMAT(12)-SMAT(11))                       MAT00870
CC      SMAT(12)=SMAT(11)+(0.-SMAT(7))/STI                              MAT00880
CC      SMAT(8)=0.                                                      MAT00890
CC    END IF                                                            MAT00900
C THE ARRAY SMAT IN BETWEEN SEES THE READ STATEMENTS ABOVE.             MAT00910
C  SMAT(4+I     ),I=1,NSEG  AND  SMAT(4+I+NSEG),I=1,NSEG               MAT00920
      SMAT(5+2*NSEG)=BDAM                                               MAT00930
      SMAT(6+2*NSEG)= HO                                                MAT00940
      SMAT(7+2*NSEG)= HS                                                MAT00950
      SMAT(8+2*NSEG)= LWP                                               MAT00960
      SMAT(9+2*NSEG)= AD                                                MAT00970
      SMAT(10+2*NSEG)=AV                                                MAT00980
C                                                                       MAT00990
C                                                                       MAT01000
C-------------- CALC CONSTANT STRAIN ENERGY.....                       MAT01010
C     DL=0                                                              MAT01020
C     PL=0                                                              MAT01030
C     DO 10 I=1,NSEG                                                    MAT01040
C           P=SMAT(4+I)                                                 MAT01050
C           D=SMAT(4+I+NSEG)                                            MAT01060
C           IF (D.LE.DY) THEN                                           MAT01070
C             CSE=CSE + 0.5*(P+PL)*(D-DL)                               MAT01080
C             PL=P                                                      MAT01090
C             DL=D                                                      MAT01100
C           ELSE IF (D.EQ.DL) THEN                                      MAT01110
C             GO TO 20                                                  MAT01120
C           ELSE                                                        MAT01130
C             PY=PL+ (DY-DL)*(P-PL)/(D-DL)                              MAT01140
C             CSE=CSE + 0.5*(PY+PL)*(DY-DL)                             MAT01150
C             GO TO 20                                                  MAT01160
C           ENDIF                                                       MAT01170
```

```
C  10 CONTINUE                                                          MAT01180
C  20 SMAT(2)=CSE                                                       MAT01190
C                                                                       MAT01200
      IF (FIRST .OR. BUG) WRITE (6,65)                                  MAT01210
      WRITE (6,67) MAT,BDAM,(SMAT(4+K),SMAT(4+K+NSEG),K=1,NSEG)         MAT01220
      WRITE (6,*)   '                                                  MAT01230
   65 FORMAT('//' SHEAR2 HYSTERESIS MODEL DATA - ',                     MAT01240
     &       'FOR WHOLE LENGTH MEMBER'                                  MAT01250
     &  /     ' ********************************',                     MAT01260
     &       '**********************',                                 MAT01270
     &      /'      BACKBONE CURVE POINTS - MAT.      ',               MAT01280
     &  5X,'BETA',6X,    3X,' STRESS',8X,' STRAIN')                    MAT01290
   67  FORMAT     ( (28X,I4,2X,3G15.6)/(49X,2G15.6))                    MAT01300
C                                                                       MAT01310
      RETURN                                                            MAT01320
C------------------------- GET STIFFNESS TERMS ------------------------MAT01330
  200 CONTINUE                                                          MAT01340
C           ---- SHEAR1 MODEL DATA STORAGE FOR MEMBER ----             MAT01350
C       SE( 1)= K                                                       MAT01360
C       SE( 2)= LRULE                                                   MAT01370
C       SE( 3)= KKL                                                     MAT01380
C       SE( 4)= DSINC                                                   MAT01390
C       SE( 5)= DMAX                                                    MAT01400
C       SE( 6)= DM                                                      MAT01410
C       SE( 7)= FM                                                      MAT01420
C       SE( 8)= KP                                                      MAT01430
C       SE( 9)= SOC                                                     MAT01440
C       SE(10)= SCY                                                     MAT01450
C       SE(11)= SOY                                                     MAT01460
C       SE(12)= STU                                                     MAT01470
C       SE(13)= SUF                                                     MAT01480
C       SE(14)= SCU                                                     MAT01490
C       SE(15)= SOB                                                     MAT01500
C       SE(16)= SOE                                                     MAT01510
C       SE(17)= FU0(I,I=1,4)                                            MAT01520
C       SE(21)= DU0(I,I=1,4)                                            MAT01530
C       SE(25)= KEYDS2                                                  MAT01540
C       SE(26)= SECNO                                                   MAT01550
C       SE(27)= S2Y                                                     MAT01560
C       SE(28)= DU17                                                    MAT01570
C       SE(29)= FU17                                                    MAT01580
C       SE(30)= AREA1                                                   MAT01590
C       SE(31)= CYCLEN                                                  MAT01600
C       SE(32)= CSE                                                     MAT01610
C       SE(33)= FU00                                                    MAT01620
C       SE(34)= DU00                                                    MAT01630
C       SE(35)= DKP                                                     MAT01640
C       SE(36)= TNPTFG                                                  MAT01650
C       SE(37)= POSTFG                                                  MAT01660
C       SE(38)= CO1                                                     MAT01670
C       SE(39)= RAT                                                     MAT01680
C       SE(40)= RAT1                                                    MAT01690
C       SE(41)= SQ6                                                     MAT01700
C       SE(42)= SR                                                      MAT01710
C       SE(43)= DP20                                                    MAT01720
C       SE(44)= SRFLAG                                                  MAT01730
C --USE    = EQUIVALENT UNLOADING STIFFNESS                             MAT01740
C       SE(45)=USE                                                      MAT01750
C       SE(46)= IOPT2                                                   MAT01760
C --IOPT2    = USED IN HYST14 SUBRTN (DNCKUL,DDISSI SUBRTN IN HYST14)   MAT01770
C       SE(47)= IOPT                                                    MAT01780
C       SE(48)= PSQ6                                                    MAT01790
C       SE(49)= KLAST                                                   MAT01800
C       SE(50)= DPTT                                                    MAT01810
C       SE(51)= DPT                                                     MAT01820
C       SE(52)= EBSE                                                    MAT01830
C       SE(53)=ESE                                                      MAT01840
C       SE(54)= EBE                                                     MAT01850
C       SE(55)=PSE                                                      MAT01860
C       SE(56)= PSE_OLD                                                 MAT01870
C       SE(57)= EXCR(1)                                                 MAT01880
C       SE(58)= EXCR(2)                                                 MAT01890
C       SE(59)= EXCR(3)                                                 MAT01900
C----- SE(60),SE(61),SE(62) SEE DNE SUBRTN                              MAT01910
C       SE(60)= EXCR(4)                                                 MAT01920
C       SE(61)= EXCR(5)                                                 MAT01930
C       SE(62)= EXCR(6)                                                 MAT01940
C      ,SE(63)=FC                                                       MAT01950
C       SE(64)=FY                                                       MAT01960
C       SE(65)=FU                                                       MAT01970
C       SE(66)=FF                                                       MAT01980
C       SE(67)=DC                                                       MAT01990
C       SE(68)=DT                                                       MAT02000
C       SE(69)=DU                                                       MAT02010
C       SE(70)=DF                                                       MAT02020
C       SE(71)=DSINCL                                                   MAT02030
C       SE(72)=UPOS(1)                                                  MAT02040
C       SE(73)=UPOS(2)                                                  MAT02050
C       SE(74)=UPOS(3)                                                  MAT02060
C       SE(75)=UNEG(1)                                                  MAT02070
C       SE(76)=UNEG(2)                                                  MAT02080
C       SE(77)=UNEG(3)                                                  MAT02090
C       SE(78)=LENGTH                                                   MAT02100
C       SE(79)=WIDTH                                                    MAT02110
C       SE(80)= SHEAR SPAN LENGTH RATIO; SMAT(11+2*NSEG) OR MQD         MAT02120
C                                                                       MAT02130
C----- INITIALIZE ---------                                            MAT02140
      SRFLAG=.FALSE.                                                    MAT02150
      POSTFG=.TRUE.                                                     MAT02160
      TNPTFG=.FALSE.                                                    MAT02170
      AREA1=0.                                                          MAT02180
      FP=0.                                                             MAT02190
      DP=0.                                                             MAT02200
      FM=0.                                                             MAT02210
      DM=0.                                                             MAT02220
      FPP=0.                                                            MAT02230
      DPP=0.                                                            MAT02240
      FMAX=0.                                                           MAT02250
      DMAX=0.                                                           MAT02260
      KKL=1                                                             MAT02270
      DU00=0.                                                           MAT02280
      SECNO=0                                                           MAT02290
      CYCLEN=1                                                          MAT02300
      DO 31 L=1,4                                                       MAT02310
   31 FU0(L)=0.                                                         MAT02320
      DO 32 L=1,4                                                       MAT02330
   32 DU0(L)=0.                                                         MAT02340
      SE( 3)=KKL                                                        MAT02350
      SE(31)=CYCLEN                                                     MAT02360
      NSEG=SMAT(1)                                                      MAT02370
      SMAT(3)=SE(78)                                                    MAT02380
      SMAT(4)=SE(79)                                                    MAT02390
      SMAT(11+2*NSEG)=SMAT(7+2*NSEG)/SMAT(4)                            MAT02400
      SE(32)=0.5*SMAT(5)*SMAT(9)+                                       MAT02410
     &       0.5*(SMAT(5)-SMAT(6))*(SMAT(10)-SMAT(9))                   MAT02420
      SE(80)=SMAT(11+2*NSEG)                                            MAT02430
      SE(32)=SMAT(2)                                                    MAT02440
      MQD=SE(80)                                                        MAT02450
C FIND CO1, THIS COEFF. IS OBSERVED FROM EXPERIMENTAL DATA              MAT02460
      CO1=0.3                                                           MAT02470
      SE(38)=CO1                                                        MAT02480
      IF(TNPTFG) SE(36)=1                                               MAT02490
      IF(.NOT.TNPTFG) SE(36)=0                                          MAT02500
      IF(POSTFG) SE(37)=1                                               MAT02510
      IF(.NOT.POSTFG) SE(37)=0                                          MAT02520
      IF(SRFLAG) SE(44)=1                                               MAT02530
      IF(.NOT.SRFLAG) SE(44)=0                                          MAT02540
C                                                                       MAT02550
C     IF(SMAT(28).EQ.0) THEN                                            MAT02560
      CALL OPNSTF(SMAT(5)  ,SMAT(6)  ,SMAT(7)  ,SMAT(8) ,              MAT02570
     &            SMAT(9)  ,SMAT(10) ,SMAT(11) ,SMAT(12),              MAT02580
     &                                SOC      ,SCY     ,              MAT02590
     &                                SOY      ,STU     ,SUF     ,SCU  ,MAT02600
     &                                SOB      ,SOE                   ) MAT02610
C     SMAT(11+2*NSEG)=MQD=HS/WID                                        MAT02620
C     MQD=HS/SE(79)                                                     MAT02630
      SMAT(12+2*NSEG)=SOC                                               MAT02640
      SMAT(13+2*NSEG)=SCY                                               MAT02650
      SMAT(14+2*NSEG)=SOY                                               MAT02660
      SMAT(15+2*NSEG)=STU                                               MAT02670
      SMAT(16+2*NSEG)=SUF                                               MAT02680
      SMAT(17+2*NSEG)=SCU                                               MAT02690
      SMAT(18+2*NSEG)=SOB                                               MAT02700
      SMAT(19+2*NSEG)=SOE                                               MAT02710
      SE(63)=SMAT(5)                                                    MAT02720
      SE(64)=SMAT(6)                                                    MAT02730
      SE(65)=SMAT(7)                                                    MAT02740
      SE(66)=SMAT(8)                                                    MAT02750
      SE(67)=SMAT(9)                                                    MAT02760
      SE(68)=SMAT(10)                                                   MAT02770
      SE(69)=SMAT(11)                                                   MAT02780
      SE(70)=SMAT(12)                                                   MAT02790
      SE( 1)=SOC                                                        MAT02800
      SE(10)=SCY                                                        MAT02810
      SE(11)=SOY                                                        MAT02820
      SE(12)=STU                                                        MAT02830
      SE(13)=SUF                                                        MAT02840
      SE(14)=SCU                                                        MAT02850
      SE(15)=SOB                                                        MAT02860
      SE(16)=SOE                                                        MAT02870
      SE( 9) = SE( 1)                                                   MAT02880
      RETURN                                                            MAT02890
C     END IF                                                            MAT02900
C                                                                       MAT02910
C------------------------- GET STIFFNESS TERMS ------------------------MAT02920
  300 CONTINUE                                                          MAT02930
      IF(SE(36).EQ.1) TNPTFG=.TRUE.                                     MAT02940
      IF(SE(36).EQ.0) TNPTFG=.FALSE.                                    MAT02950
      IF(SE(37).EQ.1) POSTFG=.TRUE.                                     MAT02960
      IF(SE(37).EQ.0) POSTFG=.FALSE.                                    MAT02970
C SRFLAG TO CONTROL KKL =1 OR 3 DEPENDING ON INITIAL DISP. INCREMENT    MAT02980
      IF(SE(44).EQ.1) SRFLAG=.TRUE.                                     MAT02990
      IF(SE(44).EQ.0) SRFLAG=.FALSE.                                    MAT03000
C.  TO SET INITIAL KKL BASED ON INITIAL DISPLACEMENT INCREMENT(DSINC)    MAT03010
C  IF INITIAL DISPL. INCREMENT IS POSITIVE, THEN POSTFG=.TRUE.          MAT03020
      J11=NSEG+5                                                        MAT03030
      J22=NSEG+6                                                        MAT03040
      J33=NSEG+7                                                        MAT03050
      J44=NSEG+8                                                        MAT03060
      J51=5+2*NSEG                                                      MAT03070
      J52=6+2*NSEG                                                      MAT03080
      J53=7+2*NSEG                                                      MAT03090
      J54=8+2*NSEG                                                      MAT03100
      J55=9+2*NSEG                                                      MAT03110
      J56=10+2*NSEG                                                     MAT03120
      J57=11+2*NSEG                                                     MAT03130
      J58=12+2*NSEG                                                     MAT03140
      J59=13+2*NSEG                                                     MAT03150
      J61=14+2*NSEG                                                     MAT03160
      J62=15+2*NSEG                                                     MAT03170
      J63=16+2*NSEG                                                     MAT03180
      J64=17+2*NSEG                                                     MAT03190
      J65=18+2*NSEG                                                     MAT03200
      J66=19+2*NSEG                                                     MAT03210
      KKL=SE(3)                                                         MAT03220
      CYCLEN=SE(31)                                                     MAT03230
      SECNO=SE(26)                                                      MAT03240
      LRULE=SE( 2)                                                      MAT03250
      IOPT2=SE(46)                                                      MAT03260
      IOPT=SE(47)                                                       MAT03270
            CALL HYST14( P, D,PL,DL,     SMAT(3),                     MAT03280
     &          SMAT( 5),SMAT(J11),SMAT( 6),SMAT(J22),SMAT( 7),       MAT03290
     &          SMAT(J33),SMAT( 8),SMAT(J44),SMAT(J51),SMAT(J52),     MAT03300
     &          SMAT(J53),SMAT(J54),SMAT(J55),SMAT(J56),SMAT(J57),    MAT03310
     &          SMAT(J58),SMAT(J59),SMAT(J61),SMAT(J62),SMAT(J63),    MAT03320
     &          SMAT(J64),SMAT(J65),SMAT(J66),                        MAT03330
     &          SE( 1)  ,LRULE  ,KKL    ,SE( 4)  ,SE( 5) ,SE( 6) ,    MAT03340
     &          SE( 7)  ,SE( 8)  ,SE( 9)  ,SE(10)  ,SE(11) ,SE(12) ,  MAT03350
     &          SE(13)  ,SE(14)  ,SE(15)  ,SE(16)  ,SE(17) ,SE(21) ,  MAT03360
     &          SE(25)  ,SECNO   ,SE(27)  ,SE(28)  ,SE(29) ,SE(30) ,  MAT03370
     &          CYCLEN  ,SE(32)  ,SE(33)  ,SE(34)  ,SE(35) ,TNPTFG ,  MAT03380
     &          POSTFG  ,SE(38)  ,SE(39)  ,SE(40)  ,SE(41) ,SE(42) ,  MAT03390
     &          SE(43)  ,SRFLAG  ,SE(45)  ,IOPT2   ,IOPT    ,         MAT03400
     &          SE(48)  ,SE(49)  ,SE(50)  ,SE(51)  ,SE(52) ,SE(55) ,  MAT03410
     &          SE(56)  ,SE(57)  ,SE(58)  ,SE(64)  ,SE(65) ,SE(66) ,  MAT03420
     &          SE(67)  ,SE(68)  ,SE(69)  ,SE(70)  ,SE(71) ,SE(72) ,  MAT03430
     &          SE(75)                                              )  MAT03440
      IF(TNPTFG) SE(36)=1                                               MAT03450
      IF(.NOT.TNPTFG) SE(36)=0                                          MAT03460
      IF(POSTFG) SE(37)=1                                               MAT03470
      IF(.NOT.POSTFG) SE(37)=0                                          MAT03480
      IF(SRFLAG) SE(44)=1                                               MAT03490
      IF(.NOT.SRFLAG) SE(44)=0                                          MAT03500
      SE( 3)=KKL                                                        MAT03510
      SE(31)=CYCLEN                                                     MAT03520
      SE(26)=SECNO                                                      MAT03530
      SE( 2)=LRULE                                                      MAT03540
      SE(46)=IOPT2                                                      MAT03550
      SE(47)=IOPT                                                       MAT03560
      RETURN                                                            MAT03570
C                                                                       MAT03580
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                        MAT03590
C       (NEGLECT DUCTILITIES CALCULATION AT PRESENT STAGE)              MAT03600
  400 EESE=SE(52)                                                       MAT03610
      EPSE=SE(55)                                                       MAT03620
C------ TRANSFER DUCTILITIES AND EXCRUSION RATIOS FOR BOTH IOPT=3 $ 4.  MAT03630
      DO 410 I=1,3                                                      MAT03640
      DUCT(I)=MAX(SE(71+I),SE(74+I))                                    MAT03650
      EXCR(I)=SE(56+I)                                                  MAT03660
  410 EXCR(I+3)=SE(59+I)                                                MAT03670
      RETURN                                                            MAT03680
C                                                                       MAT03690
C---------------------------- DAMAGE INDEX ------------------------------MAT03700
  500 EESE=SE(52)                                                       MAT03710
      EPSE=SE(55)                                                       MAT03720
C----- DDMAX IS THE LARGER OF DMAX OR DM IN HYST14 SUBRTN               MAT03730
      DDMAX=MAX(SE(5),SE(6))                                            MAT03740
      PY=SMAT(6)                                                        MAT03750
C----- DDU :FAILURE DISPLACEMENT OF BACKBONE CURVE                      MAT03760
      DDU=SMAT(12)                                                      MAT03770
      BDAM =SMAT(5+2*NSEG)                                              MAT03780
      DAMAGE=DDMAX/DDU +BDAM/(PY*DDU)*EPSE                              MAT03790
C                                                                       MAT03800
      RETURN                                                            MAT03810
 1111 FORMAT (1P,                                                       MAT03820
     &' S1A- P=',G13.5,'  D=',G13.5,'  K=',G18.9,'  RULE=',G13.5,       MAT03830
     &' DIR=',F6.3,'  A=',G13.5,' IR=',I3)                              MAT03840
 1112 FORMAT (1P,                                                       MAT03850
     &' S1B- P=',G13.5,'  D=',G13.5,'  K=',G18.9,'  RULE=',G13.5,       MAT03860
     &' DIR=',F6.3,'  A=',G13.5,' IR=',I3)                              MAT03870
 1113 FORMAT (1P,                                                       MAT03880
     &' S1C- P=',G13.5,'  D=',G13.5,'  K=',G18.9,'  RULE=',G13.5,       MAT03890
     &' DIR=',F6.3,'  A=',G13.5,' IR=',I3)                              MAT03900
      END                                                               MAT03910
C-------------------------------------------------------------------    MAT03920
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                      MAT03930
C         END DEBUG                                                     MAT03940
C-------------------------------------------------------------------    MAT03950
C ===> TO CALCULATE AVAILABLE STIFFNESS USED IN THE HYSTERESIS LOOP     MAT03960
      SUBROUTINE OPNSTF(FC,FT,FU,FF,DC,DT,DU,DF,                        MAT03970
     &                  SOC,SCY,SOY,STU,SUF,SCU,SOB,SOE)                MAT03980
C                                                                       MAT03990
C COMPUTE STIFFNESS S(OC)                                               MAT04000
      SOC=FC/DC                                                         MAT04010
```

```
C COMPUTE STIFFNESS S(CT)                                                MAT04020
      SCT=(FT-FC)/(DT-DC)                                                MAT04030
C COMPUTE STIFFNESS S(OT)                                                MAT04040
      SOT=FT/DT                                                          MAT04050
C COMPUTE STIFFNESS S(TU)                                                MAT04060
      STU=(FU-FT)/(DU-DT)                                                MAT04070
C COMPUTE STIFFNESS S(UF)                                                MAT04080
      SUF=(FF-FU)/(DF-DU)                                                MAT04090
C COMPUTE STIFFNESS S(CU)                                                MAT04100
      SCU=(FU-FC)/(DU-DC)                                                MAT04110
C COMPUTE STIFFNESS S(OB) ; B IS MIDPT. OF PT. A AND PT. Y               MAT04120
C PT. A :INTERSECTION OF EXTENSION OF OC WITH HORIZONTAL LINE THROU Y    MAT04130
      XA=FY/SOC                                                          MAT04140
      XB=(XA+DY)/2                                                       MAT04150
      SOB=FY/XB                                                          MAT04160
C COMPUTE STIFFNESS S(OE) ; E IS MIDPT. OF PT. D AND PT. U               MAT04170
C PT. D :INTERSECTION OF EXTENSION OF OY WITH HORIZONTAL LINE THROU U    MAT04180
      XD=FU/SOY                                                          MAT04190
      XE=(XD+DU)/2                                                       MAT04200
      SOE=FU/XE                                                          MAT04210
      RETURN                                                             MAT04220
      END                                                                MAT04230
C*********************************************************************** MPO00010
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                       MPO00020
C         END DEBUG                                                      MPO00030
C*********************************************************************** MPO00040
      SUBROUTINE MFORM(IOPT,INMASS,NMASS,MASS,MDM,MD,NDOF,IMD,TITLE,     MPO00050
     &                 NNODE,MAXNOD,NCOS,BUG,IBUG,                       MPO00060
     &                 ID,COSINE,IDOF,CONST,JTFLG,                       MPO00070
     &                 JTCOS,DATA,SMASS,LM,A,S,SAT)                      MPO00080
      DIMENSION ID(MAXNOD),     JTFLG(MAXNOD),   JTCOS(MAXNOD),          MPO00090
     &          IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),CONST(MAXNOD,6)         MPO00100
      DIMENSION A(6,6),S(6,6),SAT(6,6),MDM(NDOF+1),MD(7)                 MPO00110
      REAL     MASS(IMD)                                                 MPO00120
      LOGICAL BTEST                                                      MPO00130
      DIMENSION DATA(12,INMASS) ,SMASS(21) ,LM(6)                        MPO00140
      CHARACTER*130 ELNO                                                 MPO00150
      CHARACTER*(*) TITLE(2)                                             MPO00160
      LOGICAL BUG                                                        MPO00170
C                                                                        MPO00180
C-------- PRINT HEADER AND INPUT MASS DATA                               MPO00190
      IF (IOPT.EQ.1) THEN                                                MPO00200
        WRITE (6,5) TITLE(1),TITLE(2)                                    MPO00210
    5   FORMAT ('1 STRUCTURE.....: ',A,/                                 MPO00220
     &          '  SOLUTION.....: ',A,//                                MPO00230
     &          '  LUMPED NODE MASSES '/                                MPO00240
     &          ' ====================='//                              MPO00250
     &    4X,'NODE',5X,'MX',10X,'MY',10X,'MZ',                          MPO00260
     &       9X,'IXX', 9X,'IYY', 9X,'IZZ',                              MPO00270
     &       9X,'IXY', 9X,'IYZ', 9X,'IYZ',5X,'  IGEN   INC')            MPO00280
C                                                                        MPO00290
       NMASS=1                                                           MPO00300
C  INPUT:  NODE, PX,PY,PZ, RX,RY,RZ, RXY,RXZ,RYZ, IGEN, INC              MPO00310
    6  READ  (5,*) (DATA(I,NMASS),I=1,12)                               MPO00320
       NODE = DATA( 1,NMASS)                                             MPO00330
       IGEN = DATA(11,NMASS)                                             MPO00340
       INC  = DATA(12,NMASS)                                             MPO00350
       IF (NODE.GT.0) THEN                                               MPO00360
           WRITE (6,7) NODE,(DATA(I,NMASS),I=2,10),IGEN,INC              MPO00370
           IF (NMASS.LT.INMASS) THEN                                     MPO00380
             NMASS=NMASS+1                                               MPO00390
             GO TO 6                                                     MPO00400
           ENDIF                                                         MPO00410
       ELSE                                                              MPO00420
         NMASS=NMASS-1                                                   MPO00430
       ENDIF                                                             MPO00440
    7  FORMAT (2X,I6,1P,9G12.4,2I6)                                      MPO00450
C     ELSE IF (IOPT.EQ.2) THEN                                           MPO00460
C                                                                        MPO00470
      ENDIF                                                              MPO00480
C                                                                        MPO00490
      DO 100 IMASS=1,NMASS                                               MPO00500
        NODE=DATA( 1,IMASS)                                              MPO00510
        PX  =DATA( 2,IMASS)                                              MPO00520
        PY  =DATA( 3,IMASS)                                              MPO00530
        PZ  =DATA( 4,IMASS)                                              MPO00540
        RX  =DATA( 5,IMASS)                                              MPO00550
        RY  =DATA( 6,IMASS)                                              MPO00560
        RZ  =DATA( 7,IMASS)                                              MPO00570
        RXY =DATA( 8,IMASS)                                              MPO00580
        RXZ =DATA( 9,IMASS)                                              MPO00590
        RYZ =DATA(10,IMASS)                                              MPO00600
        IGEN=DATA(11,IMASS)                                              MPO00610
        INC =DATA(12,IMASS)                                              MPO00620
C                                                                        MPO00630
    9 JOINT=IQUICK(NODE,ID,NNODE)                                        MPO00640
C                                                                        MPO00650
      IF (JOINT.LE.0) THEN                                               MPO00660
        WRITE (6,*) 'NODE #',NODE,                                       MPO00670
     &  ' WAS NOT FOUND, CHECK INPUT. MASS IS OMITTED '                  MPO00680
        GO TO 90                                                         MPO00690
      ENDIF                                                              MPO00700
C                                                                        MPO00710
      ICOS  =JTCOS(JOINT)                                                MPO00720
C                                                                        MPO00730
C------> SET UP NODE MASS MATRIX IN XYZ COORDINATE SYSTEM                MPO00740
C------- SET UP TRANSFORMATION MATRIX, A = (CONST)                       MPO00750
      DO 10 I=1,6                                                        MPO00760
      DO 10 J=1,6                                                        MPO00770
   10   A(I,J)=0                                                         MPO00780
      DO 20 I=1,6                                                        MPO00790
   20   A(I,I)=1                                                         MPO00800
      A(4,2)=CONST(JOINT,1)                                              MPO00810
      A(4,3)=CONST(JOINT,2)                                              MPO00820
      A(5,1)=CONST(JOINT,3)                                              MPO00830
      A(5,3)=CONST(JOINT,4)                                              MPO00840
      A(6,1)=CONST(JOINT,5)                                              MPO00850
      A(6,2)=CONST(JOINT,6)                                              MPO00860
C                                                                        MPO00870
C------ SET UP NODE MASS MATRIX                                          MPO00880
      DO 30 I=1,6                                                        MPO00890
      DO 30 J=I+1,6                                                      MPO00900
   30   S(I,J)=0                                                         MPO00910
C                                                                        MPO00920
      S(1,1)= PX                                                         MPO00930
      S(2,2)= PY                                                         MPO00940
      S(3,3)= PZ                                                         MPO00950
      S(4,4)= RX                                                         MPO00960
      S(5,5)= RY                                                         MPO00970
      S(6,6)= RZ                                                         MPO00980
      S(4,5)=-RXY                                                        MPO00990
      S(4,6)=-RXZ                                                        MPO01000
      S(5,6)=-RYZ                                                        MPO01010
C                                                                        MPO01020
      DO 40 I=1,6                                                        MPO01030
      DO 40 J=I+1,6                                                      MPO01040
   40   S(J,I)=S(I,J)                                                    MPO01050
C                                                                        MPO01060
C------ TRANSFORM MASS MATRIX TO ACCOUNT FOR CONSTRAINTS                 MPO01070
      DO 50 I=1,6                                                        MPO01080
      DO 50 J=1,6                                                        MPO01090
        SAT(I,J)=0                                                       MPO01100
        DO 50 K=1,6                                                      MPO01110
   50     SAT(I,J)=SAT(I,J) + S(I,K)*A(J,K)                             MPO01120
C                                                                        MPO01130
      DO 60 I=1,6                                                        MPO01140
      DO 60 J=1,6                                                        MPO01150
        S(I,J)=0                                                         MPO01160
        DO 60 K=1,6                                                      MPO01170
   60     S(I,J)=S(I,J) + A(I,K)*SAT(K,J)                               MPO01180
C                                                                        MPO01190
C------ TRANSFER MASS MATRIX TO HALF STORAGE MODE                        MPO01200
          IJ=0                                                           MPO01210
          DO 70 J=1,6                                                    MPO01220
          DO 70 I=J,1,-1                                                 MPO01230
            IJ=IJ+1                                                      MPO01240
   70     SMASS(IJ)=S(I,J)                                               MPO01250
C------ INITIALIZE THE DIAGONAL MASS MATRIX                              MPO01260
          DO 80  I=1,6                                                   MPO01270
   80     LM(I)=IDOF(JOINT,I)                                            MPO01280
C                                                                        MPO01290
C                                                                        MPO01300
      IF (IOPT.EQ.1) THEN                                                MPO01310
C                                                                        MPO01320
C------ CALL SKYLIN TO RESERVE SPACE FOR THIS NODE'S MASS                MPO01330
            IF (BUG) THEN                                                MPO01340
              WRITE (ELNO,915)  NODE,(LM(I),I=1,6)                       MPO01350
              CALL LMATRX(SMASS,MD,6,21,ELNO)                            MPO01360
            ENDIF                                                        MPO01370
          CALL SKYLIN(0,2,NDOF,IBUG,TITLE,MDM,6,                         MPO01380
     &          LM,SMASS,MD,'GLOBAL MASS')                               MPO01390
C                                                                        MPO01400
      ELSE IF (IOPT.EQ.2) THEN                                           MPO01410
            IF (BUG) THEN                                                MPO01420
              WRITE (ELNO,915)  NODE,(LM(I),I=1,6)                       MPO01430
  915         FORMAT ('MASS AT NODE #',I6,' LM=',6I6)                    MPO01440
              CALL LMATRX(SMASS,MD,6,21,ELNO)                            MPO01450
            ENDIF                                                        MPO01460
C                                                                        MPO01470
          CALL FORML(MASS,MDM,NDOF,IMD,SMASS,MD,6,21,LM,1.0 )            MPO01480
C                                                                        MPO01490
      ENDIF                                                              MPO01500
   90 CONTINUE                                                           MPO01510
C                                                                        MPO01520
      IF (IGEN.GT.0) THEN                                                MPO01530
        IGEN=IGEN-1                                                      MPO01540
        NODE=NODE+INC                                                    MPO01550
        GO TO 9                                                          MPO01560
      ENDIF                                                              MPO01570
C                                                                        MPO01580
  100 CONTINUE                                                           MPO01590
C                                                                        MPO01600
      RETURN                                                             MPO01610
      END                                                                MPO01620
#PROCESS SDUMP OPT(3) GOSTMT XREF                                        MSO00010
C*********************************************************************** MSO00020
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                       MSO00030
C         END DEBUG                                                      MSO00040
C*********************************************************************** MSO00050
C  MATRIX SOLUTION OF EQUATION AX=P, WHERE THE UPPER TRIANGULAR MATRIX   MSO00060
C    OF THE SYMMETRIC MATRIX A IS STORED BY SKYLINES. P IS STORED AS     MSO00070
C    A FULL MATRIX.  THE MATRIX P IS REPLACED BY X AFTER BACK            MSO00080
C    SUBSTITUTION IS COMPLEATE.  BOTH A AND P ARE ALTERED.....           MSO00090
C  GUYAN OPTION: IF GUYAN IS .TRUE. MASS MATRIX IS ALSO CONDENSED        MSO00100
C*********************************************************************** MSO00110
      SUBROUTINE MSOLL(IOPT,PRINT,N,IMD,LROW,NLC,L1,L2,MD,A,P,FACTOR,    MSO00120
     & GUYAN ,MASS,MDMASS,IMDMAS,                                        MSO00130
     & KGCOND,KG  ,MDKG  ,IMDKG ,ABORT)                                  MSO00140
      DIMENSION A(IMD),FACTOR(N),P(LROW,NLC),MD(N+1)                     MSO00150
      DIMENSION MDMASS(N+1),MDKG(N+1)                                    MSO00160
      LOGICAL BTEST,ABORT                                                MSO00170
      REAL    MASS(IMDMAS),KG(IMDKG)                                     MSO00180
      LOGICAL GUYAN,PRINT,KGCOND                                         MSO00190
C                                                                        MSO00200
      IJI(I,J)=MD(J)+J-I                                                 MSO00210
      IJM(I,J)=MDMASS(J)+J-I                                             MSO00220
      IJKG(I,J)=MDKG(J)+J-I                                              MSO00230
C------------------------------------------------------------------------MSO00240
C   MATRIX STORAGE SCHEME:                                               MSO00250
C  -THE MATRIX A IS BANDED AND SYMMETRIC. THUS ONLY THE SKYLINE ABOVE    MSO00260
C   AND THE MAIN DIAGONAL NEEDS TO BE STORED.                            MSO00270
C   THE MATRIX A IS STORED IN A LINEAR ARRAY BY COLUMNS.  THE VALUE ON   MSO00280
C   THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST.               MSO00290
C   ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE MAIN DIAGONAL  MSO00300
C   TERMS ARE STORED IN MD.                                              MSO00310
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW...              MSO00320
C            |  1 |  3 |  |  | 12 |                                      MSO00330
C            |    |  2 |  5 |  | 11 |   NOTE: B(1,3),B(1,4) AND B(2,4) ARE ZERO.  MSO00340
C   B=       |    |    |  4 |  7 | 10 |                                  MSO00350
C            |    | SYMM.   |  6 |  9 |   MD= 1, 2, 4, 6, 8, 13          MSO00360
C            |    |         |    |  8 |                                  MSO00370
C   THUS, B(3,3)=A( MD(3) )= A(4)                                        MSO00380
C         B(I,J)=A( MD(J)+J-I ) IF J>I AND (MD(J)+J-I)<MD(J+1)           MSO00390
C               =0              IF J>I AND (MD(J)+J-I)>MD(J+1)           MSO00400
C  -MATRIX P IS NOT SYMMETRIC, AND IS STORED AS A FULL MATRIX            MSO00410
C------------------------------------------------------------------------MSO00420
C   VARIABLE TABLE:                                                      MSO00430
C   IOPT   = OPTION NUMBER                                               MSO00440
C   N      = NUMBER OF DEGREES OF FREEDOM OF A MATRIX                    MSO00450
C   NLC    = NUMBER OF LOAD CASES FOR THE P MATRIX                       MSO00460
C   L1     = FIRST ROW TO BE WORKED WITH                                 MSO00470
C   L2     = LAST ROW TO BE WORKED WITH                                  MSO00480
C   MD     = ADDRESS OF THE MAIN DIAGONAL TERMS OF A                     MSO00490
C   P      = LADD MATRIX ON INPUT, SOLN (X) OF AX=P ON COMPLEATION OF    MSO00500
C            IOPT=4                                                      MSO00510
C   FACTOR= TEMPORARY STORAGE MATRIX                                     MSO00520
C   GUYAN = GUYAN REDUCTION FLAG FOR MASS MATRIX                         MSO00530
C   MASS  = MASS MATRIX                                                  MSO00540
C   MDMASS= MASS MATRIX ADDRESSES OF MAIN DIAGONAL ELEMENTS              MSO00550
C   KGCOND= FLAG FOR CONDENSING OUT KG                                   MSO00560
C   KG    = GEOMETRIC STIFFNESS MATRIX                                   MSO00570
C   MDKG  = KG MATRIX ADDRESSES OF MAIN DIAGONAL ELEMENTS                MSO00580
C   L     = ROW NUMBER FOR ELIMINATION                                   MSO00590
C   I     = ROW NUMBER                                                   MSO00600
C   J     = COLUMN NUMBER                                                MSO00610
C------------------------------------------------------------------------MSO00620
C                                                                        MSO00630
      L1=MAX(L1,1)                                                       MSO00640
      L2=MIN(L2,N)                                                       MSO00650
C                                                                        MSO00660
      GO TO (10,110,210,310),IOPT                                        MSO00670
C                                                                        MSO00680
   10 CONTINUE                                                           MSO00690
C--------- IOPT=1, REDUCE A AND P -----------------------------------    MSO00700
C         GAUSSIAN ELIMINIATION IS USED TO REDUCE THE A AND P MATRICIES  MSO00710
C         FROM ROW L1 TO ROW L2.  IF L1 IS NOT 1, IT IS ASSUMED THAT     MSO00720
C         A AND P ARE ALLREADY REDUCED FROM 1 TO L1...                   MSO00730
C------------------------------------------------------------------      MSO00740
      IF (PRINT) THEN                                                    MSO00750
        WRITE (6,500) 'IOPT=1, REDUCE STIFFNESS AND LOADS',L1,L2,N       MSO00760
        CALL LMATRX(A,MD,N,IMD ,'INPUT STIFFNESS')                       MSO00770
        CALL WMATRX(P,N ,NLC,'INPUT LOAD')                               MSO00780
        IF (GUYAN ) CALL LMATRX(MASS,MDMASS,N,IMDMAS,'INPUT MASS')       MSO00790
        IF (KGCOND) CALL LMATRX(KG  ,MDKG  ,N,IMDKG ,'INPUT KG  ')      MSO00800
C                    WRITE (6,*) 'MD---------'                           MSO00810
C                    WRITE (6,*) (MD(II),II=1,N+1)                       MSO00820
C       IF (GUYAN ) WRITE (6,*) 'MDMASS-----'                            MSO00830
C       IF (GUYAN ) WRITE (6,*) (MDMASS(II),II=1,N+1)                    MSO00840
C       IF (KGCOND) WRITE (6,*) 'MDKG-------'                            MSO00850
C       IF (KGCOND) WRITE (6,*) (MDKG(II),II=1,N+1)                      MSO00860
      ENDIF                                                              MSO00870
C                                                                        MSO00880
      DO 100 L=L1,L2                                                     MSO00890
        IF (L.EQ.N) GO TO 100                                            MSO00900
C                                                                        MSO00910
C------- TEST FOR ZERO PIVOT.....................                       MSO00920
        IF (A(MD(L)).EQ.0) THEN                                          MSO00930
          WRITE (6,2000) L,MD(L),A(MD(L))                                MSO00940
C         CALL ERRTRA                                                    MSO00950
C         STOP '*** ABNORMAL END IN SUBROUTINE MSOLL ***'                MSO00960
          ABORT=.TRUE.                                                   MSO00970
          RETURN                                                         MSO00980
        ENDIF                                                            MSO00990
C                                                                        MSO01000
```

```
C------- CALCULATE THE FACTORS EACH ROW IS MULTIPLIED BY
         DO 30 I=(L+1),N
           LI=IJI(L,I)
           IF ( LI.LT.MD(I+1) ) THEN
                FACTOR(I)=A(LI) /A( MD(L) )
           ELSE
                FACTOR(I)=0
           ENDIF
   30    CONTINUE
C
C------- REDUCE A BY COLUMNS (J) EACH ROW (I)...
         DO 40 J=(L+1),N
           LJ=IJI(L,J)
           IF (LJ.GE.MD(J+1)) GO TO 40
           DO 35 I=(L+1),J
             IJ=IJI(I,J)
             IF (IJ.LT.MD(J+1)) A(IJ)=A(IJ)-FACTOR(I)*A(LJ)
C            A(IJ)=A(IJ)-FACTOR(I)*A(LJ)
   35        CONTINUE
   40    CONTINUE
C
C------- REDUCE P BY COLUMNS (J) EACH ROW (I)...
         DO 50 J=1,NLC
           IF (P(L,J).EQ.0) GO TO 50
           DO 45 I=(L+1),N
             P(I,J)=P(I,J)-FACTOR(I)*P(L,J)
   45    CONTINUE
   50    CONTINUE
C
C------- REDUCE MASS MATRIX BY GUYAN REDUCTION
      IF (GUYAN) THEN
C
C......... M22=M22 - K21/K11 * M11 * 1/K11 * K12
         RM=MASS(MDMASS(L))
         IF (RM.NE.0) THEN
              DO 60 J=L+1,N
                   IF (FACTOR(J).EQ.0) GO TO 60
                   I1=MAX(L+1,J-(MDMASS(J+1)-MDMASS(J)) )
                   DO 55 I=I1,J
                        IJ=IJM(I,J)
                        MASS(IJ)=MASS(IJ)+FACTOR(I)*RM*FACTOR(J)
   55              CONTINUE
   60         CONTINUE
         ENDIF
C
C......... M22=M22 - K21/K11 * M12
         DO 66 J=L+1,N
              LJ=IJM(L,J)
              IF (LJ.GE.MDMASS(J+1)) GO TO 66
              IF (MASS(LJ).EQ.0) GO TO 66
              DO 65 I=L+1,J
                   IJ=IJM(I,J)
                   IF (IJ.LT.MDMASS(J+1))
     &            MASS(IJ)=MASS(IJ)-FACTOR(I)*MASS(LJ)
   65              CONTINUE
   66    CONTINUE
C
C......... M22=M22 - M21 * 1/K11 * K12
         DO 70 J=L+1,N
              IF (FACTOR(J).EQ.0) GO TO 70
              DO 69 I=L+1,J
                   LI=IJM(L,I)
                   IF (LI.GE.MDMASS(I+1)) GO TO 69
                   IJ=IJM(I,J)
                   IF (IJ.LT.MDMASS(J+1))
     &             MASS(IJ)=MASS(IJ)-FACTOR(J)*MASS(LI)
   69              CONTINUE
   70    CONTINUE
      ENDIF
C
C------- REDUCE GEOMETRIC STIFFNESS MATRIX
      IF (KGCOND) THEN
C
C......... KG22=KG22 + K21/K11 * KG11 * 1/K11 * KG12
         RKG=KG(MDKG(L))
         IF (RKG.NE.0) THEN
              DO 80 J=L+1,N
                   IF (FACTOR(J).EQ.0) GO TO 80
                   I1=MAX(L+1,J-(MDKG(J+1)-MDKG(J)) )
                   DO 75 I=I1,J
                        IJ=IJKG(I,J)
                        KG(IJ)=KG(IJ)+FACTOR(I)*RKG*FACTOR(J)
   75              CONTINUE
   80         CONTINUE
         ENDIF
C
C......... KG22=KG22 - K21/K11 * KG12
         DO 86 J=L+1,N
              LJ=IJKG(L,J)
              IF (LJ.GE.MDKG(J+1)) GO TO 86
              IF (KG(LJ).EQ.0) GO TO 86
              DO 85 I=L+1,J
                   IJ=IJKG(I,J)
                   IF (IJ.LT.MDKG(J+1))
     &             KG(IJ)=KG(IJ)-FACTOR(I)*KG(LJ)
   85              CONTINUE
   86    CONTINUE
C
C......... KG22=KG22 - KG21 * 1/K11 * K12
         DO 90 J=L+1,N
              IF (FACTOR(J).EQ.0) GO TO 90
              DO 89 I=L+1,J
                   LI=IJKG(L,I)
                   IF (LI.GE.MDKG(I+1)) GO TO 89
                   IJ=IJKG(I,J)
                   IF (IJ.LT.MDKG(J+1))
     &             KG(IJ)=KG(IJ)-FACTOR(J)*KG(LI)
   89              CONTINUE
   90    CONTINUE
C
      ENDIF
C
C
  100 CONTINUE
      IF (PRINT) THEN
        CALL LMATRX(A,MD,N,1MD,'REDUCED STIFFNESS')
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')
        IF (GUYAN)  CALL LMATRX(MASS,MDMASS,N,IMDMAS,'REDUCED MASS')
C       CALL LMATRX(MASS,MDMASS,N,IMDMAS,'REDUCED MASS')
        IF (KGCOND) CALL LMATRX(KG  ,MDKG  ,N,IMDKG ,'REDUCED KG')
      ENDIF
      RETURN
C
  110 CONTINUE
C--------- IOPT=2, REDUCE P ONLY ------------------------------------------
C          GAUSSIAN ELIMINIATION IS USED TO REDUCE THE P MATRIX
C          FROM ROW L1 TO ROW L2.  IF L1 IS NOT 1, IT IS ASSUMED THAT
C          A AND P ARE ALLREADY REDUCED FROM 1 TO L1...
C-------------------------------------------------------------------------
      IF (PRINT) THEN
        WRITE (6,500) 'IOPT=2, REDUCE LOADS ONLY',L1,L2,N
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')
        CALL WMATRX(P,N ,NLC,'NEW LOAD MATRIX ')
      ENDIF
C
C------- LOOP FOR EACH ROW (L) TO REDUCE....
      DO 160 L=L1,L2
C
      IF (A(MD(L)).EQ.0) THEN
        WRITE (6,2000) L,MD(L),A(MD(L))
C       CALL ERRTRA
        ABORT=.TRUE.
        RETURN
      ENDIF
C
C       LOOP FOR EACH COLUMN (J)
        DO 150 J=1,NLC
          IF (P(L,J).EQ.0) GO TO 150
          DO 120 I=(L+1),N
            LI=IJI(L,I)
            IF (LI.LT.MD(I+1)) P(I,J)=P(I,J)-A(LI)*P(L,J)/A(MD(L))
  120       CONTINUE
  150   CONTINUE
  160   CONTINUE
C     CALL WMATRX(P,N ,NLC,'REDUCED LOAD')
      RETURN
C
  210 CONTINUE
C--------- IOPT=3, BACK SUBSTUTION -----------------------------------------
C          BACK SUBSTUTION IS USED TO SOLVE FOR X IN SX=V, WHERE S IS
C          THE REDUCE UPPER TRIANGULAR FACTOR OF THE A MATRIX, AND V
C          IS THE REDUCED FORRM OF THE P MATRIX.  THE RESULTS (X) ARE
C          STORED IN P.
C          BACK SUBSTUTION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.
C-------------------------------------------------------------------------
      IF (PRINT) THEN
        WRITE (6,500) 'IOPT=3, BACK SUBSTUTION',L1,L2,N
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')
      ENDIF
C
C------ LOOP FOR EACH ROW L
C     DO 250 L=L2,L1,-1                       /*ORIGINAL
C                                             /*ORIGINAL
C       IF (A(MD(L)).EQ.0) THEN               /*ORIGINAL
C         WRITE (6,2000) L,MD(L),A(MD(L))     /*ORIGINAL
C         CALL ERRTRA                         /*ORIGINAL
C         ABORT=.TRUE.                        /*ORIGINAL
C         RETURN                              /*ORIGINAL
C       ENDIF                                 /*ORIGINAL
C
C       CALCULAGE X FOR EACH LOAD CASE, STORE IN P
        DO 250 J=L2,L1,-1
        IF (A(MD(J)).EQ.0) THEN
          WRITE (6,2000) J,MD(J),A(MD(J))
          CALL ERRTRA
          ABORT=.TRUE.
          RETURN
        ENDIF
        DO 215 LC=1,NLC
  215     P(J,LC)=P(J,LC)/A(MD(J))
C
        IF (J.GT.1) THEN
         IJ=MD(J+1)
         NI=J-(MD(J+1)-MD(J)) +1
C          WRITE (6,*) 'NI,IJ',NI,IJ
         DO 220 I=NI,J-1,+1
           IJ=IJ-1
           DO 220 LC=1,NLC
C            WRITE (6,*) 'I=',I,'J=',J,'LC=',LC,'IJ=',IJ
C            WRITE (6,*) 'P(I,LC)=',P(I,LC)
C            WRITE (6,*) 'A(IJ)  =',A(IJ)
             P(I,LC)=P(I,LC)-P(J,LC)*A(IJ)
C            WRITE (6,*) 'Q(I,LC)=',P(I,LC)
  220  CONTINUE
       ENDIF
  250  CONTINUE
C
C       DO 215 J=1,NLC                  /*
C 215     P(L,J)=P(L,J)/A(MD(L))        /*
C
C       MODIFY THE REMAINDER OF V       /*
C       DO 220 I=1,(L-1)                /*
C       DO 220 J=1,NLC                  /*
C         IL=IJI(I,L)                   /*
C         IF (IL.LT.MD(L+1)) P(I,J)=P(I,J)-A(IL)*P(L,J)     /*
C 220     CONTINUE                      /*
C 250   CONTINUE                        /*
      IF (PRINT) CALL WMATRX(P,N ,NLC,'DISPLACEMENT')
C
      RETURN
C
  310 CONTINUE
C--------- IOPT=4, REDUCED MULTIPLICATION ----------------------------------
C          SOLVE FOR V IN S*D=P WHERE S HAS BEEN REDUCED BY GAUSSIAN
C          ELIMINATION.
C          THE RESULTING LOAD IS STORED IN P.
C          THE DISPLACEMENTS ARE INPUT IN D.
C          MULTIPLACATION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.
C-------------------------------------------------------------------------
      IF (PRINT) THEN
        WRITE (6,500) 'IOPT=4, REDUCED MULTIPLACATION',L1,L2,N
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')
        CALL WMATRX(FACTOR,N ,1,'DISPLACEMENT')
      ENDIF
C
C------ ONLY ONE LOAD CASE....
        LC=1
C       ---- ZERO P ----
        DO 320 J=L1,L2
          IF (A(MD(J)).EQ.0) THEN
            WRITE (6,2000) J,MD(J),A(MD(J))
C           CALL ERRTRA
            ABORT=.TRUE.
            RETURN
          ENDIF
  320     P(J,LC)=0
C
C------------ MULTIPLY S_RED * DISP = P
        DO 340 J=L1,N
          IC1=J-(MD(J+1)-MD(J)) +1
          IC2=MAX(IC1,L1)
          DO 340 I=J,IC2,-1
            IJ=IJI(I,J)
  340     P(I,LC)=P(I,LC)+A(IJ)*FACTOR(J)
      IF (PRINT) CALL WMATRX(P,N ,1,'REDUCED LOAD')
C
C------- UNREDUCE THE LOAD MATRIX ...
        DO 360 L=L2,L1,-1
C
C       LOOP FOR EACH COLUMN (J)
        DO 350 J=1,NLC
          IF (P(L,LC).EQ.0) GO TO 350
          DO 345 I=(L+1),N
            LI=IJI(L,I)
            IF (LI.LT.MD(I+1)) P(I,LC)=P(I,LC)+A(LI)*P(L,LC)/A(MD(L))
  345       CONTINUE
  350   CONTINUE
  360   CONTINUE
C
  400   CONTINUE
C
      IF (PRINT) CALL WMATRX(P,N ,1,'LOAD')
C
      RETURN
  500 FORMAT (/2X,A,2X,'L1=',I5,'   L2=',I5,'   N=',I5)
 2000     FORMAT (///1X,50('*')/
     &            ' *  ER','ROR IN SUBROUTINE MSOLL',       T51,'*'/
     &            ' *  PIVOT IS LE 0 ------- ROW=',I5,     T51,'*'/
     &            ' *                STORAGE LCN=',I5,     T51,'*'/
     &            ' *                      PIVOT=',1P,G15.6,T51,'*'/
     &            ' *  SOLUTION IS ABORTED ',              T51,'*'/
     &            1X,50('*')/)
      END
@PROCESS GOSTMT XREF
C-------------------------------------------------------------------------
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT.SUBTRACE
```

```
C          END DEBUG                                                    NOD00040
      SUBROUTINE NODDOF(NDOF,NCOND,NFREE,NREST,MAXNOD,NNODE,NCOS,       NOD00050
     &     SCALE,NSUPT,NCONST,                                          NOD00060
     &     ID,IDOF,INDEX,COORD,COSINE,CONST,                            NOD00070
     &     JTFLG,JTCOS,BUG,MSTRJT)                                      NOD00080
C                                                                       NOD00090
      DIMENSION COORD(MAXNOD,3),ID(MAXNOD),INDEX(MAXNOD),IFM(6),JFM(6), NOD00100
     &          IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),CONST(MAXNOD,6),       NOD00110
     &          JTFLG(MAXNOD),JTCOS(MAXNOD),MSTRJT(MAXNOD,6)            NOD00120
      DIMENSION VX(3),VY(3),VZ(3)                                       NOD00130
      LOGICAL FLAG,PT,BUG,TEST,BTEST                                    NOD00140
      CHARACTER*15 TYPE                                                 NOD00150
      CHARACTER*2  CX(6)                                                NOD00160
C-----------------------------------------------------------------------NOD00170
C VARIABLE TABLE...                                                     NOD00180
C COORD  = JOINT COORDINATES  X,Y,Z                                     NOD00190
C ID     = JOINT NUMBER                                                 NOD00200
C IDOF   = GLOBAL DEGREES OF FREEDOM ASSOC. WITH JOINT FX,FY,FZ,MX,MY,MZNOD00210
C JTCOS  = ROW IN COSINE MATRIX ASSOC WITH JOINT                        NOD00220
C COSINE = COSINE MATRIX                                                NOD00230
C CONST  = GLOBAL CONSTRAINT MATRIX FOR EACH NODE                       NOD00240
C JTFLG  = CONTAINS JOINT FLAG'S--- MUST BE 24 BIT'S LONG.....          NOD00250
C          BIT'S  0 TO  5 RESTRAIN   DOF FX,FY,FZ,MX,MY,MZ              NOD00260
C          BIT'S  6 TO 11 CONDENSE   DOF FX,FY,FZ,MX,MY,MZ              NOD00270
C          BIT'S 12 TO 17 CONSTRAIN  DOF FX,FY,FZ,MX,MY,MZ              NOD00280
C          BIT'S 18 TO 23 ELIMINATE  DOF FX,FY,FZ,MX,MY,MZ              NOD00290
C MSTRJT = INTERNAL MASTER JT # OF CONSTRAINED JOINTS.....              NOD00300
C-----------------------------------------------------------------------NOD00310
C                                                                       NOD00320
C-------- INPUT NODES                                                   NOD00330
      IF (BUG) WRITE (6,1000) 'INPUT NODES'                             NOD00340
      I=0                                                               NOD00350
   20 I=I+1                                                             NOD00360
      READ (5, *) ID(I),(COORD(I,J),J=1,3),JTCOS(I),IGEN                NOD00370
      IF (ID(I).LE.0) THEN                                              NOD00380
         NNODE=I-1                                                      NOD00390
         GO TO 26                                                       NOD00400
      ENDIF                                                             NOD00410
      IF (BUG) WRITE(6,30) I,ID(I),(COORD(I,J),J=1,3),JTCOS(I)          NOD00420
      IF (IGEN.GT.0) THEN                                               NOD00430
           READ (5, *) IDINC,COORD1,COORD2,COORD3                       NOD00440
           DO 25 K=1,IGEN                                               NOD00450
              I=I+1                                                     NOD00460
              ID(I)=ID(I-1)+IDINC                                       NOD00470
              IF (ID(I).LE.0) THEN                                      NOD00480
                 NNODE=I-1                                              NOD00490
                 GO TO 26                                               NOD00500
              ENDIF                                                     NOD00510
              COORD(I,1)=COORD(I-1,1)+COORD1                            NOD00520
              COORD(I,2)=COORD(I-1,2)+COORD2                            NOD00530
              COORD(I,3)=COORD(I-1,3)+COORD3                            NOD00540
              JTCOS(I)=JTCOS(I-1)                                       NOD00550
   25         IF (BUG) WRITE(6,30) I,ID(I),(COORD(I,J),J=1,3),JTCOS(I)NOD00560
      ENDIF                                                             NOD00570
      IF (I.LT.NNODE) GO TO 20                                          NOD00580
   26 CONTINUE                                                          NOD00590
C                                                                       NOD00600
C-------- SORT NODES                                                    NOD00610
      CALL ISORT(ID,INDEX,NNODE)                                        NOD00620
C                                                                       NOD00630
C         MOVE NODE INFO. TO TEMPORARY STORGAE                          NOD00640
      DO 40 I=1,NNODE                                                   NOD00650
       J=INDEX(I)                                                       NOD00660
       IDOF(I,1)=ID(J)                                                  NOD00670
C      IDOF(I,2)=JTCOS(I)                                               NOD00680
C---------------------------------------------------------             NOD00690
C== CORRECT BY GER JENG-FUN JTCOS(I) REPLACED BY JTCOS(J) ==            NOD00700
C---------------------------------------------------------             NOD00710
       IDOF(I,2)=JTCOS(J)                                               NOD00720
       DO 40 K=1,3                                                      NOD00730
   40     CONST(I,K)=COORD(J,K)*SCALE                                   NOD00740
C                                                                       NOD00750
C         MOVE NODES INFO BACK TO MAIN STORAGE, DELETE DUPLICATE INFO.  NOD00760
      NNOLD=NNODE                                                       NOD00770
      I=0                                                               NOD00780
      J=0                                                               NOD00790
   43 I=I+1                                                             NOD00800
      J=J+1                                                             NOD00810
       ID(I)=IDOF(J,1)                                                  NOD00820
       IF (I.GE.2) THEN                                                 NOD00830
         IF (ID(I).EQ.ID(I-1) ) THEN                                    NOD00840
           I=I-1                                                        NOD00850
           NNODE=NNODE-1                                                NOD00860
         ENDIF                                                          NOD00870
       ENDIF                                                            NOD00880
       JTCOS(I)=IDOF(J,2)                                               NOD00890
       DO 44 K=1,3                                                      NOD00900
   44     COORD(I,K)=CONST(J,K)                                         NOD00910
      IF (I.LT.NNODE) GO TO 43                                          NOD00920
C                                                                       NOD00930
   30 FORMAT(' NODE(',I3,'):',I4,'  X:',F10.4,'  Y:',F10.4,'  Z:',F10.4,NOD00940
     &     ',  COS#:',I3)                                               NOD00950
C                                                                       NOD00960
C------- GLOBAL JOINT COSINE MATRIX                                     NOD00970
      IF (BUG)  WRITE (6,1000) 'DIRECTION COSINES ... '                 NOD00980
      DO 55 ICOS=1,NCOS                                                 NOD00990
         READ (5,*) VX,VY                                               NOD01000
         CALL CROSS(VZ,VX,VY,0)                                         NOD01010
         CALL CROSS(VY,VZ,VX,1)                                         NOD01020
         DO 50 J=1,3                                                    NOD01030
            COSINE(1,J,ICOS)=VX(J)                                      NOD01040
            COSINE(2,J,ICOS)=VY(J)                                      NOD01050
   50       COSINE(3,J,ICOS)=VZ(J)                                     NOD01060
      IF (BUG) WRITE (6,56) ICOS,' VX:',VX,' VY:',VY,' VZ:',VZ          NOD01070
   55 CONTINUE                                                          NOD01080
   56 FORMAT(' COS(',I3,') ',                                           NOD01090
     &     3(A,5E,F9.5,' I',SP,F9.5,' J',F9.5,' K  '),S6)               NOD01100
C                                                                       NOD01110
C------- INPUT RESTRAINTS                                               NOD01120
C         ZERO IDOF MATRIX WHICH WILL HOLD RESTRAINT, CONSTRAINT AND    NOD01130
C         ULTIMATLY DOF INFORMATION                                     NOD01140
      DO 59 I=1,NNODE                                                   NOD01150
   59    JTFLG(I)=0                                                     NOD01160
      I=0                                                               NOD01170
      FLAG=.TRUE.                                                       NOD01180
   60 I=I+1                                                             NOD01190
      IF (I.LE.NSUPT) THEN                                              NOD01200
            IF (FLAG.AND.BUG) WRITE (6,1000) ' NODE RESTRAINTS '        NOD01210
            FLAG=.FALSE.                                                NOD01220
            READ (5, *) NODE,IFM,IGEN,NODINC                            NOD01230
   65       IF (NODE.EQ.0) THEN                                         NOD01240
               DO 70 J=1,NNODE                                          NOD01250
                  DO 68 II=1,6                                          NOD01260
                     IF (IFM(II).NE.0) JTFLG(J)=IBSET(JTFLG(J),II-1)    NOD01270
                     IF (IFM(II).EQ.2) JTFLG(J)=IBSET(JTFLG(J),II+17)   NOD01280
   68                CONTINUE                                           NOD01290
                  IF (BUG) WRITE (6,80) ID(J),IFM                       NOD01300
   70          CONTINUE                                                 NOD01310
            ELSE                                                        NOD01320
               J =IQUICK(NODE,ID,NNODE)                                 NOD01330
               IF (J.EQ.0) THEN                                         NOD01340
                  IF (BUG) WRITE (6, 75) NODE                           NOD01350
               ELSE                                                     NOD01360
                  DO 71 II=1,6                                          NOD01370
                     IF (IFM(II).NE.0) JTFLG(J)=IBSET(JTFLG(J),II-1)    NOD01380
                     IF (IFM(II).EQ.2) JTFLG(J)=IBSET(JTFLG(J),II+17)   NOD01390
   71                CONTINUE                                           NOD01400
                  IF (BUG) WRITE (6,80) ID(J),IFM                       NOD01410
               ENDIF                                                    NOD01420
               IF (IGEN.GT.0) THEN                                      NOD01430
                  NODE=NODE+NODINC                                      NOD01440
                  IGEN=IGEN-1                                           NOD01450
                  GO TO 65                                              NOD01460
               ENDIF                                                    NOD01470
            ENDIF                                                       NOD01480
            GO TO 60                                                    NOD01490
      ENDIF                                                             NOD01500
   75 FORMAT (' NODE#',I5,' NOT FOUND, RESTRAINT IGNORED...')           NOD01510
   80 FORMAT (' NODE(',I4,')   FX:',I2,'  FY:',I2,'  FZ:',I2,           NOD01520
     &     '  MX:',I2,'  MY:',I2,'  MZ:',I2)                           NOD01530
C     CALL PBIT(NNODE,JTFLG)                                            NOD01540
C                                                                       NOD01550
C-------- INPUT NODE DOF'S TO BE CONDENSED OUT...                       NOD01560
      I=0                                                               NOD01570
      FLAG=.TRUE.                                                       NOD01580
   90 I=I+1                                                             NOD01590
      IF (I.LE.NCOND) THEN                                              NOD01600
            IF (FLAG.AND.BUG)                                           NOD01610
     &     WRITE (6,1000) ' NODE DOF''S TO BE CONDENSED OUT'            NOD01620
            FLAG=.FALSE.                                                NOD01630
            READ (5, *) NODE,IFM, IGEN,NODINC                           NOD01640
   95       IF (NODE.EQ.0) THEN                                         NOD01650
               DO 100 J=1,NNODE                                         NOD01660
                  PT=.FALSE.                                            NOD01670
                  DO 99 II=1,6                                          NOD01680
                    IF (IFM(II).NE.0.AND. .NOT.BTEST(JTFLG(J),II-1))THENNOD01690
                       JTFLG(J)=IBSET(JTFLG(J),II+5)                    NOD01700
                       JFM(II)=IFM(II)                                  NOD01710
                       PT=.TRUE.                                        NOD01720
                    ELSE                                                NOD01730
                       JFM(II)=0                                        NOD01740
                    ENDIF                                               NOD01750
   99               CONTINUE                                            NOD01760
  100               IF (PT.AND.BUG) WRITE (6,80) ID(J),JFM              NOD01770
               ELSE                                                     NOD01780
                  J =IQUICK(NODE,ID,NNODE)                              NOD01790
                  IF (I.EQ.0) THEN                                      NOD01800
                    IF (BUG) WRITE (6,110) NODE                         NOD01810
                  ELSE                                                  NOD01820
                    DO 101 II=1,6                                       NOD01830
                      IF (IFM(II).NE.0.AND. .NOT.BTEST(JTFLG(J),II-1))THENNOD01840
                         JTFLG(J)=IBSET(JTFLG(J),II+5)                  NOD01850
                         JFM(II)=IFM(II)                                NOD01860
                      ELSE                                              NOD01870
                         JFM(II)=0                                      NOD01880
                      ENDIF                                             NOD01890
  101                 CONTINUE                                          NOD01900
                    IF (PT.AND.BUG) WRITE (6,80) ID(J),JFM              NOD01910
                  ENDIF                                                 NOD01920
                  IF (IGEN.GT.0) THEN                                   NOD01930
                     NODE=NODE+NODINC                                   NOD01940
                     IGEN=IGEN-1                                        NOD01950
                     GO TO 95                                           NOD01960
                  ENDIF                                                 NOD01970
               ENDIF                                                    NOD01980
               GO TO 90                                                 NOD01990
      ENDIF                                                             NOD02000
  110    FORMAT (' NODE#',I5,' NOT FOUND, DOF ARE NOT CONDENSED...')    NOD02010
C     CALL PBIT(NNODE,JTFLG)                                            NOD02020
C                                                                       NOD02030
C                                                                       NOD02040
C-------- INPUT CONSTRAINTS...                                          NOD02050
      I=0                                                               NOD02060
      FLAG=.TRUE.                                                       NOD02070
  400 I=I+1                                                             NOD02080
      IF (I.LE.NCONST) THEN                                             NOD02090
            IF (FLAG) WRITE (6,1000) ' NODE CONSTRAINTS '               NOD02100
            FLAG=.FALSE.                                                NOD02110
            READ (5, *) ITYPE,MASTER,NODE,IGEN,NODINC                   NOD02120
  410       J =IQUICK(NODE,ID,NNODE)                                    NOD02130
            IF (J.EQ.0) THEN                                            NOD02140
              WRITE (6,420) NODE                                        NOD02150
            ELSE                                                        NOD02160
              IF (ITYPE.EQ.0) THEN                                      NOD02170
                 TYPE='3D-RIGID BODY'                                   NOD02180
                 DO 415 K=1,6                                           NOD02190
                   IF (.NOT.BTEST(JTFLG(J),K-1)) THEN                   NOD02200
                        JTFLG(J)=IBSET(JTFLG(J),K+11)                   NOD02210
                        MSTRJT(J,K)=IQUICK(MASTER,ID,NNODE)             NOD02220
                   ELSE                                                 NOD02230
                        WRITE(6,430) NODE,K                             NOD02240
                   ENDIF                                                NOD02250
  415              CONTINUE                                             NOD02260
              ELSE IF (ITYPE.EQ.1) THEN                                 NOD02270
                 TYPE='XY-PLANE'                                        NOD02280
                 K=1                                                    NOD02290
                   IF (.NOT.BTEST(JTFLG(J),K-1)) THEN                   NOD02300
                        JTFLG(J)=IBSET(JTFLG(J),K+11)                   NOD02310
                        MSTRJT(J,K)=IQUICK(MASTER,ID,NNODE)             NOD02320
                   ELSE                                                 NOD02330
                        WRITE(6,430) NODE,K                             NOD02340
                   ENDIF                                                NOD02350
                 K=2                                                    NOD02360
                   IF (.NOT.BTEST(JTFLG(J),K-1)) THEN                   NOD02370
                        JTFLG(J)=IBSET(JTFLG(J),K+11)                   NOD02380
                        MSTRJT(J,K)=IQUICK(MASTER,ID,NNODE)             NOD02390
                   ELSE                                                 NOD02400
                        WRITE(6,430) NODE,K                             NOD02410
                   ENDIF                                                NOD02420
                 K=6                                                    NOD02430
                   IF (.NOT.BTEST(JTFLG(J),K-1)) THEN                   NOD02440
                        JTFLG(J)=IBSET(JTFLG(J),K+11)                   NOD02450
                        MSTRJT(J,K)=IQUICK(MASTER,ID,NNODE)             NOD02460
                   ELSE                                                 NOD02470
                        WRITE(6,430) NODE,K                             NOD02480
                   ENDIF                                                NOD02490
              ELSE                                                      NOD02500
                 WRITE (6,440) ITYPE,MASTER,NODE                        NOD02510
              ENDIF                                                     NOD02520
              WRITE (6,450) TYPE,MASTER,NODE                            NOD02530
              ENDIF                                                     NOD02540
              IF (IGEN.GT.0) THEN                                       NOD02550
                 NODE=NODE+NODINC                                       NOD02560
                 IGEN=IGEN-1                                            NOD02570
                 GO TO 410                                              NOD02580
              ENDIF                                                     NOD02590
            GO TO 400                                                   NOD02600
      ENDIF                                                             NOD02610
  420 FORMAT (' NODE:',I6,' IS NOT FOUND, CONSTRAINT IS IGNORED')       NOD02620
  430 FORMAT (' NODE:',I6,' IS RESTRAINED. CONSTRAINT',I2,              NOD02630
     &     ' IS IGNORED...')                                            NOD02640
  440 FORMAT (' INVALID CONSTRAINT TYPE',I3,                            NOD02650
     &        ' MASTER:',I6,' SLAVE:',I6)                               NOD02660
  450 FORMAT (3X,A,'CONSTRAINT,',                                       NOD02670
     &        ' MASTER:',I6,' SLAVE:',I6)                               NOD02680
C                                                                       NOD02690
      IF (BUG) WRITE (6,1000) 'JOINT RESTRAINT CODES...'                NOD02700
      IF (BUG) CALL PBIT(NNODE,JTFLG,ID)                                NOD02710
C                                                                       NOD02720
C-----------------------------------------                              NOD02730
C--- DETERMINE THE DEGREES OF FREEDOM...   ---                          NOD02740
C-----------------------------------------                              NOD02750
C........ZERO CONSTRAINT MATRIX WHICH WILL HOLD THE MOMENT              NOD02760
C........TRANSFORMATION FOR CONSTRAINED DOF                             NOD02770
      DO 499 I=1,NNODE                                                  NOD02780
        DO 499 J=1,6                                                    NOD02790
  499     CONST(I,J)=0                                                  NOD02800
      NDOF=0                                                            NOD02810
C                                                                       NOD02820
C........DEGREES OF FREEDOM TO BE CONDENSED OUT                         NOD02830
       DO 500 I=1,NNODE                                                 NOD02840
        DO 500 J=1,6                                                    NOD02850
          JT=JTFLG(I)                                                   NOD02860
          IF (BTEST(JT,J+5) .AND. .NOT.BTEST(JT,J+11)) THEN             NOD02870
```

```
              NDOF=NDOF+1                                                NOD02880
              IDOF(I,J)=NDOF                                             NOD02890
            ENDIF                                                        NOD02900
  500 CONTINUE                                                           NOD02910
      NCOND=NDOF                                                         NOD02920
C                                                                        NOD02930
C........UNCONSTRAINED AND UNRESTRAINED DEGREES OF FREEDOM               NOD02940
      DO 510 I=1,NNODE                                                   NOD02950
       DO 510 J=1,6                                                      NOD02960
         NODE=IDOF(I,J)                                                  NOD02970
         IF (.NOT.(BTEST(JTFLG(I),J-1) .OR. BTEST(JTFLG(I),J+5) .OR.     NOD02980
     &       BTEST(JTFLG(I),J+11)) ) THEN                                NOD02990
C          IF (IDOF(I,J).EQ. 0) THEN                                     NOD03000
              NDOF=NDOF+1                                                NOD03010
              IDOF(I,J)=NDOF                                             NOD03020
            ENDIF                                                        NOD03030
  510 CONTINUE                                                           NOD03040
      NFREE=NDOF-NCOND                                                   NOD03050
C                                                                        NOD03060
C........RESTRAINED DEGREES OF FREEDOM                                   NOD03070
      DO 530 I=1,NNODE                                                   NOD03080
       DO 530 J=1,6                                                      NOD03090
         IF (BTEST(JTFLG(I),J-1) .AND. .NOT.BTEST(JTFLG(I),J+17)         NOD03100
     &                           .AND. .NOT.BTEST(JTFLG(I),J+11)) THEN   NOD03110
              NDOF=NDOF+1                                                NOD03120
              IDOF(I,J)=NDOF                                             NOD03130
            ENDIF                                                        NOD03140
  530 CONTINUE                                                           NOD03150
C                                                                        NOD03160
C........RESTRAINED DEGREES OF FREEDOM                                   NOD03170
      DO 535 I=1,NNODE                                                   NOD03180
       DO 535 J=1,6                                                      NOD03190
         IF (BTEST(JTFLG(I),J+17) .AND. .NOT.BTEST(JTFLG(I),J+11)) THEN  NOD03200
              NDOF=NDOF+1                                                NOD03210
              IDOF(I,J)=NDOF                                             NOD03220
            ENDIF                                                        NOD03230
  535 CONTINUE                                                           NOD03240
      NREST=NDOF-NFREE-NCOND                                             NOD03250
C                                                                        NOD03260
C........CONSTRAINED DEGREES OF FREEDOM                                  NOD03270
C          I = NODE ID OF SLAVE NODE                                     NOD03280
C          II= NODE ID OF MASTER NODE                                    NOD03290
      DO 520 I=1,NNODE                                                   NOD03300
       DO 520 J=1,6                                                      NOD03310
C                                                                        NOD03320
C        ... CHECK DOF FOR CONSTRAINT ...                                NOD03330
         IF (BTEST(JTFLG(I),J+11)) THEN                                  NOD03340
C                                                                        NOD03350
C              ... DETERMINE MASTER NODE AND DOF # ...                   NOD03360
               II=I                                                      NOD03370
  515          NODE=ID(II)                                               NOD03380
               II=MSTRJT(II,J)                                           NOD03390
               IF (II.EQ.0) THEN                                         NOD03400
                  WRITE (6,525) NODE,ID(I),J                             NOD03410
                  IDOF(I,J)=0                                            NOD03420
                  GO TO 520                                              NOD03430
               ENDIF                                                     NOD03440
               IF (BTEST(JTFLG(II),J+11)) GO TO 515                      NOD03450
               IDOF(I,J)=IDOF(II,J)                                      NOD03460
C                                                                        NOD03470
C              ...CHECK FOR DIFFERENT JOINT COORDINATE SYSTEMS...        NOD03480
               IF (JTCOS(I).NE.JTCOS(II)) THEN                           NOD03490
                  WRITE (6,411) ID(I),ID(II)                             NOD03500
  411             FORMAT(' CONFLICTING COSINE ID NUMBERS, MASTER=',      NOD03510
     &            I6,'  SLAVE=',I6,'  CORRECT INPUT & RERUN...')        NOD03520
               ENDIF                                                     NOD03530
C                                                                        NOD03540
C              ...CALCULATE THE CONSTRAINT MATRIX...                     NOD03550
               IF (J.GE.4) THEN                                          NOD03560
                 XG=COORD(I,1)-COORD(II,1)                               NOD03570
                 YG=COORD(I,2)-COORD(II,2)                               NOD03580
                 ZG=COORD(I,3)-COORD(II,3)                               NOD03590
                 N  =JTCOS(I)                                            NOD03600
                 XMS=COSINE(1,1,N)*XG+COSINE(1,2,N)*YG+COSINE(1,3,N)*ZG  NOD03610
                 YMS=COSINE(2,1,N)*XG+COSINE(2,2,N)*YG+COSINE(2,3,N)*ZG  NOD03620
                 ZMS=COSINE(3,1,N)*XG+COSINE(3,2,N)*YG+COSINE(3,3,N)*ZG  NOD03630
               ENDIF                                                     NOD03640
               IF (J.EQ.4) THEN                                          NOD03650
                   CONST(I,1)=-XMS                                       NOD03660
                   CONST(I,2)= YMS                                       NOD03670
               ELSE IF (J.EQ.5) THEN                                     NOD03680
                   CONST(I,3)= ZMS                                       NOD03690
                   CONST(I,4)=-XMS                                       NOD03700
               ELSE IF (J.EQ.6) THEN                                     NOD03710
                   CONST(I,5)=-YMS                                       NOD03720
                   CONST(I,6)= XMS                                       NOD03730
               ENDIF                                                     NOD03740
            ENDIF                                                        NOD03750
  520 CONTINUE                                                           NOD03760
  525 FORMAT (' NODE#',I6,' NOT FOUND WHILE ASSIGNING CONSTRAINED',      NOD03770
     & ' DOF TO NODE#',I6,' DOF#',I2,' --- DOF IS DELETED ')             NOD03780
C                                                                        NOD03790
C        PRINT OUT ORDERED NODE INFO, AND DEGREES OF FREEDOM             NOD03800
      WRITE (6,1000) 'NODE COORDINATES AND DEGREES OF FREEDOM'           NOD03810
      WRITE (6,600) NDOF,NCOND,NFREE,NREST                               NOD03820
  600 FORMAT                                                             NOD03830
     & (4X,'TOTAL NUMBER OF DEGREES OF FREEDOM.............',I6,/       NOD03840
     & 4X,'NUMBER OF DEGREES OF FREEDOM CONDENSED OUT.....',I6,/       NOD03850
     & 4X,'NUMBER OF FREE DEGREES OF FREEDOM..............',I6,/       NOD03860
     & 4X,'NUMBER OF RESTRAINED DEGREES OF FREEDOM........',I6,//      NOD03870
     & 4X,'  NODE COS#',4X,'X-COORD',8X,'Y-COORD',8X,'Z-COORD',4X,      NOD03880
     & 4X,'FX',6X,'FY',6X,'FZ',6X,'MX',6X,'MY',6X,'MZ')                 NOD03890
      DO 610 I=1,NNODE                                                   NOD03900
        DO 605 J=1,6                                                     NOD03910
          IF (BTEST(JTFLG(I),J-1)) THEN                                  NOD03920
            CX(J)='-R'                                                   NOD03930
          ELSE IF (BTEST(JTFLG(I),(J+11))) THEN                          NOD03940
            CX(J)='-C'                                                   NOD03950
          ELSE                                                           NOD03960
            CX(J)='  '                                                  NOD03970
          ENDIF                                                          NOD03980
  605   CONTINUE                                                         NOD03990
  610  WRITE(6,620) ID(I),JTCOS(I),(COORD(I,J),J=1,3),                   NOD04000
     & (IDOF(I,K),CX(K),K=1,6)                                           NOD04010
  620 FORMAT(4X,I6,1P,I5,3G15.5,0P,6(I6,A2) )                            NOD04020
      WRITE(6,625)                                                       NOD04030
  625 FORMAT(10X,'NOTE: R - RESTRAINED DEGREE OF FREEDOM',/              NOD04040
     &       10X,'      C - CONSTRAINED DEGREE OF FREEDOM')             NOD04050
C                                                                        NOD04060
C--------- PRINT OUT CONSTRAINT MATRIX                                   NOD04070
      IF (BUG) THEN                                                      NOD04080
       FLAG=.TRUE.                                                       NOD04090
       DO 540 I=1,NNODE                                                  NOD04100
         TEST=.FALSE.                                                    NOD04110
         DO 536 J=1,6                                                    NOD04120
  536    TEST = TEST .OR. BTEST(JTFLG(I),(J+11))                         NOD04130
         IF (TEST) THEN                                                  NOD04140
            IF (FLAG) WRITE (6,1000) 'NODE CONSTRAINT MATRIX '           NOD04150
            FLAG=.FALSE.                                                 NOD04160
            WRITE(6,550) ID(I),(CONST(I,J),J=1,6)                        NOD04170
         ENDIF                                                           NOD04180
  540  CONTINUE                                                          NOD04190
  550  FORMAT(4X,I6,1P,                                                  NOD04200
     &     ' T12:',G12.5,' T13:',G12.5,                                  NOD04210
     &     ' T21:',G12.5,' T23:',G12.5,                                  NOD04220
     &     ' T31:',G12.5,' T32:',G12.5)                                  NOD04230
       ENDIF                                                             NOD04240
C                                                                        NOD04250
C--------- PRINT OUT DIRECTION COSINES                                   NOD04260
      WRITE (6,1000) 'DIRECTION COSINES ... '                            NOD04270
      DO 560 ICOS=1,NCOS                                                 NOD04280
          WRITE (6,56)  ICOS,' VX:',(COSINE(1,J,ICOS),J=1,3),            NOD04290
     &                  ' VY:',(COSINE(2,J,ICOS),J=1,3),                 NOD04300
     &                  ' VZ:',(COSINE(3,J,ICOS),J=1,3)                  NOD04310
  560 CONTINUE                                                           NOD04320
C                                                                        NOD04330
      RETURN                                                             NOD04340
 1000 FORMAT (//,5X,A/)                                                  NOD04350
      END                                                                NOD04360
C=======================================================================PBI00010
C          DEBUG UNIT(6),SUBCHK,SUBTRACE                                 PBI00020
C          END DEBUG                                                     PBI00030
      SUBROUTINE PBIT(NNODE,JTFLG,ID)                                    PBI00040
      DIMENSION JTFLG(NNODE),ID(NNODE)                                   PBI00050
      LOGICAL   FLAG(0:31)                                               PBI00060
      LOGICAL BTEST                                                      PBI00070
      WRITE (6,40)                                                       PBI00080
   40 FORMAT (14X,' RESTRAINT',' CONDENSE',' CONSTRAIN',' ELIMINATE',/ PBI00090
     &        14X,4(4X,'PFFMMM'),/14X,4(4X,'XYZXYZ'))                   PBI00100
      DO 20 I=1,NNODE                                                    PBI00110
        DO 10 J=0,23                                                     PBI00120
   10     FLAG(J)=BTEST(JTFLG(I),J)                                      PBI00130
   20   WRITE (6,30) ID(I),(FLAG(K),K=0,23),JTFLG(I)                    PBI00140
   30 FORMAT (' NODE(',I4,') ',4(4X,6L1),I10)                            PBI00150
      RETURN                                                             PBI00160
      END                                                                PBI00170
C=======================================================================REA00010
C          DEBUG UNIT(6),SUBCHK,SUBTRACE                                 REA00020
C          END DEBUG                                                     REA00030
C=======================================================================REA00040
      SUBROUTINE REACTN(IOPT,PRINT,NNODE,L1,L2,L3,L4,FORCE,DISPL,        REA00050
     &    MD,A,KM,NDOF,NLOAD,FACTOR,                                     REA00060
     &    MAXNOD,IDOF,COORD,ID,TITLE,STEPID,HEAD,                        REA00070
     &    NCOS,COSINE,JTCOS,JTFLG,SUM)                                   REA00080
      DIMENSION A(KM),MD(NDOF+1)                                         REA00090
      DIMENSION FORCE(NDOF,NLOAD),DISPL(NDOF,NLOAD)                      REA00100
      DIMENSION IDOF(MAXNOD,6)                                           REA00110
      DIMENSION ID(MAXNOD),COORD(MAXNOD,3)                               REA00120
      DIMENSION F(6),G(6),SUM(6),COSINE(3,3,NCOS),JTCOS(MAXNOD)          REA00130
      DIMENSION JTFLG(MAXNOD)                                            REA00140
      CHARACTER*(*) TITLE(2),STEPID                                      REA00150
      CHARACTER*15 CD(6)                                                 REA00160
      LOGICAL PRINT,HEAD,FLAG,BTEST                                      REA00170
C                                                                        REA00180
      IF (IOPT.EQ.1) THEN                                                REA00190
C                                                                        REA00200
C=============================================                           REA00210
C= MODIFY LOADS FOR RESTRAINT DISPLACEMENTS ===                          REA00220
C=============================================                           REA00230
C CONSIDER THE PARTITIONED STIFFNESS WHERE X2 ARE THE RESTRAINT DISPL.   REA00240
C                                                                        REA00250
C   |P1|   | K11 | K12 | |X1|                                          REA00260
C   |--| = |-----+-----| |--|                                          REA00270
C   |P2|   | K21 | K22 | |X2|                                          REA00280
C                                                                        REA00290
C   THE MODIFIED FREE LOAD IS (P1-K12*X2) = K11*X1                       REA00300
C-------------------------------------------------C                      REA00310
C     DO 10 I=L1,L2                               C  FULL STORAGE       REA00320
C     DO 10 J=1,NLOAD                             C  EQUIVALENT         REA00330
C     DO 10 K=L3,L4                               C                     REA00340
C  10   FORCE(I,J)=FORCE(I,J)-K(I,K)*DISPL(K,J)  C                      REA00350
C-------------------------------------------------C                      REA00360
      DO 20 K=L3,L4                                                      REA00370
      MDK=MD(K+1)                                                        REA00380
        DO 20 J=1,NLOAD                                                  REA00390
        DO 10 I=L2,L1,-1                                                 REA00400
          IK=MD(K)+K-I                                                   REA00410
          IF (IK.GE.MDK) GO TO 20                                        REA00420
          FIJ=FORCE(I,J)                                                 REA00430
          AIK=A(IK)                                                      REA00440
          DKJ=DISPL(K,J)                                                 REA00450
   10     FORCE(I,J)=FORCE(I,J)-A(IK)*DISPL(K,J)                         REA00460
   20   CONTINUE                                                        REA00470
        IF (PRINT)                                                       REA00480
     &  CALL WMATRX(FORCE,NDOF,NLOAD,'MODIFIED LOADS')                   REA00490
C                                                                        REA00500
      ELSE IF (IOPT.EQ.2) THEN                                           REA00510
C                                                                        REA00520
C=========================                                              REA00530
C= SOLVE FOR REACTIONS ===                                              REA00540
C=========================                                              REA00550
C CONSIDER THE PARTITIONED STIFFNESS WHERE X2 ARE THE RESTRAINT DISPL.   REA00560
C                                                                        REA00570
C   |P1|   | K11 | K12 | |X1|                                          REA00580
C   |--| = |-----+-----| |--| * FACTOR                                    REA00590
C   |P2|   | K21 | K22 | |X2|                                          REA00600
C                                                                        REA00610
C   THE REACTIONS ARE  P2 = P2 + ( K21*X1 + K22*X2 ) * FACTOR            REA00620
C   NOTE: BOTH P1 AND P2 CONTAIN FIXED END FORCES FROM MEMBER LOADS...   REA00630
C         THUS, K21*X1 + K22*X2 IS ADDED TO P2 TO GET THE REACTIONS.     REA00640
C-------------------------------------------------C                      REA00650
C     DO 120 I=L3,L4                              C    FULL STORAGE     REA00660
C     DO 120 J=1,NLOAD                            C    MODE EQUIVALENT  REA00670
CC     FORCE(I,J)=0   * P2 CONTAINS FEF!!!        C                     REA00680
C      DO 110 K=L1,L2                              C                     REA00690
C 110   FORCE(I,J)=FORCE(I,J)+K(I,K)*DISPL(K,J) C                       REA00700
C      DO 120 K=L3,L4                              C                     REA00710
C 120   FORCE(I,J)=FORCE(I,J)+K(I,K)*DISPL(K,J) C                       REA00720
C-------------------------------------------------C                      REA00730
C     DO 100 J=1,NLOAD                                                   REA00740
C     DO 100 I=L3,L4                                                     REA00750
C 100  FORCE(I,J)=0                                                      REA00760
C                                                                        REA00770
      DO 120 I=L3,L4                                                     REA00780
      DO 120 J=1,NLOAD                                                   REA00790
      DO 110 K=L2,L1,-1                                                  REA00800
        KI=MD(I)+I-K                                                     REA00810
        IF (KI.GE.MD(I+1)) GO TO 120                                     REA00820
        AKI=A(KI)                                                        REA00830
  110   FORCE(I,J)=FORCE(I,J)+A(KI)*DISPL(K,J)*FACTOR                    REA00840
  120 CONTINUE                                                           REA00850
C                                                                        REA00860
       DO 140 I=L3,L4                                                    REA00870
       DO 140 K=L3,L4                                                    REA00880
         IF (I.LE.K) THEN                                                REA00890
           IK=MD(K)+K-I                                                  REA00900
           MDK=MD(K+1)                                                   REA00910
         ELSE                                                            REA00920
           IK=MD(I)+I-K                                                  REA00930
           MDK=MD(I+1)                                                   REA00940
         ENDIF                                                           REA00950
         IF (IK.GE.MDK)  GO TO 140                                       REA00960
         AIK=A(IK)                                                       REA00970
         DO 130 J=1,NLOAD                                                REA00980
  130      FORCE(I,J)=FORCE(I,J)+A(IK)*DISPL(K,J)*FACTOR                 REA00990
  140  CONTINUE                                                          REA01000
C                                                                        REA01010
        IF (PRINT)                                                       REA01020
     &  CALL WMATRX(FORCE,NDOF,NLOAD,'FINAL LOADS AND REACTIONS')        REA01030
C                                                                        REA01040
      ELSE IF (IOPT.EQ.3 .OR. IOPT.EQ.4) THEN                            REA01050
            NF=L2                                                        REA01060
C=============================                                          REA01070
C= LOOP FOR EACH LOAD CASE  ===                                         REA01080
C=============================                                          REA01090
      DO 300 L=1,NLOAD                                                   REA01100
C                                                                        REA01110
C--------------------------------                                        REA01120
C= PRINT GLOBAL REACTIONS    ===                                        REA01130
C--------------------------------                                        REA01140
C                                                                        REA01150
      IF (PRINT) THEN                                                    REA01160
          IF (HEAD) WRITE (6,320) TITLE(1),TITLE(2)                      REA01170
            NREACT=0                                                     REA01180
```

```
            IF (IOPT.NE.4) WRITE (6,321) L,STEPID                       REA01190
            IF (IOPT.EQ.4) WRITE (6,421)                                REA01200
        ENDIF                                                           REA01210
          IF (IOPT.EQ.3) THEN                                           REA01220
            DO 190 I=1,6                                                REA01230
  190       SUM(I)=0                                                    REA01240
          ENDIF                                                         REA01250
C                                                                       REA01260
        IF ( .NOT.(NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND.             REA01270
     &           COSINE(2,2,1).EQ.1 .AND. COSINE(3,3,1).EQ.1)           REA01280
     &    .AND. IOPT.EQ.4  ) GO TO 279                                  REA01290
C                                                                       REA01300
        DO 200 I=1,NNODE                                                REA01310
C                                                                       REA01320
C.............. GET REACTNS IN JOINT COORDINATE SYSTEM                 REA01330
          FLAG=.FALSE.                                                  REA01340
          DO 209 J=1,6                                                  REA01350
            IJ=IDOF(I,J)                                                REA01360
            IF (IJ.LE.NF .OR. BTEST(JTFLG(I),J+11)) THEN                REA01370
              F(J)=0                                                    REA01380
            ELSE                                                        REA01390
              F(J)=FORCE(IJ,L)                                          REA01400
              FLAG=.TRUE.                                               REA01410
            ENDIF                                                       REA01420
  209     CONTINUE                                                      REA01430
          IF (.NOT.FLAG) GO TO 200                                      REA01440
C                                                                       REA01450
C.............. TRANSFER REACTNS TO GLOBAL COORDINATE SYSTEM           REA01460
          DO 229 I1=1,3                                                 REA01470
            I2=I1+3                                                     REA01480
            G(I1)=0                                                     REA01490
            G(I2)=0                                                     REA01500
            DO 219 J1=1,3                                               REA01510
              J2=J1+3                                                   REA01520
              G(I1)=G(I1) + COSINE(J1,I1.JTCOS(I))*F(J1)                REA01530
  219         G(I2)=G(I2) + COSINE(J1,I1.JTCOS(I))*F(J2)                REA01540
            IF (PRINT) THEN                                             REA01550
                WRITE (CD(I1),210) G(I1)                                REA01560
                WRITE (CD(I2),210) G(I2)                                REA01570
            ENDIF                                                       REA01580
  229       CONTINUE                                                    REA01590
C                                                                       REA01600
C.............. SUM REACTIONS ...........                              REA01610
          IF (IOPT.EQ.3) THEN                                           REA01620
            DO 240 I1=1,6                                               REA01630
  240         SUM(I1)=SUM(I1) + G(I1)                                   REA01640
            SUM(4)= SUM(4) - G(2)*COORD(I,3) + G(3)*COORD(I,2)          REA01650
            SUM(5)= SUM(5) + G(1)*COORD(I,3) - G(3)*COORD(I,1)          REA01660
            SUM(6)= SUM(6) - G(1)*COORD(I,2) + G(2)*COORD(I,1)          REA01670
          ENDIF                                                         REA01680
C                                                                       REA01690
C.............. PRINT GLOBAL REACTIONS .........                       REA01700
          NREACT=NREACT+1                                               REA01710
          IF (PRINT) WRITE (6,220) ID(I),(CD(K),K=1,6)                  REA01720
  200   CONTINUE                                                        REA01730
        IF (IOPT.EQ.3 .AND. PRINT ) WRITE (6,230) SUM                   REA01740
  279   IF (IOPT.EQ.4 .AND. PRINT ) WRITE (6,231) SUM                   REA01750
C                                                                       REA01760
C*****************************************************                REA01770
C* PRINT LOCAL JOINT COORDINATE SYSTEMS REACTIONS  ***                REA01780
C*****************************************************                REA01790
C                                                                       REA01800
        IF (.NOT. PRINT ) GO TO 300                                     REA01810
        IF (NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND. COSINE(2,2,1).EQ.1 REA01820
     &              .AND. COSINE(3,3,1).EQ.1 ) GO TO 300                REA01830
C                                                                       REA01840
        IF (HEAD.AND. NREACT.GT.20) WRITE (6,320) TITLE(1),TITLE(2)     REA01850
          IF (IOPT.NE.4) WRITE (6,322) L,STEPID                         REA01860
          IF (IOPT.EQ.4) WRITE (6,422)                                  REA01870
          DO 290 I=1,NNODE                                              REA01880
C                                                                       REA01890
C.............. GET REACTNS IN JOINT COORDINATE SYSTEM                 REA01900
          FLAG=.FALSE.                                                  REA01910
          DO 280 J=1,6                                                  REA01920
            IJ=IDOF(I,J)                                                REA01930
            IF (IJ.LE.NF .OR. BTEST(JTFLG(I),J+11)) THEN                REA01940
              CD(J)=' '                                                 REA01950
            ELSE                                                        REA01960
              WRITE (CD(J),210) FORCE(IJ,L)                             REA01970
              FLAG=.TRUE.                                               REA01980
            ENDIF                                                       REA01990
  280     CONTINUE                                                      REA02000
          IF (.NOT.FLAG) GO TO 290                                      REA02010
C                                                                       REA02020
C.............. PRINT REACTNS IN LOCAL COORDINATE SYSTEM .....         REA02030
          WRITE (6,221) ID(I),(CD(K),K=1,6),JTCOS(I)                    REA02040
  290   CONTINUE                                                        REA02050
CCC     IF (IOPT.EQ.4) WRITE (6,231) SUM                                REA02060
  300 CONTINUE                                                          REA02070
C                                                                       REA02080
  210 FORMAT (1P,G15.7)                                                 REA02090
  220 FORMAT (5X,I5,6A)                                                 REA02100
  221 FORMAT (5X,I5,6A,I10)                                             REA02110
  230 FORMAT ('                ',90('-'),/                              REA02120
     &        ' SUMMATION',6G15.7)                                      REA02130
  231 FORMAT (' MAX OF ALL',89('-'),/                                   REA02140
     &        ' GCS SUMM.',6G15.7 )                                     REA02150
  320 FORMAT ('1 STRUCTURE....: ',A,/                                   REA02160
     &        ' SOLUTION.....: ',A)                                     REA02170
  321 FORMAT (///'  GCS RESTRAINT REACTIONS,   LOADING #',I5,5X,A,/     REA02180
     &        ' ------------------------'//                             REA02190
     &  6X,'NODE',7X,'FX',13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/) REA02200
  322 FORMAT (///'  JCS RESTRAINT REACTIONS,   LOADING #',I5,5X,A,/     REA02210
     &        ' ------------------------'//                             REA02220
     &  6X,'NODE',7X,'FX',13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ', REA02230
     &  6X,'  COSINE #'/)                                               REA02240
  421 FORMAT (///'  MAXIMUM GCS RESTRAINT REACTIONS',                   REA02250
     &        5X,'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'/   REA02260
     &        ' ------------------------------------'//                REA02270
     &  6X,'NODE',7X,'FX',13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/) REA02280
  422 FORMAT (///'  MAXIMUM JCS RESTRAINT REACTIONS ',                  REA02290
     &        5X,'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'/   REA02300
     &        ' ------------------------------------'//                REA02310
     &  6X,'NODE',7X,'FX',13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ', REA02320
     &  6X,'  COSINE #'/)                                               REA02330
C                                                                       REA02340
        ELSE                                                            REA02350
          WRITE (6,*)'INVALID OPTION IN SUBROUTINE  REA_CTN, IOPT=',IOPT REA02360
        ENDIF                                                           REA02370
        RETURN                                                          REA02380
        END                                                             REA02390
C*********************************************************************  REA02400
C          DEBUG UNIT(6),SUBCHK,SUBTRACE                                ROT00010
C          END DEBUG                                                    ROT00020
      SUBROUTINE ROTXYZ(VX,VY,COSINE,CT,EX,EY,EZ,CTXYZ,GLOXYZ,NROW)     ROT00030
      LOGICAL BTEST                                                     ROT00040
      DIMENSION VX(3),VY(3),VZ(3),COSINE(3,3),CT(3,3),                  ROT00050
     &    XYZ(3,3),CTXYZ(3,3),CE(3,3),GLOXYZ(NROW,6)                    ROT00060
C---------------------------------------------------------------------  ROT00070
C                                                                       ROT00080
C     THIS SUBROUTINE PREFORMS SEVERAL FUNCTIONS:                       ROT00090
C     1) CALCULATE LOCAL COORDINATE SYS COSINE MATRIX.                  ROT00100
C     2) CALCULATE ROTATION MATRIX BETWEEN LOCAL AND JOINT COORD (CT)   ROT00110
C     3) CALCULATE TRANSFORMATION FOR GLOBAL CONSTRAINTS AND MEMBER     ROT00120
C        END DEPTH- STORE IN (CTXYZ)                                    ROT00130
C                                                                       ROT00140
C--- INPUT VARIABLES----------------------------------------------------ROT00150
C     COSINE = INPUT JOINT COORDINATE DIRECTION COSINE                  ROT00160
C     VZ     = INPUT X-DIRECTION OF ELEMENT COORDINATE SYS              ROT00170
C     VY     = INPUT VECTOR IN THE ELEMENT COORDINATE SYS  XY PLANE.    ROT00180
C     GLOXYZ = GLOBAL CONSTRAINT TRANSFORMATION COIEFFICIENTS           ROT00190
C     EX     = MEMBER END ECCENTRICITY IN THE LOCAL X AXIS.             ROT00200
C     EY     = MEMBER END ECCENTRICITY IN THE LOCAL Y AXIS.             ROT00210
C     EZ     = MEMBER END ECCENTRICITY IN THE LOCAL Z AXIS.             ROT00220
C                                                                       ROT00230
C--- OUTPUT VARIABLES---------------------------------------------------ROT00240
C     VX     = UNIT VECTOR, X-DIRECTION OF ELEMENT COORDINATE SYS       ROT00250
C     VY     = UNIT VECTOR, Y-DIRECTION OF ELEMENT COORDINATE SYS       ROT00260
C     VZ     = UNIT VECTOR, Z-DIRECTION OF ELEMENT COORDINATE SYS       ROT00270
C     CT     = ROTATED COORDINATE TRANSFORMATION                        ROT00280
C     CTXYZ  = CONSTRAINT AND MEMBER END DEPTH TRANSFORMATION           ROT00290
C                                                                       ROT00300
C     | FX |                                 | FX |                    ROT00310
C     | FY |                                 | FY |                    ROT00320
C     | FZ |                 | CT  |  0  |   | FZ |                    ROT00330
C     | --- |       =       |-----|-----|   |----|                    ROT00340
C     | MX |                 |CTXYZ| CT  |   | MX |                    ROT00350
C     | MZ |                                 | MY |                    ROT00360
C     | MZ |                                 | MZ |                    ROT00370
C          GLOBAL                                  ELEMENT COORD.      ROT00380
C                                                                       ROT00390
C---------------------------------------------------------------------  ROT00400
C                                                                       ROT00410
      CROSSI(A1,B1,C1,A2,B2,C2)=B1*C2-B2*C1                             ROT00420
      CROSSJ(A1,B1,C1,A2,B2,C2)=C1*A2-C2*A1                             ROT00430
      CROSSK(A1,B1,C1,A2,B2,C2)=A1*B2-A2*B1                             ROT00440
      VALUE (A1,A2,A3)=SQRT(A1**2+A2**2+A3**2)                          ROT00450
      DOT   (A1,B1,C1,A2,B2,C2)=A1*A2+B1*B2+C1*C2                       ROT00460
C                                                                       ROT00470
C ------- CHECK ANGLE BETWEEN VX AND VT                                 ROT00480
      C1=VCOS(VX,VT)                                                    ROT00490
      IF (ABS(C1).GT.999) WRITE (6,*) 'VT IS PARAELL TO VX, REINPUT VT' ROT00500
C                                                                       ROT00510
C ------- CHECK VT, IF |VT|.EQ.0 , SET VT IN GLOBAL YZ PLANE...         ROT00520
      AVT=VALUE(VT(1),VT(2),VT(3))                                      ROT00530
      IF (AVT.EQ.0) THEN                                                ROT00540
          VZ(1)=1.0                                                     ROT00550
          VZ(2)= .0                                                     ROT00560
          VZ(3)= .0                                                     ROT00570
      ELSE                                                              ROT00580
C ------- CALCULATE VZ, IF VT WAS CORRECTLY INPUT                       ROT00590
          VZ(1)=CROSSI(VX(1),VX(2),VX(3),VT(1),VT(2),VT(3))             ROT00600
          VZ(2)=CROSSJ(VX(1),VX(2),VX(3),VT(1),VT(2),VT(3))             ROT00610
          VZ(3)=CROSSK(VX(1),VX(2),VX(3),VT(1),VT(2),VT(3))             ROT00620
      ENDIF                                                             ROT00630
C                                                                       ROT00640
C ------- RECALCULATE VT TO INSURE VX,VT AND VZ ARE ORTHOGONAL...       ROT00650
      VT(1)=CROSSI(VZ(1),VZ(2),VZ(3),VX(1),VX(2),VX(3))                 ROT00660
      VT(2)=CROSSJ(VZ(1),VZ(2),VZ(3),VX(1),VX(2),VX(3))                 ROT00670
      VT(3)=CROSSK(VZ(1),VZ(2),VZ(3),VX(1),VX(2),VX(3))                 ROT00680
C                                                                       ROT00690
C ------- CHECK ANGLE BETWEEN VX AND VT                                 ROT00700
      C1=VCOS(VX,VT)                                                    ROT00710
      IF (ABS(C1).GT.999) WRITE (6,*) 'VT IS PARAELL TO VX, REINPUT VT' ROT00720
C                                                                       ROT00730
C ------- NORMALIZE VX,VT,VZ                                           ROT00740
      AVX=VALUE(VX(1),VX(2),VX(3))                                      ROT00750
      AVT=VALUE(VT(1),VT(2),VT(3))                                      ROT00760
      AVZ=VALUE(VZ(1),VZ(2),VZ(3))                                      ROT00770
      DO 10 I=1,3                                                       ROT00780
         VX(I)=VX(I)/AVX                                                ROT00790
         VT(I)=VT(I)/AVT                                                ROT00800
         VZ(I)=VZ(I)/AVZ                                                ROT00810
         CE(1,I)=VX(I)                                                  ROT00820
         CE(2,I)=VT(I)                                                  ROT00830
  10     CE(3,I)=VZ(I)                                                  ROT00840
C                                                                       ROT00850
C ------- CALC THE DIRECTION COSINE MATRIX                              ROT00860
      DO 15 I=1,3                                                       ROT00870
      DO 15 J=1,3                                                       ROT00880
          CT(I,J)=0                                                     ROT00890
          DO 15 K=1,3                                                   ROT00900
  15         CT(I,J)=CT(I,J)+COSINE(I,K)*CE(J,K)                       ROT00910
C                                                                       ROT00920
C ------- REMOVE THE FUZZ FROM THE COSINE MATRIX                        ROT00930
      DO 20 I=1,3                                                       ROT00940
      DO 20 J=1,3                                                       ROT00950
  20    IF (ABS(CT(I,J)).LT. 0.0001) CT(I,J)=0.                         ROT00960
C                                                                       ROT00970
C ------- SET UP THE GLOBAL CONSTRAINT MATRIX                           ROT00980
      XYZ(1,1)= .0                                                      ROT00990
      XYZ(1,2)=GLOXYZ(1,1)                                              ROT01000
      XYZ(1,3)=GLOXYZ(1,2)                                              ROT01010
      XYZ(2,1)=GLOXYZ(1,3)                                              ROT01020
      XYZ(2,2)= .0                                                      ROT01030
      XYZ(2,3)=GLOXYZ(1,4)                                              ROT01040
      XYZ(3,1)=GLOXYZ(1,5)                                              ROT01050
      XYZ(3,2)=GLOXYZ(1,6)                                              ROT01060
      XYZ(3,3)= .0                                                      ROT01070
C                                                                       ROT01080
      DO 25 I=1,3                                                       ROT01090
        DO 25 J=1,3                                                     ROT01100
          CTXYZ(I,J)=0                                                  ROT01110
          DO 25 K=1,3                                                   ROT01120
  25        CTXYZ(I,J)=CTXYZ(I,J) + XYZ(I,K)*CT(K,J)                  ROT01130
C                                                                       ROT01140
C ------- SKIP THE MEMBER END ECCENTRICITY MATRIX IF EX=EY=EZ=0         ROT01150
      IF (EX.EQ.0 .AND. EY.EQ.0 .AND. EZ.EQ.0) RETURN                   ROT01160
C                                                                       ROT01170
C ------- CALC THE MEMBER END ECCENTRICITY MATRIX                       ROT01180
      XYZ(1,1)= .0                                                      ROT01190
      XYZ(1,2)=-EZ                                                      ROT01200
      XYZ(1,3)= EY                                                      ROT01210
      XYZ(2,1)= EZ                                                      ROT01220
      XYZ(2,2)= .0                                                      ROT01230
      XYZ(2,3)=-EX                                                      ROT01240
      XYZ(3,1)=-EY                                                      ROT01250
      XYZ(3,2)= EX                                                      ROT01260
      XYZ(3,3)= .0                                                      ROT01270
C                                                                       ROT01280
C ------- CALC THE MEMBER END ECCENTRICITY MATRIX                       ROT01290
      DO 30 I=1,3                                                       ROT01300
        DO 30 J=1,3                                                     ROT01310
          DO 30 K=1,3                                                   ROT01320
  30        CTXYZ(I,J)=CTXYZ(I,J) + CT(I,K)*XYZ(K,J)                  ROT01330
C                                                                       ROT01340
      RETURN                                                            ROT01350
      END                                                               ROT01360
C*********************************************************************  ROT01370
C          DEBUG UNIT(6),SUBCHK,SUBTRACE                                SKY00010
C          END DEBUG                                                    SKY00020
      SUBROUTINE SKTLIN(JFLAG,IOPT,NDOF,IBUG,TITLE,MDIAG,               SKY00030
     &                  IELDOF,LM,SE,MD,NAME)                           SKY00040
      DIMENSION MDIAG(NDOF+1)                                           SKY00050
      DIMENSION MD(IELDOF+1),LM(IELDOF),SE(IELDOF*(IELDOF+1)/2)         SKY00060
      LOGICAL BTEST                                                     SKY00070
      CHARACTER*(*) NAME,TITLE(2)                                       SKY00080
      CHARACTER*1 CX(100)                                               SKY00090
      LOGICAL PT,BUG                                                    SKY00100
C                                                                       SKY00110
      IF (IOPT.EQ.1) THEN                                               SKY00120
        DO 10 I=1,NDOF+1                                                SKY00130
  10      MDIAG(I)=I                                                    SKY00140
        IF ( BTEST(IBUG,12) ) THEN                                      SKY00150
          WRITE (6,*) 'INITIAL VALUES'                                  SKY00160
          WRITE (6,320) (I,MDIAG(I),I=1,NDOF)                           SKY00170
        ENDIF                                                           SKY00180
C                                                                       SKY00190
      ELSE IF (IOPT.EQ.2) THEN                                          SKY00200
C---------- DETERMINE THE SKYLINE OF THIS ELEMENT, AND MODIFY THE       SKY00210
C           GLOBAL SKYLINE...                                           SKY00220
C..... I    = THE ROW OF THE MEMBER STIFFNESS MATRIX                    SKY00230
```

```
C....... J    = THE COLUMN OF THE MEMBER STIFFNESS MATRIX                      SKT00250
C....... IJ   = THE ADDRESS OF THE MEMBER STIFFNESS TERM I,J                  SKT00260
C....... LMI  = THE ROW OF THE GLOBAL STIFFNESS MATRIX CORRESPONDING          SKT00270
C               TO ROW I OF THE MEMBER STIFFNESS MATRIX                       SKT00280
C....... LMJ  = THE COLUMN OF THE GLOBAL STIFFNESS MATRIX CORRESPONDING       SKT00290
C               TO COLUMN J OF THE MEMBER STIFFNESS MATRIX                    SKT00300
C....... MDIAG(JJ) = THE JJ'TH ELEMENT OF THE GLOBAL SKYLINE MATRIX.          SKT00310
C               AT THIS POINT THE GLOBAL SKYLINE MATRIX CONTAINS THE          SKT00320
C               LOWEST ROW NUMBER THAT CONTAINS A NON-ZERO STIFFNESS          SKT00330
C               TERM.                                                         SKT00340
C                                                                             SKT00350
      DO 250 I=1,IELDOF                                                       SKT00360
        LMI=LM(I)                                                             SKT00370
        IF (LMI.LE.0) GO TO 250                                               SKT00380
        DO 240 J=I,IELDOF                                                     SKT00390
          LMJ=LM(J)                                                           SKT00400
          IF (LMJ.LE.0) GO TO 240                                             SKT00410
            IJ=MD(J)+J-I                                                      SKT00420
            IF (SE(IJ).EQ.0 .AND. JFLAG .EQ. 0 ) GO TO 240                    SKT00430
C           IF (SE(IJ).EQ.0  ) GO TO 240                                      SKT00440
            IF (LMI.LE.LMJ) THEN                                              SKT00450
                MDIAG(LMJ)=MIN(MDIAG(LMJ),LMI)                                SKT00460
            ELSE                                                              SKT00470
                MDIAG(LMI)=MIN(MDIAG(LMI),LMJ)                                SKT00480
            ENDIF                                                             SKT00490
  240     CONTINUE                                                            SKT00500
  250   CONTINUE                                                              SKT00510
C-----------------------------------------------------------------------------SKT00520
C                                                                             SKT00530
      ELSE IF (ICPT.EQ.3) THEN                                                SKT00540
C----------- CALCULATE THE INDICIES OF THE MAIN DIAGONAL TERMS...             SKT00550
      IF ( BTEST(IBUG,12) ) THEN                                              SKT00560
          WRITE (6,*) 'AFTER COLUMN HEIGHTS'                                  SKT00570
          WRITE (6,320) (I,MDIAG(I),I=1,NDOF)                                 SKT00580
      ENDIF                                                                   SKT00590
      MDIAG(1)=1                                                              SKT00600
      ICOLHT=1                                                                SKT00610
      DO 310 IROW=2,(NDOF+1)                                                  SKT00620
         ILOCN=ICOLHT + MDIAG(IROW-1)                                         SKT00630
         ICOLHT=IROW - (MDIAG(IROW)-1)                                        SKT00640
  310    MDIAG(IROW)=ILOCN                                                    SKT00650
      IF ( BTEST(IBUG,12) ) THEN                                              SKT00660
          WRITE (6,*) 'ADDRESS OF MAIN DIAGONAL ELEMENTS'                     SKT00670
          WRITE (6,320) (I,MDIAG(I),I=1,NDOF)                                 SKT00680
  320     FORMAT (' MDIAG(',I3,')=',I5)                                       SKT00690
      ENDIF                                                                   SKT00700
  321 FORMAT (2I6)                                                            SKT00710
C                                                                             SKT00720
      IF ( .NOT.(BTEST(IBUG,12) .OR. BTEST(IBUG,9)) ) RETURN                  SKT00730
C---------- PRINT A MAP OF THE STIFFNESS ARRAY                                SKT00740
      NL= MDIAG(NDOF+1)-MDIAG(1) +1                                           SKT00750
      NP=NDOF/100                                                             SKT00760
      NR=MOD(NDOF,100)                                                        SKT00770
      IF (NR.GT.0) NP=NP+1                                                    SKT00780
      DO 270 IP=1,NP                                                          SKT00790
         JC1=(IP-1)*100                                                       SKT00800
         JC2=MIN(IP*100,NDOF)                                                 SKT00810
         J2=100                                                               SKT00820
         IF (IP.EQ.NP) J2=NR                                                  SKT00830
         WRITE (6,290) TITLE(1),TITLE(2),NAME,NL,(I,I=JC1,JC2,10)             SKT00840
         DO 270 I=1,NDOF                                                      SKT00850
            PT=.FALSE.                                                        SKT00860
            DO 260 JJ=1,J2                                                    SKT00870
              J=JJ+(IP-1)*100                                                 SKT00880
              IJ=MDIAG(J)+J-I                                                 SKT00890
              IF (IJ.LT.MDIAG(J+1) .AND. I.LT.J) THEN                         SKT00900
                   CX(JJ)='X'                                                 SKT00910
                   PT=.TRUE.                                                  SKT00920
              ELSE IF (I.EQ.J) THEN                                           SKT00930
                   CX(JJ)='D'                                                 SKT00940
                   PT=.TRUE.                                                  SKT00950
              ELSE                                                            SKT00960
                   CX(JJ)=' '                                                 SKT00970
              ENDIF                                                           SKT00980
  260       CONTINUE                                                          SKT00990
  270 IF (PT) WRITE (6,295) I,(CX(J),J=1,J2)                                  SKT01000
C                                                                             SKT01010
  290 FORMAT ('1 STRUCTURE....: ',A,//                                        SKT01020
     &        '  SOLUTION.....: ',A,//                                       SKT01030
     &        '  SKYLINE OF THE ',A,' MATRIX ',//                            SKT01040
     & '  THE MATRIX REQUIRES',I6,' STORAGE LOCATIONS',                       SKT01050
     & /2X,12(I4,'....'))                                                    SKT01060
  295 FORMAT (16,132A1)                                                       SKT01070
C                                                                             SKT01080
      ENDIF                                                                   SKT01090
C                                                                             SKT01100
      RETURN                                                                  SKT01110
      END                                                                     SKT01120
C-----------------------------------------------------------------------------SKT01010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT                                  SKT00020
C         END DEBUG                                                           SKT00030
      SUBROUTINE SKYLN2(NDOF,L3,L4,MDS,MD,MDMASS,MDKG,KGFORM)                 SKT00040
      DIMENSION MDMASS(NDOF+1),MD(NDOF+1),MDS(NDOF+1),MDKG(NDOF+1)            SKT00050
C                                                                             SKT00060
C                                                                             SKT00070
C------- DEVELOP SKYLINE OF DYNAMIC STIFFNESS (S)                             SKT00080
C....... MDS(I)   = ADDRESS OF DYNAMIC STIFF. MAIN DIAGONAL TERM ROW I        SKT00090
C....... MD(I)    = ADDRESS OF STIFFNESS MAIN DIAGONAL TERM ROW I            SKT00100
C....... MDMAS(I) = ADDRESS OF MASS MAIN DIAGONAL TERM ROW I                  SKT00110
C....... MDKG(I)  = ADDRESS OF GEOMETRIC STIFF. MAIN DIAGONAL TERM ROW I     SKT00120
C                                                                             SKT00130
C....... LSTIFF = # OF STIFFNESS TERMS IN COLUMN J                            SKT00140
C....... LMASS  = # OF MASS TERMS IN COLUMN J                                 SKT00150
C....... LKG    = # OF KG TERMS IN COLUMN J                                   SKT00160
C....... L      = # OF S  TERMS IN COLUMN J                                   SKT00170
C                                                                             SKT00180
      NFREE=L4-L3+1                                                           SKT00190
      MDS(1)=1                                                                SKT00200
      LKG=0                                                                   SKT00210
      DO 110 J1=2,NFREE+1                                                     SKT00220
        J=J1+L3-1                                                             SKT00230
        LSTIFF=MD(J)-MD(J-1)                                                  SKT00240
        LMASS =MDMASS(J)-MDMASS(J-1)                                          SKT00250
        IF (KGFORM.EQ.2)  LKG = MDKG(J)-MDKG(J-1)                             SKT00260
        L=MIN(J1-1,MAX(LSTIFF,LMASS,LKG))                                     SKT00270
  110   MDS(J1)=MDS(J1-1)+L                                                   SKT00280
C                                                                             SKT00290
      RETURN                                                                  SKT00300
      END                                                                     SKT00310
C-----------------------------------------------------------------------------SMA00010
C--- SUBROUTINE SMAX,  FINDS THE MAXIMUN AND MINIMUN VALUES                   SMA00020
C                                          (AFTER LANL)                       SMA00030
C         DEBUG UNIT(6),SUBCHK,SUBTRACE                                       SMA00040
C         END DEBUG                                                           SMA00050
      SUBROUTINE SMAX (N,DT,A,TMX,TMN,T1,T2,NLEN,AVE,STDEV,RMS)               SMA00060
      DIMENSION A(NLEN)                                                       SMA00070
      LOGICAL BTEST                                                           SMA00080
      REAL*8  T   ,SUMX,SUMSQ ,ADBL                                           SMA00090
          AVE=0                                                               SMA00100
          STDEV=0                                                             SMA00110
          RMS=0                                                               SMA00120
          TMX=A(1)                                                            SMA00130
          TMN=A(1)                                                            SMA00140
          T=DT                                                                SMA00150
          T1=DT                                                               SMA00160
          T2=DT                                                               SMA00170
          IF (N.LE.1) RETURN                                                  SMA00180
          SUMX=A(1)                                                           SMA00190
          ADBL=A(1)                                                           SMA00200
          SUMXSQ=ADBL**2                                                      SMA00210
      DO 40 I=2,N                                                             SMA00220
       ADBL=A(I)                                                              SMA00230
          SUMX   = SUMX   + ADBL                                              SMA00240
          SUMXSQ = SUMXSQ + ADBL**2                                           SMA00250
            IF (TMX.LT.A(I)) THEN                                             SMA00260
                TMX=A(I)                                                      SMA00270
                T1=T                                                          SMA00280
            ELSE IF (TMN.GT.A(I)) THEN                                        SMA00290
                TMN=A(I)                                                      SMA00300
                T2=T                                                          SMA00310
            ENDIF                                                             SMA00320
          T=T+DT                                                              SMA00330
   40   CONTINUE                                                             SMA00340
        AVE=SUMX/N                                                            SMA00350
        IF ((N*SUMXSQ-SUMX**2) .GT.0 )                                        SMA00360
     &    STDEV=SQRT( (N*SUMXSQ-SUMX**2)/(N*(N-1)) )                          SMA00370
        RMS=SQRT(SUMXSQ/N)                                                    SMA00380
        RETURN                                                                SMA00390
        END                                                                   SMA00400
C-----------------------------------------------------------------------------SOL00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE ,INIT                                 SOL00020
C         END DEBUG                                                           SOL00030
      SUBROUTINE SOLN(OPTION,PLACE,CPU,TCPU,REQCPU)                           SOL00040
      LOGICAL BTEST                                                           SOL00050
      CHARACTER*80 NAME,OPTION                                                SOL00060
      CHARACTER*6  PLACE                                                      SOL00070
      INCLUDE (ZCOMN)                                                         SOL00080
      STEPID=' '                                                              SOL00090
C                                                                             SOL00100
C*************************                                                    SOL00110
C= DETERMINE SOLUTION NO. ==                                                  SOL00120
C*************************                                                    SOL00130
      CALL GETINT(OPTION,'SOL',NSOLN,0,.TRUE.,.FALSE.)                        SOL00140
C                                                                             SOL00150
C*************************                                                    SOL00160
C= CHOOSE THE SOLUTION ==                                                     SOL00170
C*************************                                                    SOL00180
      IF (NSOLN.EQ.1) THEN                                                    SOL00190
            CALL SOL01                                                        SOL00200
      ELSE IF (NSOLN.EQ.2) THEN                                               SOL00210
            NLOAD=1                                                           SOL00220
            MAXELD=0                                                          SOL00230
            CALL SOL02(PLACE,CPU,TCPU,REQCPU)                                 SOL00240
      ELSE IF (NSOLN.EQ.3) THEN                                               SOL00250
            NLOAD=1                                                           SOL00260
            MAXELD=0                                                          SOL00270
            CALL SOL03                                                        SOL00280
      ELSE IF (NSOLN.EQ.4) THEN                                               SOL00290
            NLOAD=1                                                           SOL00300
            CALL SOL04                                                        SOL00310
      ELSE IF (NSOLN.EQ.5) THEN                                               SOL00320
            NLOAD=1                                                           SOL00330
            MAXELD=0                                                          SOL00340
            CALL SOL05                                                        SOL00350
      ELSE                                                                    SOL00360
            WRITE (6,*) 'INVALID SOLN#',NSOLN                                 SOL00370
            STOP 'INVALID SOLUTION NUMBER'                                    SOL00380
      ENDIF                                                                   SOL00390
   30 RETURN                                                                  SOL00400
      END                                                                     SOL00410
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP                                        SOL00010
C-----------------------------------------------------------------------------SOL00020
C         DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT                                  SOL00030
C         END DEBUG                                                           SOL00040
C-----------------------------------------------------------------------------SOL00050
      SUBROUTINE SOL01                                                        SOL00060
      LOGICAL PRINT,DSPFLG,HEAD,AXIAL,BTEST                                   SOL00070
      CHARACTER*80 NAME                                                       SOL00080
      DIMENSION RINPUT(100)                                                   SOL00090
      INCLUDE (ZCOMN)                                                         SOL00100
C                                                                             SOL00110
      READ (5,*)  NLOAD,MAXELD                                                SOL00120
C                                                                             SOL00130
      WRITE (6,1) TITLE(1),TITLE(2),NLOAD                                     SOL00140
    1 FORMAT ('1 STRUCTURE....: ',A,/                                         SOL00150
     &        '  SOLUTION.....: ',A,//                                       SOL00160
     &  &1X,' SOLUTION #1, STATIC - ELASTIC ANALYSIS '/                       SOL00170
     &  &1X,'=======================================''//                     SOL00180
     &  &1X,' NUMBER OF LOAD CASES ..........',I5)                            SOL00190
C                                                                             SOL00200
      IF (N3(I2KGDT+3).EQ.2) WRITE (6,2)                                      SOL00210
    2 FORMAT (1X,' GEOMETRIC STIFFNESS IS NOT INCLUDED. ',                    SOL00220
     &        ' USE KGDATA(3)=1 TO INCLUDE GEOMETRIC STIFFNESS.')            SOL00230
C                                                                             SOL00240
C       ---- INITIALIZE STORAGE FOR STIFFNESS, LOAD AND DISPL... -----       SOL00250
      ELSTIC=.TRUE.                                                           SOL00260
C                                                                             SOL00270
C       ---- INITIALIZE STORAGE FOR STIFFNESS, LOAD AND DISPL... -----       SOL00280
C               Z(I2STIF) = LOCATION OF THE STIFFNESS MATRIX                  SOL00290
C               Z(I2LOAD) = LOCATION OF THE LOAD MATRIX                       SOL00300
C               Z(I2LOAD) = LOCATION OF THE DISPLACEMENT MATRIX               SOL00310
      IF (I2LOAD.EQ.0) THEN                                                   SOL00320
          I2LOAD = I2                                                         SOL00330
          I2     = I2+NDOF*NLOAD                                              SOL00340
          ENDIF                                                               SOL00350
      IF (I2DISP.EQ.0) THEN                                                   SOL00360
          I2DISP = I2                                                         SOL00370
          I2     = I2+NDOF*NLOAD                                              SOL00380
          ENDIF                                                               SOL00390
       IF (I2VEL .EQ.0) THEN                                                  SOL00400
          I2VEL  = I2                                                         SOL00410
          I2     = I2+NDOF                                                    SOL00420
          ENDIF                                                               SOL00430
          DO 3 I=1,NDOF                                                       SOL00440
    3     Z(I+I2VEL -1) =0                                                    SOL00450
       IF (I2DVEL.EQ.0) THEN                                                  SOL00460
          I2DVEL = I2                                                         SOL00470
          I2     = I2+NDOF                                                    SOL00480
          ENDIF                                                               SOL00490
          DO 4 I=1,NDOF                                                       SOL00500
    4     Z(I+I2DVEL-1) =0                                                    SOL00510
       IF (I2FUB .EQ.0) THEN                                                  SOL00520
          I2FUB  = I2                                                         SOL00530
          I2     = I2+NDOF                                                    SOL00540
          DO 5 I=1,NDOF                                                       SOL00550
    5     Z(I+I2FUB -1) =0                                                    SOL00560
          ENDIF                                                               SOL00570
C                                                                             SOL00580
C       ---- SET UP CONSTANTS ------------------                             SOL00590
        L1=1                                                                  SOL00600
        L2=NCOND +NFREE                                                       SOL00610
        L3=NCOND +NFREE+1                                                     SOL00620
        L4=NDOF                                                               SOL00630
        IF (MAXELD.GT.0) THEN                                                 SOL00640
            I2IELD=I2                                                         SOL00650
            I2=I2 + MAXELD*5                                                  SOL00660
            I2ELD=I2                                                          SOL00670
            I2=I2 + MAXELD*12                                                 SOL00680
        ENDIF                                                                 SOL00690
        IW=I2+1                                                               SOL00700
        IMD=N2(I2LM+L4)                                                       SOL00710
        CALL SOL01A                                                           SOL00720
     &    (L1,L2,L3,L4,IMD,NLOAD,MAXELD,NELD,NNODE,NCOS,NELMT,ABORT,          SOL00730
     &     I2DISP,I2LOAD,MSTOR,IBUG,MAXNOD,TITLE,STEPID,ESE,                  SOL00740
     &     Z(I2STIF), Z(I2LOAD), Z(I2DISP), Z(I2VEL ), Z(I2DVEL),             SOL00750
     &     Z(I2FUB ),N2(I2IELD), Z(I2ELD ), Z(IW    ),N2(I2MD  ),             SOL00760
     &    N2(I2ID  ),N2(I2IDOF), Z(I2CNST),N2(I2JFLG), Z(I2CORD),            SOL00770
     &     Z(I2COS ),N2(I2JCOS),SUMACT)                                       SOL00780
      RETURN                                                                  SOL00790
C                                                                             SOL00800
      END                                                                     SOL00810
C-----------------------------------------------------------------------------SOL00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT                                  SOL00020
C         END DEBUG                                                           SOL00030
```

```
C*********************************************************************SOL00040
      SUBROUTINE SOL01A                                                 SOL00050
     &    (L1,L2,L3,L4,IMD,NLOAD,MAXELD,NELD,NNODE,NCOS,NELMT,ABORT,    SOL00060
     &     I2DISP,I2LOAD, MSTOR,IBUG,MAXNOD,TITLE,STEPID,ESE,           SOL00070
     &     STIF  , LOAD , DISP , VEL  , DVEL ,                          SOL00080
     &     FUB   , IELD , ELD  , WORK , MD   ,                          SOL00090
     &     ID    , IDOF , CONST, JTFLG, COORD,                          SOL00100
     &     COSINE, JTCOS,SUMRCT)                                        SOL00110
      COMMON /ISPEC/ ISP1, SGD(3000,2),REACF(6)                         SOL00120
      CHARACTER*(*) TITLE(2),STEPID                                     SOL00130
      LOGICAL PRINT,DSPFLG,HEAD,AXIAL,BTEST,BUG,ABORT,BUG3              SOL00140
      CHARACTER*80 NAME                                                 SOL00150
      DIMENSION RINPUT(100),SUMRCT(6)                                   SOL00160
      REAL LOAD                                                         SOL00170
      DIMENSION STIF(IMD),LOAD(L4,NLOAD),DISP(L4,NLOAD),VEL(L4),DVEL(L4)SOL00180
      DIMENSION FUB(L4),IELD(MAXELD,5),ELD(MAXELD,12),WORK(1),MD(L4+1)  SOL00190
      DIMENSION COORD(MAXNOD,3),ID(MAXNOD),                             SOL00200
     &          IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),CONST(MAXNOD,6),       SOL00210
     &          JTFLG(MAXNOD),JTCOS(MAXNOD)                             SOL00220
      BUG=BTEST(IBUG,5)                                                 SOL00230
      BUG3=BTEST(IBUG,3)                                                SOL00240
      ISP1=1                                                            SOL00250
C                                                                       SOL00260
C*********************                                                  SOL00270
C* INPUT LOADING DATA **                                               SOL00280
C*********************                                                  SOL00290
C                                                                       SOL00300
C       ----- ZERO LOAD AND DISPLACEMENT MATRIX ----------------------  SOL00310
        DO  6 I=1,L4                                                    SOL00320
        DO  6 J=1,NLOAD                                                 SOL00330
          LOAD(I,J)=0                                                   SOL00340
    6     DISP(I,J)=0                                                   SOL00350
C                                                                       SOL00360
C       ----- INPUT JOINT LOADS ---------------------                  SOL00370
C       JOINT LOADS AND SUPPORT DISPLACEMENTS ARE STORED IN THE         SOL00380
C       DISPLACEMENT MATRIX                                             SOL00390
        CALL JOILOA(BUG3,MAXNOD,L4,NNODE,NLOAD,L2,                      SOL00400
     &        ID,IDOF,CONST,DISP,                                       SOL00410
     &        DSPFLG,JTFLG,TITLE,.FALSE.)                               SOL00420
C                                                                       SOL00430
C       ----- INPUT ELEMENT LOADS -------------------                   SOL00440
C       ELEMENT LOADS ARE STORED IN THE LOAD MATRIX                     SOL00450
        IF (MAXELD.GT.0) THEN                                           SOL00460
          CALL ELELOA(BUG3,L3,NELMT,MAXELD,NELD,L4,NLOAD,               SOL00470
     &          LOAD,IELD,ELD,I2DISP,I2LOAD,NAME,TITLE,.FALSE.)         SOL00480
C                                                                       SOL00490
C         ...ADD ELEMENT LOADS ON FREE JOINTS TO THE JOINT LOADS ON FREE SOL00500
C         ...JOINTS, STORE IN THE DISPLACEMENT MATRIX                   SOL00510
C         ...SAVE ELEMENT LOADS ON ALL JOINTS IN LOAD MATRIX, USED LATER SOL00520
C         ...FOR CALCULATING THE REACTIONS.                             SOL00530
            DO 10 I=L1,L2                                               SOL00540
            DO 10 J=1,NLOAD                                             SOL00550
   10         DISP(I,J)=DISP(I,J)+LOAD(I,J)                            SOL00560
C                                                                       SOL00570
C         ...SWAP THE SIGN OF THE FEF AT REACTIONS                      SOL00580
            DO 15 I=L3,L4                                               SOL00590
            DO 15 J=1,NLOAD                                             SOL00600
   15         LOAD(I,J)= -LOAD(I,J)                                    SOL00610
C  15         DISP(I,J)= -LOAD(I,J)                                    SOL00620
C                                                                       SOL00630
        ENDIF                                                           SOL00640
C*                                                                      SOL00650
        DO 19 I=1,L4                                                    SOL00660
   19     SGD(I,1)=DISP(I,1)                                           SOL00670
C*                                                                      SOL00680
C*************************                                             SOL00690
C* FORMULATE STIFFNESS ****                                            SOL00700
C*************************                                             SOL00710
        CALL FORM(1)                                                    SOL00720
C                                                                       SOL00730
C***********************************************                        SOL00740
C* MODIFY LOADS FOR RESTRAINT DISPLACEMENTS ***                        SOL00750
C***********************************************                        SOL00760
        IF (DSPFLG) THEN                                                SOL00770
          IOPT=1                                                        SOL00780
          CALL REACTN(IOPT,BUG,NNODE,L1,L2,L3,L4,DISP,DISP,             SOL00790
     &    MD,STIF,IMD,L4,NLOAD,1.0,                                     SOL00800
     &    MAXNOD,IDOF,COORD,ID,TITLE,STEPID,.FALSE.,                    SOL00810
     &    NCOS,COSINE,JTCOS,JTFLG,SUMRCT)                               SOL00820
        ENDIF                                                           SOL00830
C                                                                       SOL00840
C*************************                                             SOL00850
C* SOLVE FOR FREE DISPL *****                                          SOL00860
C*************************                                             SOL00870
C                                                                       SOL00880
        IF (BTEST(IBUG,5)) THEN                                         SOL00890
         CALL LMATRX(STIF,MD,L4,IMD,'GLOBAL STIFFNESS DUMP=7')          SOL00900
         CALL WMATRX(DISP,L4,NLOAD,'LOAD MATRIX  DUMP=7')               SOL00910
        ENDIF                                                           SOL00920
C                                                                       SOL00930
        IOPT=1                                                          SOL00940
        CALL MSOLL(IOPT,BTEST(IBUG,6),L2,IMD,L4,NLOAD,L1,L2,            SOL00950
     &             MD,STIF,DISP,WORK,                                   SOL00960
     &            .FALSE.,WORK,WORK,L4,                                 SOL00970
     &            .FALSE.,WORK,WORK,L4,ABORT)                           SOL00980
        IF (ABORT) RETURN                                               SOL00990
C                                                                       SOL01000
        IOPT=3                                                          SOL01010
        CALL MSOLL(IOPT,BTEST(IBUG,6),L2,IMD,L4,NLOAD,L1,L2,            SOL01020
     &             MD,STIF,DISP,WORK,                                   SOL01030
     &            .FALSE.,WORK,WORK,L4,                                 SOL01040
     &            .FALSE.,WORK,WORK,L4,ABORT)                           SOL01050
        IF (ABORT) RETURN                                               SOL01060
C                                                                       SOL01070
        IF (BTEST(IBUG,5))                                              SOL01080
     &    CALL WMATRX(DISP,L4,NLOAD,'DISPLACEMENT  DUMP=7')             SOL01090
C*                                                                      SOL01100
        DO 20 I=1,L4                                                    SOL01110
   20   SGD(I,2)=DISP(I,1)                                             SOL01120
C*                                                                      SOL01130
C                                                                       SOL01140
C************************                                              SOL01150
C* SOLVE FOR REACTIONS ***                                             SOL01160
C************************                                              SOL01170
        IOPT=2                                                          SOL01180
        CALL REACTN(IOPT,BUG,NNODE,L1,L2,L3,L4,LOAD,DISP,               SOL01190
     &    MD,STIF,IMD,L4,NLOAD,1.0,                                     SOL01200
     &    MAXNOD,IDOF,COORD,ID,TITLE,STEPID,.FALSE.,                    SOL01210
     &    NCOS,COSINE,JTCOS,JTFLG,SUMRCT)                               SOL01220
        IF (BTEST(IBUG,5))                                              SOL01230
     &    CALL WMATRX(LOAD,L4,NLOAD,'FINAL LOADS AND REACTIONS')        SOL01240
C                                                                       SOL01250
C*********************************                                     SOL01260
C* PRINT GLOBAL DISPLACEMENTS ***                                      SOL01270
C*********************************                                     SOL01280
        CALL JOIDSP(1,MAXNOD,L4,NNODE,NLOAD,0,L2,ID,                    SOL01290
     &    IDOF,CONST,DISP,                                              SOL01300
     &    TITLE,'DISPLACEMENTS',STEPID,.TRUE.,                          SOL01310
     &    NCOS,COSINE,JTCOS,JTFLG)                                      SOL01320
C                                                                       SOL01330
C*********************************                                     SOL01340
C* PRINT GLOBAL REACTIONS ***                                          SOL01350
C*********************************                                     SOL01360
        IOPT=3                                                          SOL01370
        CALL REACTN(IOPT,.TRUE.,NNODE,L1,L2,L3,L4,LOAD,DISP,            SOL01380
     &    MD,STIF,IMD,L4,NLOAD,1.0,                                     SOL01390
     &    MAXNOD,IDOF,COORD,ID,TITLE,STEPID,.TRUE.,                     SOL01400
     &    NCOS,COSINE,JTCOS,JTFLG,SUMRCT)                               SOL01410
        DO 115 I=1,6                                                    SOL01420
  115     REACF(I)=SUMRCT(I)                                            SOL01430
C                                                                       SOL01440
C*************************************                                 SOL01450
```

```
C* CALCULATE AND PRINT MEMBER FORCES ***                               SOL01460
C*************************************                                 SOL01470
        IOPT=4                                                          SOL01480
        LSTTYP=0                                                        SOL01490
        PRINT=.TRUE.                                                    SOL01500
        HEAD =.TRUE.                                                    SOL01510
        DO 300 IELNO=1,NELMT                                            SOL01520
  300   CALL ELELIB                                                     SOL01530
     &    (IOPT  ,LSTTYP ,PRINT ,IREL,HEAD,   NAME ,EESE,EPSE,DAMAGE,   SOL01540
     &     DUCFAG,                                                      SOL01550
     &     IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,I2DISP,I2LOAD,MSTOR)  SOL01560
C                                                                       SOL01570
C*************************************************                      SOL01580
C* ADD LAST LOAD CASE TO TOTAL MEMBER LOADS ***                        SOL01590
C*************************************************                      SOL01600
C THIS SETS THE TOTAL LOAD IN THE ELEMENT SUBROUTINE EQUAL TO THE       SOL01610
C SUM OF THE TOTAL LOAD AND THE LAST LOAD CASE.                         SOL01620
C TWO MAJOR BENEFITS ARE DERIVED FROM THIS:                             SOL01630
C 1) TOTAL ELEMENT LOADS ARE USED AS INITIAL VALUES FOR BUCKLING AND    SOL01640
C    DYNAMIC LOADINGS.                                                  SOL01650
C 2) TOTAL ELEMENT LOADS ARE USED IN NONLINEAR ANALYSIS                 SOL01660
C                                                                       SOL01670
        IOPT=5                                                          SOL01680
        LSTTYP=0                                                        SOL01690
        PRINT=.FALSE.                                                   SOL01700
        HEAD =.FALSE.                                                   SOL01710
        DO 310 IELNO=1,NELMT                                            SOL01720
  310   CALL ELELIB                                                     SOL01730
     &    (IOPT  ,LSTTYP ,PRINT ,IREL,HEAD,   NAME ,EESE,EPSE,DAMAGE,   SOL01740
     &     DUCFAG,                                                      SOL01750
     &     IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,I2DISP,I2LOAD,MSTOR)  SOL01760
C                                                                       SOL01770
C*********************************                                     SOL01780
C* OBTAIN TOTAL STRAIN ENERGY   ***                                    SOL01790
C*********************************                                     SOL01800
        ESE=0                                                           SOL01810
        DO 1312 IELNO=1,NELMT                                           SOL01820
          CALL ELELIB                                                   SOL01830
     &    (10,LSTTYP ,.FALSE.,IREL,.FALSE.,  NAME ,EESE,EPSE,DAMAGE,    SOL01840
     &     DUCFAG,                                                      SOL01850
     &     IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,I2DISP,I2LOAD,MSTOR)  SOL01860
C         WRITE(54,*)IELNO,EESE                                         SOL01870
 1312     ESE=ESE+EESE                                                 SOL01880
          ESE=ESE/4                                                     SOL01890
        WRITE(6,*)                                                      SOL01900
        WRITE(6,1313)ESE                                                SOL01910
 1313 FORMAT(1X,'THE STATICS STRAIN ENERGY OF SYSTEM =',G12.4/)         SOL01920
        WRITE(6,*) '=========================================='        SOL01930
C                                                                       SOL01940
        IF (NLOAD.EQ.1) RETURN                                          SOL01950
C                                                                       SOL01960
C                                                                       SOL01970
C*************************************                                 SOL01980
        WRITE (6,1000) TITLE(1),TITLE(2)                                SOL01990
 1000 FORMAT ('1 STRUCTURE....: ',A,/                                   SOL02000
     &        '  SOLUTION.....: ',A,//                                 SOL02010
     &       &1X,' MAXIMUN VALUES FOR ALL LOAD CASES '/                 SOL02020
     &       &1X,'==================================')                 SOL02030
C                                                                       SOL02040
C*************************************                                 SOL02050
C* GET AND PRINT MAXIMUM DISPL ***                                     SOL02060
C*************************************                                 SOL02070
C                                                                       SOL02080
        DO 1010 I=1,L4                                                  SOL02090
          WORK(I   )=DISP(I,1)                                          SOL02100
          DO 1010 J=2,NLOAD                                             SOL02110
          IF (ABS(DISP(I,J)).GT.ABS(WORK(I))) WORK(I)=DISP(I,J)         SOL02120
 1010 CONTINUE                                                          SOL02130
        CALL JOIDSP(2,MAXNOD,L4,NNODE,1,0,L2,ID,                        SOL02140
     &  IDOF,CONST,WORK,                                                SOL02150
     &  TITLE,'MAXIMUM DISPLACEMENTS',STEPID,.FALSE.,                   SOL02160
     &  NCOS,COSINE,JTCOS,JTFLG)                                        SOL02170
C                                                                       SOL02180
C*************************************                                 SOL02190
C* GET AND PRINT MAXIMUM REACTIONS ***                                 SOL02200
C*************************************                                 SOL02210
C                                                                       SOL02220
        WRITE (6,1000) TITLE(1),TITLE(2)                                SOL02230
        DO 1020 I=1,L4                                                  SOL02240
          WORK(I   )=LOAD(I,1)                                          SOL02250
          DO 1020 J=2,NLOAD                                             SOL02260
          IF (ABS(LOAD(I,J)).GT.ABS(WORK(I))) WORK(I)=LOAD(I,J)         SOL02270
 1020 CONTINUE                                                          SOL02280
        DO 1030 I=1,6                                                   SOL02290
 1030   SUMRCT(I)=0                                                     SOL02300
        CALL REACTN(4,.TRUE.,NNODE,L1,L2,L3,L4,WORK,DISP.               SOL02310
     &  MD,STIF,IMD,L4,1    ,1.0,                                       SOL02320
     &  MAXNOD,IDOF,COORD,ID,TITLE,STEPID,.FALSE.,                      SOL02330
     &  NCOS,COSINE,JTCOS,JTFLG,SUMRCT)                                 SOL02340
C                                                                       SOL02350
C*************************************                                 SOL02360
C* PRINT MAXIMUN ELEMENT FORCES ***                                    SOL02370
C*************************************                                 SOL02380
C                                                                       SOL02390
        WRITE (6,1000) TITLE(1),TITLE(2)                                SOL02400
        LSTTYP=0                                                        SOL02410
        PRINT=.TRUE.                                                    SOL02420
        HEAD =.FALSE.                                                   SOL02430
        DO 1310 IELNO=1,NELMT                                           SOL02440
 1310   CALL ELELIB                                                     SOL02450
     &    (12  ,LSTTYP ,PRINT ,IREL,HEAD,   NAME ,EESE,EPSE,DAMAGE,     SOL02460
     &     DUCFAG,                                                      SOL02470
     &     IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,I2DISP,I2LOAD,MSTOR)  SOL02480
C                                                                       SOL02490
        RETURN                                                          SOL02500
C                                                                       SOL02510
  220   FORMAT (5X,I5,1P,6G15.6)                                        SOL02520
  200   FORMAT ('1 GLOBAL ',A,'   LOADING #',I5//                       SOL02530
     &  4X,'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ'/)SOL02540
        END                                                             SOL02550
C*********************************************************************SOL02560
C           DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT                          SOL00010
C           END DEBUG                                                   SOL00020
C*********************************************************************SOL00030
      SUBROUTINE SOL02(PLACE,CPU,TCPU,REQCPU)                           SOL00040
      LOGICAL PRINT,DSPFLG,HEAD,REPEAT,SLMASS,AXIAL,TEST,NEWKDT,BTEST   SOL00050
      LOGICAL UNBAL                                                     SOL00060
      DIMENSION RINPUT(100)                                             SOL00070
      CHARACTER*1  NAME                                                 SOL00080
      CHARACTER*6  PLACE                                                SOL00090
      CHARACTER*20 INTEG                                                SOL00100
      INCLUDE (%COMN)                                                   SOL00110
C                                                                       SOL00120
C                                                                       SOL00130
C*******************                                                   SOL00140
C* PRINT HEADER ***                                                    SOL00150
C*******************                                                   SOL00160
      WRITE (6,10) TITLE(1),TITLE(2)                                    SOL00170
   10 FORMAT ('1 STRUCTURE....: ',A,/                                   SOL00180
     &        '  SOLUTION.....: ',A,/)                                 SOL00190
C                                                                       SOL00200
C************************                                              SOL00210
C* INPUT LOADING DATA **                                               SOL00220
C************************                                              SOL00230
C                                                                       SOL00240
      READ (5,*) INTEG,THETA, ELSTIC,UNBAL                             SOL00250
      READ (5,*) IPRNT,IWRITE,MAXACC,SLMASS                            SOL00260
      READ (5,*) TO,DT,TF,GRAV                                          SOL00270
      IPRNT=MAX(IPRNT,1)                                                SOL00280
C                                                                       SOL00290
      IF (TEST(INTEG,'LINEAR' ,.FALSE.)) THEN                           SOL00300
            INTEGO=1                                                    SOL00310
```

```
            WRITE (6,11) 'LINEAR ACCELERATION METHOD',
     &                   '============================'
         ELSE IF (TEST(INTEG,'AVERAGE',.FALSE.)) THEN
            INTEGO=2
            WRITE (6,11) 'AVERAGE ACCELERATION METHOD',
     &                   '============================'
         ELSE IF (TEST(INTEG,'WILSON' ,.FALSE.)) THEN
            INTEGO=3
            WRITE (6,11) 'WILSON THETA METHOD',
     &                   '====================='
            WRITE (6,12) THETA
         ELSE IF (TEST(INTEG,'WILSON' ,.FALSE.)) THEN
            WRITE (6,*) 'INVALID INTEGRATION METHOD'
            WRITE (6,11) INTEG,'====================='
            RETURN
         ENDIF
C
C------- CHECK FOR INVALID INTEGRATION WITH PROPORTIONAL DAMPING
         IF ((INTEGO.EQ.1 .OR. INTEGO.EQ.2) .AND. FDAMP.NE.1) THEN
            WRITE (6,19)
     19     FORMAT (5X,'INTEGRATION METHOD IS ONLY VALID FOR ',
     &             'PROPORTIONAL DAMPING'//,5X,'SOLN IS ABORTED')
            RETURN
         ENDIF
C
         WRITE (6,13) TO,DT,TF,GRAV,IPRNT
         IF (ELSTIC) WRITE (6,14)
C
     11  FORMAT (
       &1X,' SOLUTION #2, ',A,' OF NUMERICAL INTEGRATION'/
       &1X,'==============',A,'============================'//)
     12  FORMAT (
       &1X,' THETA.............................',1P,G15.6)
     13  FORMAT (
       &1X,' INITIAL TIME .....................',1P,G15.6/
       &1X,' TIME STEP ........................',1P,G15.6/
       &1X,' FINAL TIME .......................',1P,G15.6/
       &1X,' ACCELERATION DUE TO GRAVITY ......',1P,G15.6/
       &1X,' STEP INTERVAL FOR PRINTING .......',0P,I5)
     14  FORMAT (
       &1X,' THE STRUCTURE IS ASSUMED TO BEHAVE ELASTICALLY')
C
C=================================
C= SET GEOMETRIC STIFFNESS FLAGS ===
C=================================
         KGLOAD=N%(I%KGDT+1)
         KGTYPE=N%(I%KGDT+2)
         KGFORM=N%(I%KGDT+3)
         NEWKG=.TRUE.
         IF (KGLOAD.EQ.0 .OR. KGTYPE.EQ.0 .OR. KGFORM.EQ.0) NEWKG=.FALSE.
C
C        ---- SET UP CONSTANTS -------------------
         L1=1
         L2=NCOND
         L3=L2+1
         L4=NCOND +NFREE
         L5=L4+1
         L6=NDOF
C
C        ---- INITIALIZE STORAGE FOR LOAD, DISPL, VELOC, ACCEL
C        Z(I%ZLOAD) = LOCATION OF THE TOTAL LOAD MATRIX
C        Z(I%DISP) = LOCATION OF THE TOTAL DISPLACEMENT MATRIX
C        Z(I%VEL ) = LOCATION OF THE TOTAL VELOCITY MATRIX
C        Z(I%ACC ) = LOCATION OF THE TOTAL ACCELERATION MATRIX
C        Z(I%DLOA) = LOCATION OF THE INCRIMENTAL LOAD MATRIX
C        Z(I%DDSP) = LOCATION OF THE INCRIMENTAL DISPLACEMENT MATRIX
C        Z(I%DVEL) = LOCATION OF THE INCRIMENTAL VELOCITY MATRIX
C        Z(I%DACC) = LOCATION OF THE INCRIMENTAL ACCELERATION MATRIX
C
         IF (I%ZLOAD.EQ.0) THEN
            I%LOAD = I%
            I%     = I%+NDOF
            CALL CKSTOR(0,0)
            DO 1 I=1,NDOF
     1        Z(I+I%LOAD-1) =0
            ENDIF
C
         IF (I%DISP.EQ.0) THEN
            I%DISP = I%
            I%     = I%+NDOF
            CALL CKSTOR(0,0)
            DO 2 I=1,NDOF
     2        Z(I+I%DISP-1) =0
            ENDIF
C
         IF (I%VEL .EQ.0) THEN
            I%VEL  = I%
            I%     = I%+NDOF
            CALL CKSTOR(0,0)
            DO 3 I=1,NDOF
     3        Z(I+I%VEL -1) =0
            ENDIF
C
         IF (I%ACC .EQ.0) THEN
            I%ACC  = I%
            I%     = I%+NDOF
            CALL CKSTOR(0,0)
            DO 4 I=1,NDOF
     4        Z(I+I%ACC-1) =0
            ENDIF
C
         IF (I%DLOA.EQ.0) THEN
            I%DLOA = I%
            I%     = I%+NDOF
            CALL CKSTOR(0,0)
            DO 5 I=1,NDOF
     5        Z(I+I%DLOA-1) =0
            ENDIF
         IF (I%DDSP.EQ.0) THEN
            I%DDSP = I%
            I%     = I%+NDOF
            CALL CKSTOR(0,0)
            DO 6 I=1,NDOF
     6        Z(I+I%DDSP-1) =0
            ENDIF
         IF (I%DVEL.EQ.0) THEN
            I%DVEL = I%
            I%     = I%+NDOF
            CALL CKSTOR(0,0)
            DO 7 I=1,NDOF
     7        Z(I+I%DVEL-1) =0
            ENDIF
         IF (I%DACC.EQ.0) THEN
            I%DACC = I%
            I%     = I%+NDOF
            CALL CKSTOR(0,0)
            DO 8 I=1,NDOF
     8        Z(I+I%DACC-1) =0
            ENDIF
         IF (I%FUB .EQ.0) THEN
            I%FUB  = I%
            I%     = I%+NDOF
            CALL CKSTOR(0,0)
            DO 9 I=1,NDOF
     9        Z(I+I%FUB-1) =0
            ENDIF
         IW=I%+1
C
C---------- FORM MASS MATRIX -----
C
         IMDMS =N%(I%MDMS+NDOF)
         I%MASS=I%MASS
         DO 400 I=1,IMDMS
   400      Z(I+I%MASS-1)=0.
C
         CALL MFORM(2,INMASS,NMASS,Z(I%MASS),N%(I%MDMS),MD,NDOF,IMDMS,
     &              TITLE,NNODE,MAXNOD,NCOS,BTEST(IBUG,5),IBUG,
     &              N%(I%ID),Z(I%COS),N%(I%IDOF),Z(I%CNST),N%(I%JFLG),
     &              N%(I%JCOS),Z(I%MDAT),Z(IW),N%(IW+21),Z(IW+27),
     &               Z(IW+ 63),Z(IW+ 99))
C
C---------- FORM SPECIAL MASS MATRIX -----
C
         IF (SLMASS) THEN
            READ (5,*) INMASS,IFMASS
C           INMASS=INMASS+1
            CALL MASSIN(INMASS,IFMASS,.FALSE.,BUG,IBUG,TITLE,MAXZ,MAXDZ,
     &      MAXNOD,NCOS,NNODE,IZ,IW,IZID,Z,NZ,DZ,
     &      I%SLMD,I%SLMS,I%LM,I%KE,IZMDT2,
     &      NDOF,NMASS,IZCOS,I%IDOF,IZCNST,IZJFLG,IZJCOS,'GLOBAL  MASS',
     &      NCOND,NFREE,IZMD)
            IMDMSL=N%(I%SLMD+NDOF)
         ELSE
            I%SLMS=0
            I%GLMD=0
            IMDMSL=1
         ENDIF
         IMDMS =N%(I%MDMS+NDOF)
C
C------- SET UP STORAGE FOR MAIN DIAGONAL ADDRESS OF DYNAMIC STIFFNESS
         I%MDS= IZ
         I%   = IZ + NFREE +1
         CALL CKSTOR(0,IW)
C
C------- DEVELOP SKYLINE OF DYNAMIC STIFFNESS (S)
         CALL SKYLN2(NDOF,L3,L4,
     &      N%(I%MDS),N%(IZMD),N%(IZMDMS),N%(IZMDKG),KGFORM)
C
C---- DETERMINE THE LENGTH OF THE DYNAMIC STIFFNESS MATRIX
         IMDS=N%(IZMDS+NFREE) - N%(IZMDS) +1
C
C---- DETERMINE THE ADDRESS OF THE DYNAMIC STIFFNESS MATRIX
         I%S    = I%
         I%     = I%+N%(I%MDS+NFREE)
C
C---------- DEFINE STORAGE REQD FOR NUMERICAL INTEGRATION
C          AND INITILIZE VALUES
         I%AN   = I%
         I%     = I%+NFREE
         I%BN   = I%
         I%     = I%+NFREE
         I%PBAR = I%
         I%     = I%+NFREE
C
C---- SAVE STORAGE FOR DYNAMIC GROUND ACCELERATIONS
            IZAGTX=I%
            I%    =I%+MAXACC
            IZAGTY=I%
            I%    =I%+MAXACC
            IZAGTZ=I%
            I%    =I%+MAXACC
            IZAINP=I%
            NAINP =MAX(NDOF,MAXACC)
            I%    =I%+NAINP
            I%MCOS=I%
            I%    =I%+NDOF*3
C
C---- SAVE STORAGE FOR ENERGY CALCULATIONS
            I%%ERD=I%
            I%    =I%+NDOF
            I%DM  =I%
            I%    =I%+NDOF
            I%VM  =I%
            I%    =I%+NDOF
            I%DBAR=I%
            I%    =I%+NDOF
C           I%LO2 =I%
            I%    =I%+NDOF
            I%LO2=I%
            I%    =I%+NDOF
            I%TFD =I%
            I%    =I%+NDOF
            I%DFD =I%
            I%    =I%+NDOF
            I%FG  =I%
            I%    =I%+NDOF
            I%DFG =I%
            I%    =I%+NDOF
         DO 25 I=0,NDOF-1
           Z(I%FG  +I)=0
           Z(I%DFG +I)=0
           Z(I%%ERG+I)=0
           Z(I%DM  +I)=0
           Z(I%VM  +I)=0
           Z(I%DBAR+I)=0
C          Z(I%LO2 +I)=0
           Z(I%LO2+I)=0
           Z(I%TFD +I)=0
     25    Z(I%DFD +I)=0
C
C=====================================
         IZDMAX=I%
         I%    =I%+NDOF
         I%VMAX=I%
         I%    =I%+NDOF
         I%AMAX=I%
         I%    =I%+NDOF
         I%RMAX=I%
         I%    =I%+NDOF +6
         DO 28 I=0,NDOF
           Z(IZDMAX+I)=0
           Z(IZVMAX+I)=0
           Z(I%AMAX+I)=0
     28    Z(I%RMAX+I)=0
         DO 29 I=1,6
     29    Z(I%RMAX-1+NDOF+I)=0
         DO 30 I=1,6
     30    SUMRCT(I)=0
C
C=====================================
C=INPUT FILE OUTPUT CONTROLL DATA ===
C=====================================
         IF (IWRITE.GT.0) THEN
            IOPT=1
            CALL DMPDAT(IOPT,IWRITE,TO,DT)
         ENDIF
C
C------ SET UP WORK STORAGE AREA ...
         IW     = IZ + 1
C
C------ CHECK STORAGE REQUIRMENTS ...
         CALL CKSTOR(NDOF,IW)
C
C---- DETERMINE THE LENGTH OF THE STIFFNESS MATRIX
               IMD=N%(I%MD+NDOF) - N%(IZMD) +1
C---- DETERMINE THE LENGTH OF THE MASS MATRIX
               IMDMS=N%(I%MDMS+NDOF) - N%(IZMDMS) +1
C---- DETERMINE THE LENGTH OF THE GEOMETRIC STIFFNESS MATRIX
               IF (N%(IZKGDT+3).EQ.2) THEN
                  IMDKG =N%(IZMDKG+NDOF) - N%(IZMDKG) +1
               ELSE
                  IMDKG =1
               ENDIF
```

```
C                                                                       SOL03160
C------ CALL SOL2A TO CALC DATA ....                                    SOL03170
      CALL SOL02A                                                       SOL03180
     & (L1,L2,L3,L4,L5,L6,IMD,IMDMS,IMDKG,IMDS,IMDMSL,ABORT,ITLCPU,     SOL03190
     &  PLACE,CPU,TCPU,REQCPU,                                          SOL03200
     &  NLOAD    , MAXELD   , NNODE    , NCOS     , NELMT    ,         SOL03210
     &  I&DISP   , I&LOAD   , MSTOR    , IBUG     , MAXNOD   ,         SOL03220
     &  TITLE    , STEPID   , NAINP    , NDOF     , NFREE    ,         SOL03230
     &  KGFORM   , KGLOAD   , KGTYPE   , NCOND    , INTEGO   ,         SOL03240
     &  NEWKG    , NEWK     , MCOND    , KGCOND   , UNBAL    ,         SOL03250
     &  INTEG    , THETA    , ELSTIC   , IPRNT    , IWRITE   ,GROUND(1),SOL03260
     &  MAXACC   , SLMASS   , TO       , DT       , TF       ,GROUND(4),SOL03270
     &  GRAV     , FDAMP    , FMASS    , I&DDSP   ,I&DLOA    ,GROUND(7),SOL03280
     &  &(I&DISP), &(I&VEL ), &(I&ACC ), &(I&LOAD), &(I&AN  ),&(I&DMAX),SOL03290
     &  &(I&DDSP), &(I&DVEL), &(I&DACC), &(I&DLOA), &(I&BN  ),&(I&VMAX),SOL03300
     &  &(I&STIF), &(I&MASS), &(I&SLMS), &(I&KG  ), &(I&S   ),&(I&AMAX),SOL03310
     & N&(I&MD  ),N&(I&MDMS),N&(I&SLMD),N&(I&MDKG),N&(I&MDS ),&(I&RMAX),SOL03320
     &  &(I&DAMP), &(I&PBAR), &(I&MCOS), &(I&AGTX),          &(I&DLO2),SOL03330
     &  &(I&AGTY), &(I&AGTZ), &(I&AINP), &(I&FUB ), &(I&W   ),  SUMRCT ,SOL03340
     & N&(I&ID  ),N&(I&IDOF), &(I&CNST),N&(I&JFLG), &(I&COORD),&(I&DFD ),SOL03350
     &  &(I&COS ),N&(I&JCOS), &(I&TFD ),                                SOL03360
     &  EIE,ESE,PSE,EKE,EDD,                                            SOL03370
     &  &(I&ZERO), &(I&FG  ) , &(I&DFG  )  )                            SOL03380
C                                                                       SOL03390
      RETURN                                                            SOL03400
C                                                                       SOL03410
      END                                                               SOL03420
C*********************************************************************SOL00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT                            SOL00020
C         END DEBUG                                                     SOL00030
C*********************************************************************SOL00040
      SUBROUTINE SOL02A                                                 SOL00050
     & (L1,L2,L3,L4,L5,L6,IMD,IMDMS,IMDKG,IMDS,IMDMSL,ABORT,ITLCPU,     SOL00060
     &  PLACE,CPU,TCPU,REQCPU,                                          SOL00070
     &  NLOAD    , MAXELD   , NNODE    , NCOS     , NELMT    ,         SOL00080
     &  I&DISP   , I&LOAD   , MSTOR    , IBUG     , MAXNOD   ,         SOL00090
     &  TITLE    , STEPID   , NAINP    , NDOF     , NFREE    ,         SOL00100
     &  KGFORM   , KGLOAD   , KGTYPE   , NCOND    , INTEGO   ,         SOL00110
     &  NEWKG    , NEWK     , MCOND    , KGCOND   , UNBAL    ,         SOL00120
     &  INTEG    , THETA    , ELSTIC   , IPRNT    , IWRITE   , GAT    ,SOL00130
     &  MAXACC   , SLMASS   , TO       , DT       , TF       , GVT    ,SOL00140
     &  GRAV     , FDAMP    , FMASS    , I&DDSP   ,I&DLOA    , GDT    ,SOL00150
     &  DISP     , VEL      , ACC      , DLOAD    , AN       , DSPMAX ,SOL00160
     &  DDISP    , DVEL     , DACC     , DLOAD    , BN       , VELMAX ,SOL00170
     &  STIFF    , MASS     , MASSSL   , KG       , SBAR     , ACCMAX ,SOL00180
     &  MD       , MDMS     , MDMSSL   , MDKG     , MDS      , RCTMAX ,SOL00190
     &  DAMP     , PBAR     , MCOS     , AGTX     ,          DLOAD2,  SOL00200
     &  AGTT     , AGTZ     , AINP     , FUB      , WORK     , SUMRC  ,SOL00210
     &  ID       , IDOF     , CNST     , JTFLG    , COORD    , TFDAMP,SOL00220
     &  COSINE   , JTCOS    , DFDAMP   ,                                SOL00230
     &  EIE,ESE,PSE,EKE,EDD,                                            SOL00240
     &    ZERO   , FG       , DFG      )                               SOL00250
C                                                                       SOL00260
      DOUBLE PRECISION  EIE,ESE,PSE,EDD,EKE                             SOL00270
      COMMON /ISPEC/ ISP1,SGD(3000,2),REACF(6)                          SOL00280
      COMMON /IDSTEP/ ISTEP                                             SOL00290
      CHARACTER*(*) TITLE(2),STEPID                                     SOL00300
      LOGICAL PRINT,DSPFLG,HEAD,REPEAT,SLMASS,AXIAL,TEST,NEWKDT,BTEST   SOL00310
      LOGICAL NEWKG,NEWK,ELSTIC,MCOND,KGCOND,UNBAL,MCND2,KGCND2,WRITE   SOL00320
      LOGICAL ABORT,BUG3,BUG4,BUG5,BUG6,BUG11                           SOL00330
      LOGICAL DUCLOG                                                    SOL00340
      CHARACTER*80 NAME                                                 SOL00350
      CHARACTER*20 INTEG                                                SOL00360
      CHARACTER*6  PLACE                                                SOL00370
      REAL LOAD,MASS,MASSSL,KG,MCOS,KE                                  SOL00380
      INTEGER FDAMP,FMASS                                               SOL00390
      INTEGER ITLCPU,CPUREM,CPUELA                                      SOL00400
      DIMENSION ZERO(L6), DLOAD2(L6), FG(L6), DFG(L6)                   SOL00410
      DIMENSION SUMRC(6),DSUMRC(6),SUMFD(6),DSUMFD(6),SUMUB(6)          SOL00420
      DIMENSION TFDAMP(L6),DFDAMP(L6)                                   SOL00430
      DIMENSION DSPMAX(L6),VELMAX(L6),ACCMAX(L6),RCTMAX(L6+6)           SOL00440
      DIMENSION  GD(3), GV(3), GA(3)                                    SOL00450
      DIMENSION GDT(3),GVT(3),GAT(3), TSUM(6)                           SOL00460
      DIMENSION RINPUT(100)                                             SOL00470
      DIMENSION STIFF(IMD) ,MASS(IMDMS),  KG(IMDKG),SBAR(IMDS),         SOL00480
     &          MASSSL(IMDMSL)                                          SOL00490
      DIMENSION   MD(L6+1),MDMS(L6+1 ),MDKG(L6+1 ), MDS(L6+1),          SOL00500
     &          MDMSSL(L6+1)                                            SOL00510
      DIMENSION  LOAD(L6), DISP(L6), VEL(L6), ACC(L6), AN(L6)           SOL00520
      DIMENSION DLOAD(L6),DDISP(L6),DVEL(L6),DACC(L6), BN(L6)           SOL00530
      DIMENSION   FUB(L6), PBAR(L6),MCOS(L6,3),DAMP(3)                  SOL00540
      DIMENSION WORK(2*L6)                                              SOL00550
      DIMENSION AGTX(MAXACC),AGTY(MAXACC),AGTZ(MAXACC),AINP(NAINP)      SOL00560
      DIMENSION COORD(MAXNOD,3),ID(MAXNOD),                             SOL00570
     &          IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),CNST(MAXNOD,6),        SOL00580
     &          JTFLG(MAXNOD),JTCOS(MAXNOD)                             SOL00590
      SUMUB(1)=0                                                        SOL00600
      SUMUB(2)=0                                                        SOL00610
      SUMUB(3)=0                                                        SOL00620
      SUMUB(4)=0                                                        SOL00630
      SUMUB(5)=0                                                        SOL00640
      SUMUB(6)=0                                                        SOL00650
      SUMFOR=0                                                          SOL00660
      SUMMOM=0                                                          SOL00670
      SUMFO=0                                                           SOL00680
      SUMMO=0                                                           SOL00690
      DUCLOG= .FALSE.                                                   SOL00700
      DUCFAG= -1                                                        SOL00710
C-----------------------------------------                              SOL00720
C           WRITE (6,*) '<----- DDISP -----> AT SUBR '                  SOL00730
C           WRITE (6,*) (DDISP(I),I=1,14)                               SOL00740
      BUG3 =BTEST(IBUG, 3)                                              SOL00750
      BUG4 =BTEST(IBUG, 4)                                              SOL00760
      BUG5 =BTEST(IBUG, 5)                                              SOL00770
      BUG6 =BTEST(IBUG, 6)                                              SOL00780
      BUG11=BTEST(IBUG,11)                                              SOL00790
C-----------------------------------------                              SOL00800
        IF (BUG5) THEN                                                  SOL00810
          CALL LMATRX(MASS,MDMS,NDOF,IMDMS,'MASS MATRIX DUMP=7')        SOL00820
        ENDIF                                                           SOL00830
C                                                                       SOL00840
C-----------------------------------------------                        SOL00850
C= INITIALIZE GROUND ACCELERATION CONSTANTS ===                         SOL00860
C-----------------------------------------------                        SOL00870
      DO 91 I=1,3                                                       SOL00880
        GD(I)=0                                                         SOL00890
        GV(I)=0                                                         SOL00900
        GA(I)=0                                                         SOL00910
        GDT(I)=0                                                        SOL00920
        GVT(I)=0                                                        SOL00930
   91   GAT(I)=0                                                        SOL00940
      ACCOLD=0                                                          SOL00950
      XGACC=0                                                           SOL00960
      YGACC=0                                                           SOL00970
      ZGACC=0                                                           SOL00980
C                                                                       SOL00990
C---------------------------------                                      SOL01000
C= INITIALIZE ENERGY CONSTANTS ===                                      SOL01010
C---------------------------------                                      SOL01020
C     EIE=0                                                             SOL01030
C     ESE=0                                                             SOL01040
      PSE=0                                                             SOL01050
      EKE=0                                                             SOL01060
      EDD=0                                                             SOL01070
      EIEM =0                                                           SOL01080
      ESEM =0                                                           SOL01090
      PSEM =0                                                           SOL01100
      EKEM =0                                                           SOL01110
      EDDM =0                                                           SOL01120
      DO 92 I=1,6                                                       SOL01130
C         SUMRC(I)=0                                                    SOL01140
   92     SUMFD(I)=0                                                    SOL01150
C                                                                       SOL01160
C-----------------------------------                                    SOL01170
C= SET GEOMETRIC STIFFNESS FLAGS ===                                    SOL01180
C-----------------------------------                                    SOL01190
      NEWKG=.TRUE.                                                      SOL01200
      IF (KGLOAD.EQ.0 .OR. KGTYPE.EQ.0 .OR. KGFORM.EQ.0) NEWKG=.FALSE.  SOL01210
C                                                                       SOL01220
C------------------------                                               SOL01230
C= SET STIFFNESS FLAGS ===                                              SOL01240
C------------------------                                               SOL01250
      NEWK = .TRUE.                                                     SOL01260
      KSAME= 0                                                          SOL01270
C                                                                       SOL01280
C       ---- INPUT INITIAL VALUES FOR DISPL, VELOC, ACCEL               SOL01290
      CALL INTDVA(BUG3,NDOF,MAXNOD,NNODE,TITLE,IDOF,                    SOL01300
     &  ID,JTFLG,DISP  ,VEL   ,ACC   ,LOAD  )                           SOL01310
C                                                                       SOL01320
      IF( ISP1 .EQ. 1 ) THEN                                            SOL01330
          ESE=ESE                                                       SOL01340
          EIE=ESE                                                       SOL01350
          DO 95 I=1,6                                                   SOL01360
   95       REACF(I)=REACF(I)                                          SOL01370
      ELSE IF( ISP1 .EQ. 0 ) THEN                                       SOL01380
          ESE=0                                                         SOL01390
          EIE=0                                                         SOL01400
          DO 96 I=1,6                                                   SOL01410
   96       REACF(I)=0                                                 SOL01420
      ENDIF                                                             SOL01430
C                                                                       SOL01440
C---- INITILIZE TIME TO CALC INITIAL GROUND ACCELERATION                SOL01450
      T=TO                                                              SOL01460
      TL=T-DT                                                           SOL01470
C                                                                       SOL01480
C---- READ AND INITILIZE DYNAMIC GROUND ACCELERATIONS                   SOL01490
      IF (SLMASS) THEN                                                  SOL01500
        CALL DYLOA(1,BUG3,MAXACC,NDOF,NAINP,T,TL,                       SOL01510
     &  ACCEL&,TOACC,DTACC,GRAV,                                        SOL01520
     &  MAXNOD,NNODE,COSINE,IDOF,JTFLG,JTCOS,NCOS,                      SOL01530
     &  AGTX  ,AGTY  ,AGTZ  ,MASSSL,MDMSSL,IMDMSL,                      SOL01540
     &  AINP  ,MCOS  ,DLOAD ,GA    ,GAT    )                           SOL01550
      ELSE                                                              SOL01560
        CALL DYLOA(1,BUG3,MAXACC,NDOF,NAINP,T,TL,                       SOL01570
     &  ACCEL&,TOACC,DTACC,GRAV,                                        SOL01580
     &  MAXNOD,NNODE,COSINE,IDOF,JTFLG,JTCOS,NCOS,                      SOL01590
     &  AGTX  ,AGTY  ,AGTZ  ,MASS  ,MDMS  ,IMDMS,                       SOL01600
     &  AINP  ,MCOS  ,DLOAD ,GA    ,GAT    )                           SOL01610
      ENDIF                                                             SOL01620
C                                                                       SOL01630
C---- GET INITIAL DYNAMIC LOAD                                          SOL01640
      IF (SLMASS) THEN                                                  SOL01650
        CALL DYLOA(2,BUG3,MAXACC,NDOF,NAINP,T,TL,                       SOL01660
     &  ACCEL&,TOACC,DTACC,GRAV,                                        SOL01670
     &  MAXNOD,NNODE,COSINE,IDOF,JTFLG,JTCOS,NCOS,                      SOL01680
     &  AGTX  ,AGTY  ,AGTZ  ,MASSSL,MDMSSL,IMDMSL,                      SOL01690
     &  AINP  ,MCOS  , LOAD ,GA    ,GAT    )                           SOL01700
      ELSE                                                              SOL01710
        CALL DYLOA(2,BUG3,MAXACC,NDOF,NAINP,T,TL,                       SOL01720
     &  ACCEL&,TOACC,DTACC,GRAV,                                        SOL01730
     &  MAXNOD,NNODE,COSINE,IDOF,JTFLG,JTCOS,NCOS,                      SOL01740
     &  AGTX  ,AGTY  ,AGTZ  ,MASS  ,MDMS  ,IMDMS,                       SOL01750
     &  AINP  ,MCOS  , LOAD ,GA    ,GAT    )                           SOL01760
      ENDIF                                                             SOL01770
C                                                                       SOL01780
C---- INITILIZE INTEGRATION ROUTINES                                    SOL01790
C                                                                       SOL01800
      IF (INTEGO.EQ.1) THEN                                             SOL01810
        CALL LINACC(1,BUG4,DT,TO,TF,T,1.00,NEWKDT,                      SOL01820
     &    L1,L2,L3,L4,L5,L6,IMDMS,IMD,IMDS,IMDKG,NDOF,NFREE,            SOL01830
     &    KGLOAD,KGFORM,                                                SOL01840
     &     ACC, VEL   , DISP  , LOAD  ,                                 SOL01850
     &     DACC  , DVEL  , DDISP , DLOAD2,WORK,                         SOL01860
     &     MASS  ,MDMS   , DAMP  , STIFF ,MD       ,                   SOL01870
     &     KG    ,MDKG   , FZ    ,GRAV,                                 SOL01880
     &     SBAR  ,MDS    , AN    , BN    , PBAR  ,                     SOL01890
     &    FDAMP,FMASS,C1,C2,C3,C4,C5,C6,C7,ALPHA,BETA,C10,C11,ABORT)    SOL01900
      ELSE IF (INTEGO.EQ.2) THEN                                        SOL01910
        CALL AVEACC(1,BUG4,DT,TO,TF,T,THETA,NEWKDT,                     SOL01920
     &    L1,L2,L3,L4,L5,L6,IMDMS,IMD,IMDS,IMDKG,NDOF,NFREE,            SOL01930
     &    KGLOAD,KGFORM,                                                SOL01940
     &     ACC, VEL   , DISP  , LOAD  ,                                 SOL01950
     &     DACC  , DVEL  , DDISP , DLOAD2,WORK,                         SOL01960
     &     MASS  ,MDMS   , DAMP  , STIFF ,MD       ,                   SOL01970
     &     KG    ,MDKG   , FZ    ,GRAV,                                 SOL01980
     &     SBAR  ,MDS    , AN    , BN    , PBAR  ,                     SOL01990
     &    FDAMP,FMASS,C1,C2,C3,C4,C5,C6,C7,ALPHA,BETA,C10,C11,ABORT)    SOL02000
      ELSE IF (INTEGO.EQ.3) THEN                                        SOL02010
        CALL WILSON(1,BUG4,DT,TO,TF,T,THETA,NEWKDT,                     SOL02020
     &    L1,L2,L3,L4,L5,L6,IMDMS,IMD,IMDS,IMDKG,NDOF,NFREE,            SOL02030
     &    KGLOAD,KGFORM,                                                SOL02040
     &     ACC, VEL   , DISP  , LOAD  ,                                 SOL02050
     &     DACC  , DVEL  , DDISP , DLOAD2,WORK,                         SOL02060
     &     MASS  ,MDMS   , DAMP  , STIFF ,MD       ,                   SOL02070
     &     KG    ,MDKG   , FZ    ,GRAV,                                 SOL02080
     &     SBAR  ,MDS    , AN    , BN    , PBAR  ,                     SOL02090
     &    FDAMP,FMASS,C1,C2,C3,C4,C5,C6,C7,ALPHA,BETA,C10,C11,ABORT)    SOL02100
      ENDIF                                                             SOL02110
      IF (ABORT) GO TO 5000                                             SOL02120
C                                                                       SOL02130
C---------------------------------------                                SOL02140
C=WRITE INITIAL DATA TO OUTPUT FILE ===                                 SOL02150
C---------------------------------------                                SOL02160
      IF (IWRITE.GT.0) THEN                                             SOL02170
         IOPT=2                                                         SOL02180
         CALL DMPDAT(IOPT,IWRITE,TO,DT)                                 SOL02190
      ENDIF                                                             SOL02200
C                                                                       SOL02210
C*************************************************************************SOL02220
C= LOOP FOR EACH TIME STEP , ISTEP=CURRENT STEP # ======================SOL02230
C*************************************************************************SOL02240
C... NSTEP = NUMBER OF TIME STEPS                                       SOL02250
C... ISTEP = TIME STEP NUMBER                                           SOL02260
C... T     = CURRENT TIME                                               SOL02270
C... TL    = TIME AT PREVIOUS TIME STEP                                 SOL02280
C... TO    = INITIAL TIME                                               SOL02290
C... DT    = TIME INCRIMENT                                             SOL02300
C... TL    = FINAL TIME                                                 SOL02310
C                                                                       SOL02320
      MCND2 = MCOND                                                     SOL02330
      NSTEP = (TF-TO)/DT                                                SOL02340
      T     = TO                                                        SOL02350
      DO 500 ISTEP=1,NSTEP                                              SOL02360
        T =TO+ISTEP*DT                                                  SOL02370
        TL=T - DT                                                       SOL02380
        WRITE (STEPID,90) ISTEP,T                                       SOL02390
C*      WRITE (6,90) ISTEP,T                                            SOL02400
   90   FORMAT ('STEP:',I6,',  TIME:',1P,G13.5)                         SOL02410
C                                                                       SOL02420
C-----------------------------                                          SOL02430
C= CALC INCRIMENTAL LOADS  ==                                           SOL02440
C-----------------------------                                          SOL02450
      I2=2                                                              SOL02460
      CALL DYLOA(2,BUG3,MAXACC,NDOF,NAINP,T,TL,                         SOL02470
     &    ACCEL&,TOACC,DTACC,GRAV,                                      SOL02480
     &  MAXNOD,NNODE,COSINE,IDOF,JTFLG,JTCOS,NCOS,                      SOL02490
     &  AGTX  ,AGTY  ,AGTZ  ,MASS  ,MDMS   ,IMDMS,                      SOL02500
     &  AINP  ,MCOS  ,DLOAD ,GA    ,GAT    )                           SOL02510
      F&     =ACCEL&                                                    SOL02520
C                                                                       SOL02530
C---- ADD UNBALANCED LOAD'S TO THE LOAD VECTOR. ---> STORE AS DLOAD2    SOL02540
C     STORE OLD UNBALANCED LOADS AT REACTIONS IN DLOAD                  SOL02550
      IF (UNBAL) THEN                                                   SOL02560
         DO 195 I=L1,L4                                                 SOL02570
```

```
               DLOAD2(I)=DLOAD(I)+FUB(I)                                SOL02580
C              WRITE(50,*)DLOAD(1),FUB(1),DLOAD2(1)                     SOL02590
  195          FUB(I)=0.0                                               SOL02600
            DO 196 I=L5,L6                                              SOL02610
               DLOAD2(I)=-FUB(I)                                        SOL02620
C***                                                                    SOL02630
C***    OMIT UNBALANCED LOAD FROM REACTIONS...                          SOL02640
C***    UNBALANCED LOADS ADDED TO THE FRAME ARE RESISTED BY             SOL02650
C***    1) INERTIAL FORCES                                              SOL02660
C***    2) DAMPING FORCES                                               SOL02670
C***    3) MEMBER FORCES                                                SOL02680
C***    SUBTRACTING UNBALANCED FORCES FROM REACTIONS DUE TO MEMBER FORCESSOL02690
C***    NEGLECTS THE REACTIONS DUE TO INERTIAL AND DAMPING FORCES.       SOL02700
C***            DLOAD(I)= -FUB(I)                                       SOL02710
C***                                                                    SOL02720
  196          FUB(I)=0.0                                               SOL02730
         ELSE                                                           SOL02740
            DO 197 I=L1,L6                                              SOL02750
  197          DLOAD2(I)=DLOAD(I)                                       SOL02760
         ENDIF                                                          SOL02770
C                                                                       SOL02780
C============================                                          SOL02790
C= FORMULATE STIFFNESS ====                                            SOL02800
C============================                                          SOL02810
C  FORMULATE THE STIFFNESS IF THE FLAG NEWK IS TRUE, THIS FLAG IS       SOL02820
C       INITIALIZED ABOVE. AND SET WHEN THE ELEMENT FORCES ARE CALC.    SOL02830
C       THE STIFFNESS IS ALSO FORMULATED IF KG IS TO BE CALCULATED      SOL02840
C       AND CONDENSATION IS TO BE PREFORMED, (THE STIFFNESS IS NEEDED   SOL02850
C       FOR THE CONDENSATION PROCEDURE...)                              SOL02860
         REPEAT=.TRUE.                                                  SOL02870
         KSAME=0                                                        SOL02880
  100 IF (NEWK) THEN                                                    SOL02890
            I1=1                                                        SOL02900
            CALL FORM(I1)                                               SOL02910
C------ CALL LMATRX(STIFF,MD,L4,IMD,'GLOBAL STIFFNESS MATRIX')         SOL02920
            IF (BUG5) THEN                                              SOL02930
              CALL LMATRX(STIFF,MD,NDOF,IMD,'GLOBAL STIFFNESS DUMP=7')  SOL02940
            ENDIF                                                       SOL02950
         ENDIF                                                          SOL02960
C                                                                       SOL02970
C==============================                                        SOL02980
C= FORM GEOMETRIC STIFFNESS ===                                        SOL02990
C==============================                                        SOL03000
C  FORMULATE THE GEOMETRIC STIFFNESS IF THE FLAG NEWK IS TRUE, THIS     SOL03010
C       FLAG INITIALIZED ABOVE.                                         SOL03020
C       IF KGDATA(1)=1, THE GEOMETRIC STIFFNESS IS FORMED ONCE, AND     SOL03030
C                       MULTIPLIED BY A SCALER TO ACCOUNT FOR CHANGES   SOL03040
C                       IN THE TOTAL VERTICAL LOAD. THUS THE GEOMETRIC  SOL03050
C                       STIFFNESS ONLY NEEDS TO BE FORMED ONCE.         SOL03060
C       IF KGDATA(1)=2, THE GEOMETRIC STIFFNESS IS BASED ON THE AXIAL   SOL03070
C                       FORCE IN THE ELEMENTS, AND IS RECALCULATED EACH SOL03080
C                       TIME THE AXIAL FORCE CHANGES.                   SOL03090
C       IF KGDATA(3)=1, THE GEOMETRIC STIFFNESS IS FORMED WITH THE      SOL03100
C                       STIFFNESS, THUS A SEPERATE CALL TO FORM IS      SOL03110
C                       NOT NEEDED                                      SOL03120
C       IF KGDATA(3)=2, A SEPERATE GEOMETRIC STIFFNESS MATRIX IS FORMED SOL03130
         IF (NEWKG) THEN                                                SOL03140
C                                                                       SOL03150
C              FORMULATE SEPERATE GEOMETRIC STIFFNESS MATRIX            SOL03160
               IF (KGFORM.EQ.2) THEN                                    SOL03170
                  FZ    =0                                            SOL03180
                  I2=2                                                  SOL03190
                  CALL FORM(I2)                                         SOL03200
                  KGCND2=KGCOND                                         SOL03210
                  IF (BUG5) THEN                                        SOL03220
                    CALL LMATRX(KG,MDKG,NDOF,IMDKG,'KG MATRIX DUMP=7')  SOL03230
                   ENDIF                                                SOL03240
               ELSE                                                     SOL03250
                  KGCND2=.FALSE.                                        SOL03260
               ENDIF                                                    SOL03270
C                                                                       SOL03280
         ENDIF                                                          SOL03290
C                                                                       SOL03300
C============================                                          SOL03310
C= CONDENSE OUT FREE DISPL===                                          SOL03320
C============================                                          SOL03330
C... IF MCOND IS TRUE THE MASS MATRIX IS ALSO REDUCED BY GUYAN REDUCTIONSOL03340
C... IF KGCND2 IS TRUE THEN KG IS ALSO REDUCED                         SOL03350
         IF (NCOND.GT.0 .AND. NEWK ) THEN                               SOL03360
            CALL MSOLL(1,BUG5,L4,IMD,NDOF,1,L1,L2,                      SOL03370
     &                 MD      ,STIFF ,DLOAD2,WORK,                     SOL03380
     &                 MCND2,MASS  ,MDMS   ,IMDMS,                     SOL03390
     &                 KGCND2,KG    ,MDKG   ,IMDKG,ABORT)              SOL03400
            IF (ABORT) GO TO 5000                                       SOL03410
            MCND2 =.FALSE.                                              SOL03420
            KGCND2=.FALSE.                                              SOL03430
         ELSE IF (NCOND.GT.0) THEN                                      SOL03440
C           REDUCE THE DYNAMIC LOAD MATRIX IF OLD STIFFNESS IS USED...  SOL03450
            CALL MSOLL(2,BUG5,L4,IMD,NDOF,1,L1,L2,                      SOL03460
     &                 MD      ,STIFF ,DLOAD2,WORK,                     SOL03470
     &                 MCND2,MASS  ,MDMS   ,IMDMS,                     SOL03480
     &                 KGCND2,KG    ,MDKG   ,IMDKG,ABORT)              SOL03490
            IF (ABORT) GO TO 5000                                       SOL03500
         ENDIF                                                          SOL03510
C                                                                       SOL03520
C-------------------------------------------------------------          SOL03530
C= DETERMINE IF A NEW DYNAMIC STIFFNESS NEEDS TO BE CALC ===           SOL03540
C-------------------------------------------------------------          SOL03550
         IF (NEWK .OR. NEWKG .OR. (KGLOAD.EQ.1 .AND. ACCEL2.EQ.ACCOLD))THENSOL03560
            NEWKDY=.TRUE.                                               SOL03570
            ACCOLD=ACCEL2                                               SOL03580
         ELSE                                                           SOL03590
            NEWKDY=.FALSE.                                              SOL03600
         ENDIF                                                          SOL03610
C                                                                       SOL03620
C           WRITE (6,*) '*----- DDISP -----* BEFORE LINACC '            SOL03630
C           WRITE (6,*) (DDISP(I),I=1,14)                               SOL03640
C                                                                       SOL03650
C-------------------------------------------------                      SOL03660
C= SOLVE DYNAMIC EQUATION FOR INCRIMENTAL DISPL ==                     SOL03670
C-------------------------------------------------                      SOL03680
         I2=2                                                           SOL03690
         IF (INTEGO.EQ.1) THEN                                          SOL03700
          CALL LINACC(I2,BUG4,DT,TO,TF,T,1.00,NEWKDY,                   SOL03710
     &     L1,L2,L3,L4,L5,L6,IMDMS,IMD,IMDS,IMDKG,NDOF,NFREE,           SOL03720
     &     KGLOAD,KGFORM,                                               SOL03730
     &      ACC, VEL   , DISP  , LOAD  ,                                SOL03740
     &      DACC  , DVEL  , DDISP , DLOAD2,WORK,                        SOL03750
     &      MASS  ,MDMS   , DAMP  , STIFF ,MD       ,                  SOL03760
     &      KG    ,MDKG   , FZ    ,GRAV,                               SOL03770
     &      SBAR  ,MDS    , AN    , BN    , PBAR  ,                     SOL03780
     &     FDAMP,FMASS,C1,C2,C3,C4,C5,C6,C7,ALPHA,BETA,C10,C11,ABORT)   SOL03790
         ELSE IF (INTEGO.EQ.2) THEN                                     SOL03800
          CALL AVEACC(I2,BUG4,DT,TO,TF,T,THETA,NEWKDY,                  SOL03810
     &     L1,L2,L3,L4,L5,L6,IMDMS,IMD,IMDS,IMDKG,NDOF,NFREE,           SOL03820
     &     KGLOAD,KGFORM,                                               SOL03830
     &      ACC, VEL   , DISP  , LOAD  ,                                SOL03840
     &      DACC  , DVEL  , DDISP , DLOAD2,WORK,                        SOL03850
     &      MASS  ,MDMS   , DAMP  , STIFF ,MD       ,                  SOL03860
     &      KG    ,MDKG   , FZ    ,GRAV,                               SOL03870
     &      SBAR  ,MDS    , AN    , BN    , PBAR  ,                     SOL03880
     &     FDAMP,FMASS,C1,C2,C3,C4,C5,C6,C7,ALPHA,BETA,C10,C11,ABORT)   SOL03890
         ELSE IF (INTEGO.EQ.3) THEN                                     SOL03900
          CALL WILSON(I2,BUG4,DT,TO,TF,T,THETA,NEWKDY,                  SOL03910
     &     L1,L2,L3,L4,L5,L6,IMDMS,IMD,IMDS,IMDKG,NDOF,NFREE,           SOL03920
     &     KGLOAD,KGFORM,                                               SOL03930
     &      ACC, VEL   , DISP  , LOAD  ,                                SOL03940
     &      DACC  , DVEL  , DDISP , DLOAD2,WORK,                        SOL03950
     &      MASS  ,MDMS   , DAMP  , STIFF ,MD       ,                  SOL03960
     &      KG    ,MDKG   , FZ    ,GRAV,                               SOL03970
     &      SBAR  ,MDS    , AN    , BN    , PBAR  ,                     SOL03980
     &     FDAMP,FMASS,C1,C2,C3,C4,C5,C6,C7,ALPHA,BETA,C10,C11,ABORT)   SOL03990
         ENDIF                                                          SOL04000
         IF (ABORT) GO TO 5000                                          SOL04010
C                                                                       SOL04020
C-------------------------------------------------                      SOL04030
C= BACK SUBSTIUTION TO SOLVE FOR NON-DYNAMIC DOF ===                   SOL04040
C-------------------------------------------------                      SOL04050
C  DISP_C = T * DISP_F                                                 SOL04060
C  DISP_C = INV(K_CC) * (-K_CF)*DISP(F)                                SOL04070
C  SUBRTN REA_CTN (IOPT=1);                                            SOL04080
C     PREFORMS THE MULTIPICATION DISP_C=(-K_CF)*DISP_F                 SOL04090
C  SUBRTN MSOLL  (IOPT=3);                                             SOL04100
C     PREFORMS THE BACK SUBSTITUTION TO EQUILIVENT TO                  SOL04110
C     DISP_C=INV(K_CC)*DISP_C                                          SOL04120
C                                                                       SOL04130
      ONE=1.00                                                          SOL04140
      IF (NCOND.GT.0) THEN                                              SOL04150
            I3=3                                                        SOL04160
            I1=1                                                        SOL04170
C------------------ DISPLACEMENTS ---------------------------------     SOL04180
            IF (BUG5) WRITE(6,*)'*---- CALC DISPL'                      SOL04190
C           WRITE (6,*) '*---- DDISP ----*'                             SOL04200
C           WRITE (6,*) (DDISP(I),I=1,14)                               SOL04210
            CALL REACTN(I1,BUG5,NNODE,L1,L2,L3,L4,DDISP,DDISP,          SOL04220
     &          MD      ,STIFF ,MD(NDOF),NDOF,1,ONE,                   SOL04230
     &         MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,             SOL04240
     &           NCOS,COSINE,JTCOS,JTFLG,DSUMRC)                       SOL04250
            CALL MSOLL(I3,BUG5,L4,IMD,NDOF,I1,L1,L2,MD ,STIFF,          SOL04260
     &              DDISP ,WORK,                                        SOL04270
     &              .FALSE.,MASS  ,MDMS   ,1,                           SOL04280
     &              .FALSE.,KG    ,MDKG   ,1,ABORT)                     SOL04290
            IF (ABORT) GO TO 5000                                       SOL04300
C------------------ VELOCITIES ------------------------------------     SOL04310
            IF (BUG5) WRITE(6,*)'*---- CALC VELOC'                      SOL04320
              CALL REACTN(I1,BUG5,NNODE,L1,L2,L3,L4,DVEL,DVEL,          SOL04330
     &          MD      ,STIFF ,MD(NDOF),NDOF,1,ONE,                   SOL04340
     &         MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,             SOL04350
     &           NCOS,COSINE,JTCOS,JTFLG,DSUMRC)                       SOL04360
            CALL MSOLL(I3,BUG5,L4,IMD,NDOF,I1,L1,L2,MD ,STIFF,          SOL04370
     &              DVEL  ,WORK,                                        SOL04380
     &              .FALSE.,MASS  ,MDMS   ,1,                           SOL04390
     &              .FALSE.,KG    ,MDKG   ,1,ABORT)                     SOL04400
            IF (ABORT) GO TO 5000                                       SOL04410
C------------------ ACCELERATION ----------------------------------     SOL04420
            IF (BUG5) WRITE(6,*)'*---- CALC ACCEL'                      SOL04430
            CALL REACTN(I1,BUG5,NNODE,L1,L2,L3,L4,DACC,DACC,            SOL04440
     &          MD      ,STIFF ,MD(NDOF),NDOF,1,ONE,                   SOL04450
     &         MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,             SOL04460
     &           NCOS,COSINE,JTCOS,JTFLG,DSUMRC)                       SOL04470
            CALL MSOLL(I3,BUG5,L4,IMD,NDOF,I1,L1,L2,MD ,STIFF,          SOL04480
     &              DACC  ,WORK,                                        SOL04490
     &              .FALSE.,MASS  ,MDMS   ,1,                           SOL04500
     &              .FALSE.,KG    ,MDKG   ,1,ABORT)                     SOL04510
            IF (ABORT) GO TO 5000                                       SOL04520
      ENDIF                                                             SOL04530
C                                                                       SOL04540
C---------------------------------------                                SOL04550
C= CALCULATE INCRIMENTAL MEMBER FORCES ===                             SOL04560
C---------------------------------------                                SOL04570
C ELELIB CALLS THE INDIVIDUAL ELEMENT LIBRARIES, EACH OF WHICH DETERMINSSOL04580
C  1) DID THE STIFFNESS CHANGE?  IF SO; NEWK=.TRUE.                     SOL04590
C  2) SHOULD THE INCRIMENTAL DISPLACEMENT BE RECALCULATED?              SOL04600
C     IF SO; KSAME=KSAME+___ (A VALUE DEPENDING ON ELEMENT TYPE)        SOL04610
C                                                                       SOL04620
      IOPT=4                                                            SOL04630
      LSTTYP=0                                                          SOL04640
      PRINT=.FALSE.                                                     SOL04650
      HEAD =.FALSE.                                                     SOL04660
      NEWK=.FALSE.                                                      SOL04670
      NELMT=NELMT                                                       SOL04680
      DO 150 IELNO=1,NELMT                                              SOL04690
  150   CALL ELELIB                                                     SOL04700
     &  (IOPT,LSTTYP ,PRINT ,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,    SOL04710
     &   IELNO ,ISLDOF,KGDOF,PGEOM,AINPUT,AXIAL,I2DDSP,I2DLOA,MSTOR)    SOL04720
C                                                                       SOL04730
C-----------------------------------------------                        SOL04740
C= ZERO STIFFNESS CHANGE FLAGS IF ELASTIC ===                          SOL04750
C-----------------------------------------------                        SOL04760
      IF (ELSTIC) THEN                                                  SOL04770
         NEWK=.FALSE.                                                   SOL04780
         KSAME=0                                                        SOL04790
      ENDIF                                                             SOL04800
C                                                                       SOL04810
C-------------------------------------------------                      SOL04820
C= DETERMINE IF KG IS TO BE UPDATED BASED ON  ===                      SOL04830
C=       CHANGING ELEMENT AXIAL FORCE         ===                      SOL04840
C-------------------------------------------------                      SOL04850
            IF (KGLOAD.EQ.2) THEN                                       SOL04860
               NEWKG=.TRUE.                                             SOL04870
               IF (NCOND.GT.0) NEWK=.TRUE.                              SOL04880
            ELSE                                                        SOL04890
               NEWKG=.FALSE.                                            SOL04900
            ENDIF                                                       SOL04910
C                                                                       SOL04920
C-----------------------------                                         SOL04930
C= REPEAT STEP IF NESSART ===                                          SOL04940
C-----------------------------                                         SOL04950
C  IF AN ELEMENT STIFFNESS CHANGE DICTATED THAT THE STEP BE REANALIZED, SOL04960
C     REFORMULATE THE STIFFNESS AND REANALIZE                           SOL04970
C     THE VARIABLE REPEAT IS SET UP TO SUPRESS THE REANALIZING OF A STEPSOL04980
C     CURENTLY, EACH STEP MAY ONLY BE REANALIZED ONCE, THIS PREVENTS    SOL04990
C     AN ENDLESS LOOP.                                                  SOL05000
C                                                                       SOL05010
      IF (KSAME.NE.0 .AND. REPEAT) THEN                                 SOL05020
          REPEAT=.FALSE.                                                SOL05030
          GO TO 100                                                     SOL05040
      ENDIF                                                             SOL05050
C                                                                       SOL05060
C-------------------------                                              SOL05070
C= SOLVE FOR REACTIONS ===                                             SOL05080
C-------------------------                                              SOL05090
C REACTIONS ARE DUE TO INTERNAL MEMBER FORCES                           SOL05100
C                                                                       SOL05110
      IF (BUG5) WRITE(6,*)'*---- CALC REACT=K*X'                        SOL05120
      CALL REACTN(2,BUG5,NNODE,L1,L4,L5,L6,DLOAD ,DDISP ,               SOL05130
     &  MD      ,STIFF ,MD(NDOF),NDOF,1,1.0,                           SOL05140
     &  MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,                     SOL05150
     &           NCOS,COSINE,JTCOS,JTFLG,DSUMRC)                        SOL05160
C                                                                       SOL05170
C----- GET SUMMATION OF REACTIONS                                       SOL05180
      CALL REACTN(3,BUG5,NNODE,L1,L4,L5,L6,DLOAD ,DDISP,                SOL05190
     &          MDKG    ,KG    ,MDKG(NDOF),NDOF,1,1.0,                 SOL05200
     &         MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,              SOL05210
     &           NCOS,COSINE,JTCOS,JTFLG,DSUMRC)                        SOL05220
C                                                                       SOL05230
C----- GET SUMMATION OF REACTIONS DUE TO UNBALANCED LOAD                SOL05240
      IF (.NOT.ELSTIC)                                                  SOL05250
     & CALL REACTN(3,BUG5,NNODE,L1,L4,L5,L6,DLOAD ,DDISP,               SOL05260
     &          MDKG    ,KG    ,MDKG(NDOF),NDOF,1,1.0,                 SOL05270
     &         MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,              SOL05280
     &           NCOS,COSINE,JTCOS,JTFLG,SUMUB)                         SOL05290
C                                                                       SOL05300
C-------------------------------------------------------------          SOL05310
C= SOLVE FOR INCRIMENTAL GEOMETRIC STIFFNESS FORCE, DFG ===            SOL05320
C-------------------------------------------------------------          SOL05330
C                                                                       SOL05340
C------- INCRIMENTAL GEOMETRIC STIFFNESS FORCE                          SOL05350
      IF (KGFORM.EQ.2) THEN                                             SOL05360
         IF (BUG5) WRITE(6,*) '*------ INCRIMENTAL FORCE DUE TO KG '    SOL05370
         CALL TMPLY(BUG5,.TRUE.,IMD,L4,L3,L4,L3,L4,                     SOL05380
     &          MDKG,KG,DFG,DDISP)                                      SOL05390
C        SCALE FORCE TO INLCUDE GROUND ACCELERATION ...                 SOL05400
         DO 159 I=L3,L4                                                 SOL05410
```

```
  159       DPG(I)= (1+F%)*DPG(I)
      ENDIF
C
C*****************************************************
C= SOLVE FOR INCRIMENTAL DAMPING FORCE,  DFDAMP ===
C*****************************************************
C
      IF (BETA.NE.0) THEN
C------- INCRIMENTAL DAMPING FORCE DUE TO STIFFNESS ...
          IF (BUG5) WRITE(6,*)
     &     ' *----- DAMPING FORCE/REACT DUE TO STIFFNESS'
          CALL TMPLY(BUG5,.TRUE.,IMD,L6,L3,L4,L3,L4,
     &       MD,STIFF,DFDAMP,DVEL)
          CALL TMPLY(BUG5,.TRUE.,IMD,L6,L5,L6,L1,L4,
     &       MD,STIFF,DFDAMP,DVEL)

C------- INCRIMENTAL DAMPING FORCE DUE TO KG  ...
          IF (KGFORM.EQ.2) THEN
            IF (BUG5) WRITE(6,*)
     &     ' *----- DAMPING FORCE/REACT DUE TO KG '
          CALL TMPLY(BUG5,.TRUE.,IMD,L6,L3,L4,L3,L4,
     &       MDKG,KG,WORK,DVEL)
          CALL TMPLY(BUG5..TRUE.,IMD,L6,L5,L6,L1,L4,
     &       MDKG,KG.WORK.DVEL)
            DO 158 I=L3,L6
  158         DFDAMP(I)=DFDAMP(I)-WORK(I)
          ENDIF

C------------ CALC SUM OF STIFFNESS PROPORTIONAL DAMPING REACTIONS
      CALL REACTN(3,BUG5,NNODE,L1,L4,L5,L6,DFDAMP,DVEL ,
     &       MDMS   ,MASS  ,MDMS(NDOF),NDOF,1,1.00,
     &       MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,
     &         NCOS,COSINE,JTCOS,JTFLG,DSUMFD)

C------- MULTIPLY BY BETA
          DO 160 I=1,6
            DSUMFD(I)=DSUMFD(I)*BETA
  160       IF (BUG5) WRITE (6,*) 'I,DSUMFD(I),BETA',I,DSUMFD(I),BETA
          DO 161 I=L3,L4
  161       DFDAMP(I)*DFDAMP(I)*BETA
      ENDIF
C
      IF (ALPHA.NE.0) THEN
C------- INCRIMENTAL DAMPING FORCE DUE TO MASS ...
          IF (BUG5) WRITE(6,*) '*----- DAMPING FORCE DUE TO MASS '
          DO 162 I=L1,L6
  162       WORK(I)=0.00
          CALL TMPLY(BUG5,.FALSE.,IMD,L4,L3,L4,L3,L4,
     &       MDMS,MASS,WORK,DVEL)

C------------ CALC SUM OF FD
          CALL REACTN(3,BUG5,NNODE,L1,L2,L3,L4,WORK,DVEL ,
     &       MDMS   ,MASS  ,MDMS(NDOF),NDOF,1,1.00,
     &       MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,
     &         NCOS,COSINE,JTCOS,JTFLG,TSUM)

C------- MULTIPLY BY ALPHA ADD TO PREVIOUS SUM ...
          IF (BETA.NE.0) THEN
            DO 165 I=L3,L4
  165       DFDAMP(I)=DFDAMP(I)+ALPHA*WORK(I)
            DO 167 I=1,6
  167       DSUMFD(I)=DSUMFD(I)-TSUM(I)*ALPHA
          ELSE
            DO 168 I=L3,L4
  168       DFDAMP(I)=           ALPHA*WORK(I)
            DO 166 I=1,6
  166       DSUMFD(I)=-TSUM(I)*ALPHA
          ENDIF
          IF (BUG5) WRITE (6,*) 'DSUMFD=',DSUMFD
      ENDIF
      IF (BUG5) THEN
        DO 169 I=L3,L6
  169   WRITE (6,*) 'I',I,'DFDAMP',DFDAMP(I),'FDAMP',TFDAMP(I)
      ENDIF
C
C*****************************************
C= CALC AND ENERGIES AND DUCTILITIES  ===
C*****************************************
      WRITE=  MOD(ISTEP,IPRNT).EQ.0
      HEAD = WRITE
C
      IF (HEAD) WRITE (6,250) TITLE(1),TITLE(2),' ',ISTEP,STEPID
      IF (SLMASS) THEN
C------------ ENERGY FORMULATION WITH SPECIAL LOADING MASS ---
C             BASED ON APPLIED LOADING
       CALL ENERGY
     & (L1,L2,L3,L4,L5,L6,NDOF,MAXNOD,NCOS,BUG11,.FALSE.,
     &  IMD,IMDMS,IMDKG,MD,MDMS,MDKG,.TRUE.,WRITE,ABORT,
     &  STIFF,KG,MASS,
     &  EIE,ESE,PSE,EKE,EDD,
     &  ALPHA,BETA,KGFORM,NELMT,NNODE,DT,SUMUB,
     &   SUMRC,DSUMRC,  SUMFD,DSUMFD,TFDAMP,DFDAMP,  FZ,
     &  ZERO,GV,GD,ZERO,GVT,GDT,   WORK,WORK(L6+1),PG,DPG,
     &  DACC,DVEL,DDISP,ACC,VEL,DISP, LOAD,DLOAD,
     &  ID      , IDOF     , CNST     , JTFLG    , COORD    ,
     &  COSINE   , JTCOS    )

      ELSE
C------------ ENERGY FORMULATION WITH/O SPECIAL LOADING MASS ---
C             BASED ON GROUND MOTION
       CALL ENERGY
     & (L1,L2,L3,L4,L5,L6,NDOF,MAXNOD,NCOS,BUG11,.FALSE.,
     &  IMD,IMDMS,IMDKG,MD,MDMS,MDKG,.TRUE.,WRITE,ABORT,
     &  STIFF,KG,MASS,
     &  EIE,ESE,PSE,EKE,EDD,
     &  ALPHA,BETA,KGFORM,NELMT,NNODE,DT,SUMUB,
     &   SUMRC,DSUMRC,  SUMFD,DSUMFD,TFDAMP,DFDAMP,  FZ,
     &  GA,GV,GD,GAT,GVT,GDT,   WORK,WORK(L6+1),PG,DPG,
     &  DACC,DVEL,DDISP,ACC,VEL,DISP,  ZERO,ZERO,
     &  ID      , IDOF     , CNST     , JTFLG    , COORD    ,
     &  COSINE   , JTCOS    )
      ENDIF
      EIEM =AMAX1(SNGL(EIE),EIEM )
      ESEM =AMAX1(SNGL(ESE),ESEM )
      PSEM =AMAX1(SNGL(PSE),PSEM )
      EKEM =AMAX1(SNGL(EKE),ELEM )
      EDDM =AMAX1(SNGL(EDD),EDDM )
      IF (ABORT) GO TO 5000
C
C*****************************************************
C= GET TOTAL GLOBAL LOAD, DISPL, VELOC AND ACCEL ===
C*****************************************************
      IF (BUG4) WRITE (6,320)
      DO 310 I=1,NDOF
        ACC(I)   = DACC(I)   + ACC(I)
        VEL(I)   = DVEL(I)   + VEL(I)
        DISP(I)  = DDISP(I)  + DISP(I)
C
        LOAD(I)  = DLOAD(I)  + LOAD(I)
        TFDAMP(I)= DFDAMP(I) + TFDAMP(I)
        FG(I)    = DFG(I)    + FG(I)
        IF (ABS( ACC(I)).GT.ABS(ACCMAX(I))) ACCMAX(I)= ACC(I)
        IF (ABS(VEL(I)).GT.ABS(VELMAX(I))) VELMAX(I)= VEL(I)
        IF (ABS(DISP(I)).GT.ABS(DSPMAX(I))) DSPMAX(I)=DISP(I)
        IF (ABS(LOAD(I)).GT.ABS(RCTMAX(I))) RCTMAX(I)=LOAD(I)
        IF (BUG4)
     &  WRITE (6,330) I,DDISP(I),DVEL(I),DACC(I),DLOAD(I),
     &                  DISP(I), VEL(I), ACC(I), LOAD(I)
        DACC(I)=0.00
        DVEL(I)=0.00
        DDISP(I)=0.00
        DFDAMP(I)=0.00
  310   DLOAD(I) =0.00
C
        DO 309 I=1,6
          SUMRC(I) = DSUMRC(I) + SUMRC(I)
  309     SUMFD(I) = DSUMFD(I) + SUMFD(I)
        SUMFOR=SQRT(SUMRC(1)**2+SUMRC(2)**2+SUMRC(3)**2)
        SUMMOM=SQRT(SUMRC(4)**2+SUMRC(5)**2+SUMRC(6)**2)
        SUMMFO=MAX(SUMFOR,SUMMFO)
        SUMMOM=MAX(SUMMOM,SUMMMO)
C
  320   FORMAT ('   DOF',6X,'DD',12X,'DV',12X,'DA',12X,'DP',
     &                  12X,' D',12X,' V',12X,' A',12X,' P')
  330   FORMAT (1X,I6,1P,8G14.6)
C
C*****************************************************************
C= PRINT TOTAL GLOBAL DISPLACEMENTS, VELOCITIES, ACCELERATIONS ===
C*****************************************************************
      IF (WRITE) THEN
        CALL JOIDSP(1,MAXNOD,NDOF,NNODE,1,L3,L4,ID,
     &       IDOF,CNST  ,DISP  ,
     &       TITLE,'DISPLACEMENTS',STEPID,.TRUE.,
     &       NCOS,COSINE,JTCOS,JTFLG)
        CALL JOIDSP(1,MAXNOD,NDOF,NNODE,1,L3,L4,ID,
     &       IDOF,CNST  ,VEL   ,
     &       TITLE,'VELOCITIES',STEPID,.TRUE.,
     &       NCOS,COSINE,JTCOS,JTFLG)
        CALL JOIDSP(1,MAXNOD,NDOF,NNODE,1,L3,L4,ID,
     &       IDOF,CNST  ,ACC   ,
     &       TITLE,'ACCELERATIONS',STEPID,.TRUE.,
     &       NCOS,COSINE,JTCOS,JTFLG)
      ENDIF
C*********************************
C= PRINT TOTAL GLOBAL REACTIONS ===
C*********************************
      IOPT=3
      CALL REACTN(IOPT,WRITE,NNODE,L1,L4,L5,L6,LOAD,ACC ,
     &     MDMS   ,MASS  ,MDMS(NDOF),NDOF,1,ONE,
     &     MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.TRUE.,
     &     NCOS,COSINE,JTCOS,JTFLG,SUMRC)
      IF (WRITE) WRITE (6,313) SUMFOR,SUMMOM
  313 FORMAT (/4X,'RESULTANT OF REACTIONS,   FORCE=',1P,G12.4,
     &                                    '   MOMENT=',G12.4)
      DO 311 I=1,6
        J=NDOF+I
        IF (ABS(SUMRC(I)).GT.ABS(RCTMAX(J))) RCTMAX(J)=SUMRC(I)
  311 CONTINUE
C*********************************************
C= CALCULATE AND PRINT TOTAL MEMBER FORCES ===
C*********************************************
      IOPT=5
      LSTTYP=0
      IF (WRITE) THEN
          PRINT=.TRUE.
          HEAD =.TRUE.
      ELSE
          PRINT=.FALSE.
          HEAD =.FALSE.
      ENDIF
      DO 300 IELNO=1,NELMT
  300   CALL ELELIB
     &  (IOPT  ,LSTTYP ,PRINT ,IREL,HEAD,   NAME ,EESE,EPSE,DAMAGE,
     &   DUCFAG,
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,I&DDSP,I&DLOA,MSTOR)
C
C*************************************************
C= CALCULATE AND PRINT SYSTEM DUCTILITY BASE
C*************************************************
C.... SYSTEM DUCTILITY IS BASED ON THAT THE FIRST ELEMENT REACH TO
C     ITS CRITICAL LOAD
      IF( .NOT. ELSTIC) THEN
      IOPT=11
      LSTTYP=0
      IF( .NOT. DUCLOG) THEN
        PRINT=.FALSE.
        HEAD =.FALSE.
        DO 3120 IELNO=1,NELMT
          DUCFAG=0
          CALL ELELIB
     &    (IOPT,LSTTYP ,PRINT ,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,
     &     IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,I&DDSP,I&DLOA,MSTOR)
          IF( DUCFAG .GE. 1 ) THEN
            WRITE(6,3140) IELNO
            CALL JOIDSP(1,MAXNOD,NDOF,NNODE,1,L3,L4,ID,
     &           IDOF,CNST  ,DISP  ,
     &           TITLE,'DISPLACEMENTS',STEPID,.FALSE.,
     &           NCOS,COSINE,JTCOS,JTFLG)
            DUCLOG= .TRUE.
            GO TO 3130
          ENDIF
 3120   CONTINUE
      ENDIF
      ENDIF
 3140 FORMAT ('  SYSTEM DISPLACEMENT ( FIRST ELEMENT REACH TO CRITICAL',
     &        ' LOAD'/
     &        '  ------------------------------------------------------',
     &        '------'/
     &        ' ** ELEMENT ',I5,' FIRST REACH TO CRITICAL LOAD **')
C
C
C*************************************************************
C= CALCULATE AND PRINT DUCTILITY AND EXCURSION RATIOS ===
C*************************************************************
 3130 IOPT=11
      LSTTYP=0
      IF (WRITE) THEN
        PRINT=.TRUE.
        HEAD =.TRUE.
        DO 312 IELNO=1,NELMT
  312     CALL ELELIB
     &    (IOPT  ,LSTTYP ,PRINT ,IREL,HEAD,   NAME ,EESE,EPSE,DAMAGE,
     &     DUCFAG,
     &     IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,I&DDSP,I&DLOA,MSTOR)
      ENDIF
C
C*************************************
C= WRITE DATA TO OUTPUT FILES  ===
C*************************************
      IF (IWRITE.GT.0) THEN
        IF (MOD(ISTEP,IWRITE).EQ.0) THEN
          IOPT=2
          CALL DMPDAT(IOPT,IWRITE,TO,DT)
        ENDIF
      ENDIF
C
C*****************************************************************
C= CHECK ELAPSED CPU TIME ON THE UMRBX OR CORNELLF SYSTEM ===
C*****************************************************************
      DO 390 I=1,NDOF
        IF( ABS(DISP(I)) .GT. 300 ) THEN
          WRITE(6,*)' DISP(',I,') GREATER THAN 300. PROGRAM STOP'
          GO TO 5000
        ENDIF
  390 CONTINUE
C
      IF( PLACE .EQ. 'UMRVMB' ) THEN
        CALL TIMEIT(CPU)
        TCPU=CPU+TCPU
        ITLCPU=REQCPU
        CPUREM=REQCPU-TCPU
        CPUELA=ITLCPU-CPUREM
```

```
      IF (ITLCPU.GT.0) THEN                                             SOL08260
        IF (CPUREM.LT.0.05*ITLCPU .OR. CPUREM.LT.5) THEN                SOL08270
         WRITE (6,4891) ITLCPU,CPUELA,CPUREM,ISTEP,T                    SOL08280
         ABORT=.TRUE.                                                   SOL08290
         GO TO 5000                                                     SOL08300
 4891    FORMAT (//                                                     SOL08310
     &    ' ==========================================================='/ SOL08320
     &    ' ==                                                      ==='/ SOL08330
     &    ' ==            TIME LIMIT IS EXCEEDED                    ==='/ SOL08340
     &    ' ==              SOLUTION IS ABORTED                     ==='/ SOL08350
     &    ' ==                                                      ==='/ SOL08360
     &    ' ==  CPU TIME AT BEGINING OR EXECUTION...', I7,' (SEC) ==='/ SOL08370
     &    ' ==  ELAPSED CPU TIME.....................', I7,' (SEC) ==='/ SOL08380
     &    ' ==  CPU TIME REMAINING...................', I7,' (SEC) ==='/ SOL08390
     &    ' ==  NUMBER OF STEPS COMPLETED............', I7,'     ==='/ SOL08400
     &    ' ==  SOLUTION STOP TIME.....',1P,G14.7  ,'     ==='/ SOL08410
     &    ' ==                                                      ==='/ SOL08420
     &    ' ==========================================================') SOL08430
        ENDIF                                                           SOL08440
      ENDIF                                                             SOL08450
C                                                                       SOL08460
      ELSE IF( PLACE .EQ. 'CORNEL' ) THEN                               SOL08470
        CALL TIMEIT(CPU,TCPU)                                           SOL08480
        TCPU=CPU+TCPU                                                   SOL08490
        ITLCPU=REQCPU                                                   SOL08500
        CPUREM=REQCPU-TCPU                                              SOL08510
        CPUELA=ITLCPU-CPUREM                                            SOL08520
        IF (ITLCPU.GT.0) THEN                                           SOL08530
          IF (CPUREM.LT.0.05*ITLCPU .OR. CPUREM.LT.5) THEN              SOL08540
          WRITE (6,4891) ITLCPU,CPUELA,CPUREM,ISTEP,T                   SOL08550
          ABORT=.TRUE.                                                  SOL08560
          GO TO 5000                                                    SOL08570
         ENDIF                                                          SOL08580
        ENDIF                                                           SOL08590
      ENDIF                                                             SOL08600
C                                                                       SOL08610
  500 CONTINUE                                                          SOL08620
C                                                                       SOL08630
C-------------------------                                              SOL08640
C- CLOSE OUTPUT FILES  ---                                              SOL08650
C-------------------------                                              SOL08660
 5000 IF (IWRITE.GT.0) THEN                                             SOL08670
        IOPT=3                                                          SOL08680
        CALL DMPDAT(IOPT,IWRITE,TO,DT)                                  SOL08690
      ENDIF                                                             SOL08700
C                                                                       SOL08710
C---------------------------------------                                SOL08720
C- PRINT MAXIMUM GLOBAL DISPLACEMENTS ---                               SOL08730
C---------------------------------------                                SOL08740
  251 FORMAT ('1 STRUCTURE....: ',A,//                                  SOL08750
     &        '  SOLUTION.....: ',A,//                                  SOL08760
     &        '  MAXIMUN VALUES FOR ALL STEPS '/                        SOL08770
     &        ' ------------------------------')                        SOL08780
      STEPID=' '                                                        SOL08790
      ISTEP = 0                                                         SOL08800
      WRITE (6,251) TITLE(1),TITLE(2)                                   SOL08810
          CALL JOIDSP(2,MAXNOD,NDOF,NNODE,1,L3,L4,ID,                   SOL08820
     &              IDOF,CNST  ,DSPMAX,                                 SOL08830
     &              TITLE,'MAXIMUM DISPLACEMENTS',STEPID,.FALSE.,       SOL08840
     &              NCOS,COSINE,JTCOS,JTFLG)                            SOL08850
      WRITE (6,251) TITLE(1),TITLE(2)                                   SOL08860
          CALL JOIDSP(2,MAXNOD,NDOF,NNODE,1,L3,L4,ID,                   SOL08870
     &              IDOF,CNST  ,VELMAX  ,                               SOL08880
     &              TITLE,'MAXIMUM VELOCITIES',STEPID,.FALSE.,          SOL08890
     &              NCOS,COSINE,JTCOS,JTFLG)                            SOL08900
      WRITE (6,251) TITLE(1),TITLE(2)                                   SOL08910
          CALL JOIDSP(2,MAXNOD,NDOF,NNODE,1,L3,L4,ID,                   SOL08920
     &              IDOF,CNST  ,ACCMAX  ,                               SOL08930
     &              TITLE,'MAXIMUM ACCELERATIONS',STEPID,.FALSE.,       SOL08940
     &              NCOS,COSINE,JTCOS,JTFLG)                            SOL08950
C                                                                       SOL08960
C---------------------------------------                                SOL08970
C- PRINT MAXIMUM GLOBAL REACTIONS ---                                   SOL08980
C---------------------------------------                                SOL08990
      WRITE (6,251) TITLE(1),TITLE(2)                                   SOL09000
      DO 5100 I=1,6                                                     SOL09010
        SUMRC(I)=RCTMAX(I+NDOF)                                         SOL09020
 5100   CONTINUE                                                        SOL09030
        CALL REACTN(4,.TRUE.,NNODE,L1,L4,L5,L6,RCTMAX,DSPMAX,           SOL09040
     &       MDMS   ,MASS  ,MDMS(NDOF),NDOF,1,ONE,                     SOL09050
     &       MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,                SOL09060
     &         NCOS,COSINE,JTCOS,JTFLG,SUMRC)                           SOL09070
      WRITE (6,5111) SUMMFO,SUMMMO                                      SOL09080
 5111 FORMAT (/4X,'MAXIMUM RESULTANT OF REACTIONS,   FORCE=',1P,G12.4,   SOL09090
     &                                              MOMENT=',G12.4)     SOL09100
C                                                                       SOL09110
C---------------------------------                                      SOL09120
C- PRINT MAXIMUM MEMBER FORCES ---                                      SOL09130
C---------------------------------                                      SOL09140
      LSTTYP=0                                                          SOL09150
      PRINT=.TRUE.                                                      SOL09160
      HEAD =.FALSE.                                                     SOL09170
      WRITE (6,251) TITLE(1),TITLE(2)                                   SOL09180
      DO 5300 IELNO=1,NELMT                                             SOL09190
 5300   CALL ELELIB                                                     SOL09200
     &  (12    ,LSTTYP ,PRINT,IREL ,HEAD  ,NAME ,EESE,EPSE,DAMAGE,     SOL09210
     &   DUCFAG,                                                        SOL09220
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)    SOL09230
C                                                                       SOL09240
C-------------------------------------------                            SOL09250
C- MEMBER DUCTILITY AND EXCURSION RATIOS ---                            SOL09260
C-------------------------------------------                            SOL09270
      IF (.NOT. ELSTIC) THEN                                            SOL09280
          LSTTYP=0                                                      SOL09290
          PRINT=.TRUE.                                                  SOL09300
          HEAD =.FALSE.                                                 SOL09310
          WRITE (6,251) TITLE(1),TITLE(2)                               SOL09320
          DO 5310 IELNO=1,NELMT                                         SOL09330
 5310       CALL ELELIB                                                SOL09340
     &       (11    ,LSTTYP ,PRINT,IREL ,HEAD  ,NAME ,EESE,EPSE,DAMAGE, SOL09350
     &        DUCFAG,                                                   SOL09360
     &        IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)SOL09370
        ENDIF                                                           SOL09380
C                                                                       SOL09390
C-----------------------------                                          SOL09400
C- PRINT DAMAGE INDICIES  ---                                           SOL09410
C-----------------------------                                          SOL09420
      IF (.NOT. ELSTIC) THEN                                            SOL09430
          LSTTYP=0                                                      SOL09440
          HEAD =.FALSE.                                                 SOL09450
          PRINT=.TRUE.                                                  SOL09460
          WRITE (6,251) TITLE(1),TITLE(2)                               SOL09470
          DAMAGE=0                                                      SOL09480
          DO 5320 IELNO=1,NELMT                                         SOL09490
            CALL ELELIB                                                 SOL09500
     &      (14    ,LSTTYP ,PRINT,IREL ,HEAD  ,NAME ,EESE,EPSE,EDAM,   SOL09510
     &       DUCFAG,                                                    SOL09520
     &       IELNO,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR) SOL09530
 5320       DAMAGE=DAMAGE+EDAM                                         SOL09540
          IF( (ESE+PSE) .EQ. 0 ) RETURN                                 SOL09550
          DAMAGE=DAMAGE/(ESE+PSE)                                       SOL09560
          WRITE (6,5330) DAMAGE                                         SOL09570
 5330     FORMAT (/10X,'STRUCTURE DAMAGE INDEX=',1P,G15.5)              SOL09580
      ENDIF                                                             SOL09590
C                                                                       SOL09600
C-----------------------------                                          SOL09610
C- PRINT PEAK ENERGY INFO ---                                           SOL09620
C-----------------------------                                          SOL09630
      WRITE (6,251) TITLE(1),TITLE(2)                                   SOL09640
      WRITE (6,5340) EIEM,ESEM,PSEM,EKEM,EDDM                           SOL09650
 5340 FORMAT (/'   PEAK ENERGY VALUES ' / 1P,                          SOL09660
     & '   MAX INPUT ENERGY.............',G12.5/                       SOL09670
     & '   MAX ELASTIC STRAIN ENERGY.....',G12.5/                     SOL09680
     & '   MAX PLASTIC STRAIN ENERGY.....',G12.5/                     SOL09690
     & '   MAX KINETIC ENERGY............',G12.5/                     SOL09700
     & '   MAX DAMPING ENERGY............',G12.5/)                    SOL09710
C                                                                       SOL09720
C                                                                       SOL09730
      RETURN                                                            SOL09740
C                                                                       SOL09750
  250 FORMAT ('1 STRUCTURE....: ',A,/                                   SOL09760
     &        '  SOLUTION.....: ',A,//                                 SOL09770
     &        2X,A,1X,',  LOADING #',I5,5X,A)                         SOL09780
C                                                                       SOL09790
      END                                                               SOL09800
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP                                  SOL00010
C=======================================================================SOL00020
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    SOL00030
C           END DEBUG                                                   SOL00040
C=======================================================================SOL00050
      SUBROUTINE SOL03                                                  SOL00060
      LOGICAL TEST,QUIT,BTEST                                           SOL00070
      INCLUDE (%COMM)                                                   SOL00080
C                                                                       SOL00090
C       ---- SET UP CONSTANTS ------------------                        SOL00100
      L1=1                                                              SOL00110
      L2=NCOND                                                          SOL00120
      L3=L2+1                                                           SOL00130
      L4=NCOND +NFREE                                                   SOL00140
      L5=L4+1                                                           SOL00150
      L6=NDOF                                                           SOL00160
      ELSTIC=.TRUE.                                                     SOL00170
C                                                                       SOL00180
C       ---- INITIALIZE STORAGE                                         SOL00190
C     %(IZEVAL) = LOCATION OF THE EIGENVALUES                           SOL00200
C     %(IZEVEC) = LOCATION OF THE EIGENVECTORS                          SOL00210
C     %(IZA   ) = LOCATION OF THE SYMMETRIC STIFFNESS                   SOL00220
C     %(IZB   ) = LOCATION OF THE SYMMETRIC MASS OR KG                  SOL00230
C                                                                       SOL00240
       IZEVAL = I%                                                      SOL00250
       IZ     = I%+NFREE                                                SOL00260
       IZEVEC = I%                                                      SOL00270
       I%     = I%+NFREE*NFREE                                          SOL00280
       IZA    = I%                                                      SOL00290
       I%     = I%+NFREE*NFREE                                          SOL00300
       I%B    = I%                                                      SOL00310
       I%     = I%+NFREE*NFREE                                          SOL00320
       IZTMP= I%                                                        SOL00330
       I%   = I%+NDOF                                                   SOL00340
       IW=I%                                                            SOL00350
C                                                                       SOL00360
      CALL CKSTOR(NDOF,IW)                                              SOL00370
C                                                                       SOL00380
      IMD  =N%(IZMD  +NDOF)                                             SOL00390
      IMDMS=N%(IZMDMS+NDOF)                                             SOL00400
      IMDKG=N%(IZMDKG+NDOF)                                             SOL00410
      IMDAB=NFREE*NFREE                                                 SOL00420
C                                                                       SOL00430
C---------- FORM MASS MATRIX -----                                      SOL00440
C                                                                       SOL00450
   DO 400 I=1,IMDMS                                                     SOL00460
 400     %(I+IZMASS-1)=0.                                               SOL00470
C                                                                       SOL00480
      CALL MFORM(2,INMASS,NMASS,%(IZMASS),N%(IZMDMS),MD,NDOF,IMDMS,     SOL00490
     &           TITLE,NNODE,MAXNOD,NCOS,BTEST(IBUG,5),IBUG,            SOL00500
     &           N%(IZID),%(IZCOS),N%(IZIDOF),%(IZCNST),N%(IZJFLG),     SOL00510
     &           N%(IZJCOS),%(IZMDAT),Z(IW),N%(IW+21),Z(IW+27),         SOL00520
     &            %(IW+ 63),%(IW+ 99))                                  SOL00530
C                                                                       SOL00540
C                                                                       SOL00550
C                                                                       SOL00560
      CALL SOL03A                                                       SOL00570
     & (L1,L2,L3,L4,L5,L6,IMD,IMDMS,IMDKG,       KGCOND,MCOND,ABORT,    SOL00580
     &  %(IZKGDT),N%(IZKGDT+1),N%(IZKGDT+2),N%(IZKGDT+3),               SOL00590
     &  NDOF     , NCOND    , NFREE    , TITLE    , STEPID   ,         SOL00600
     &  %(IZSTIF), %(IZMASS), Z(IZKG  ), %(IZA   ), %(IZB   ),         SOL00610
     & N%(IZMD  ),N%(IZMDMS),N%(IZMDKG), Z(IW    ), Z(IZTMP ),         SOL00620
     &  NNODE    , NCOS     , NELMT    , MAXNOD   ,IBUG      ,         SOL00630
     & N%(IZID  ),N%(IZIDOF), %(IZCNST),N%(IZJFLG), %(IZCORD),         SOL00640
     &  %(IZCOS ),N%(IZJCOS), %(IZEVAL), %(IZEVEC) )                    SOL00650
C                                                                       SOL00660
      RETURN                                                            SOL00670
      END                                                               SOL00680
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP                                  SOL00010
C=======================================================================SOL00020
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                    SOL00030
C           END DEBUG                                                   SOL00040
C=======================================================================SOL00050
      SUBROUTINE SOL03A                                                 SOL00060
     & (L1,L2,L3,L4,L5,L6,IMD,IMDMS,IMDKG,       KGCOND,MCOND,ABORT,    SOL00070
     &  KGDATA,KGLOAD,KGTYPE,KGFORM,                                    SOL00080
     &  NDOF     , NCOND    , NFREE    , TITLE    , STEPID   ,         SOL00090
     &  STIFF    , MASS     , KG       , A        , B        ,         SOL00100
     &  MD       , MDMS     , MDKG     , WORK     , TMP      ,         SOL00110
     &  NNODE    , NCOS     , NELMT    , MAXNOD   , IBUG     ,         SOL00120
     &  ID       , IDOF     , CNST     , JTFLG    , COORD    ,         SOL00130
     &  COSINE   , JTCOS    , EVAL     , EVEC     )                   SOL00140
C-- COMMON BLOCK FOR IMSL SUBROUTINE DGVCSP                            SOL00150
C     COMMON/WORKSP/RWKSP                                               SOL00160
C     REAL RWKSP(100000)                                                SOL00170
      LOGICAL TEST,QUIT,BTEST,MCOND,KGCOND,MCOND1,KCOND1,ABORT          SOL00180
      REAL MASS, KG, KGDATA                                             SOL00190
      DIMENSION STIFF(IMD),MASS(IMDMS),KG(IMDKG)                        SOL00200
      DIMENSION A(NFREE,NFREE),B(NFREE,NFREE)                           SOL00210
      DIMENSION MD(L6+1),MDMS(L6+1),MDKG(L6+1),WORK(L6),TMP(L6)         SOL00220
      DIMENSION EVAL(NFREE),EVEC(NFREE,NFREE),SUMRCT(6)                 SOL00230
      DIMENSION COORD(MAXNOD,3),ID(MAXNOD),                             SOL00240
     &          IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),CNST(MAXNOD,6),        SOL00250
     &          JTFLG(MAXNOD),JTCOS(MAXNOD)                             SOL00260
      DIMENSION G(6),D(6)                                               SOL00270
      DIMENSION WK2(80, 80)                                             SOL00280
C NOTE: WK2 DIMENSION SHOLUD BE GREATER THAN NFREE*NFREE                SOL00290
C                                                                       SOL00300
      CHARACTER*1   NAME                                                SOL00310
      CHARACTER*40  OPTION                                              SOL00320
      CHARACTER*(*) TITLE(2),STEPID                                     SOL00330
C                                                                       SOL00340
C-----------------                                                      SOL00350
C- INPUT DATA ---                                                       SOL00360
C-----------------                                                      SOL00370
      READ (5,*) OPTION,IPRT,IPRT2                                      SOL00380
C                                                                       SOL00390
      IF (TEST(OPTION,'FREQ',.FALSE.)) THEN                             SOL00400
         IOPT=1                                                         SOL00410
         OPTION='NATURAL FREQUENCIES'                                   SOL00420
      ELSE IF (TEST(OPTION,'BUCK',.FALSE.)) THEN                        SOL00430
         IOPT=2                                                         SOL00440
         OPTION='ELASTIC BUCKLING LOAD'                                 SOL00450
      ELSE                                                              SOL00460
         WRITE (6,*) 'SOL-03 INVALID OPTION: ',OPTION                   SOL00470
         RETURN                                                         SOL00480
      ENDIF                                                             SOL00490
C                                                                       SOL00500
      IF (IPRT.LE.0) IPRT=NFREE                                         SOL00510
      IPRT=MIN(NFREE,IPRT)                                              SOL00520
      IPRT2=MIN(IPRT2,IPRT)                                             SOL00530
C                                                                       SOL00540
      WRITE (6,10) TITLE(1),TITLE(2),OPTION,IPRT                        SOL00550
   10 FORMAT ('1 STRUCTURE....: ',A,//                                  SOL00560
     &        '  SOLUTION.....: ',A,//                                 SOL00570
     &  1X,' SOLUTION #3, DETERMINE EIGENVALUES AND EIGENVECTORS ',//,  SOL00580
     &  1X,'=============================================='//          SOL00590
     &  1X,' OPTION:.....',A/                                           SOL00600
     &  1X,' NUMBER OF EIGENVALUES:.....',I5)                           SOL00610
```

```
C                                                                       SOL00620
C*************************                                              SOL00630
C* FORMULATE STIFFNESS ****                                            SOL00640
C*************************                                              SOL00650
      I1=1                                                              SOL00660
      CALL FORM(I1)                                                     SOL00670
       IF (BTEST(IBUG,5)) CALL LMATRX(STIFF,MD,NDOF ,IMD,               SOL00680
     &    'GLOBAL STIFFNESS MATRIX')                                    SOL00690
C                                                                       SOL00700
C*****************************                                         SOL00710
C* FORM GEOMETRIC STIFFNESS ***                                        SOL00720
C*****************************                                         SOL00730
C  FORMULATE THE GEOMETRIC STIFFNESS IF THE FLAG NEWK IS TRUE, THIS     SOL00740
C       FLAG INITIALIZED ABOVE.                                         SOL00750
C       IF KGDATA(1)=1, THE GEOMETRIC STIFFNESS IS FORMED ONCE, AND     SOL00760
C                       MULTIPLIED BY A SCALER TO ACCOUNT FOR CHANGES   SOL00770
C                       IN THE TOTAL VERTICAL LOAD. THUS THE GEOMETRIC  SOL00780
C                       STIFFNESS ONLY NEEDS TO BE FORMED ONCE.         SOL00790
C       IF KGDATA(1)=2. THE GEOMETRIC STIFFNESS IS BASED ON THE AXIAL   SOL00800
C                       FORCE IN THE ELEMENTS, AND IS RECALCULATED EACH SOL00810
C                       TIME THE AXIAL FORCE CHANGES.                   SOL00820
C       IF KGDATA(3)=1, THE GEOMETRIC STIFFNESS IS FORMED WITH THE      SOL00830
C                       STIFFNESS, THUS A SEPERATE CALL TO FORM IS      SOL00840
C                       NOT NEEDED                                      SOL00850
C       IF KGDATA(3)=2, A SEPERATE GEOMETRIC STIFFNESS MATRIX IS FORMED SOL00860
C                                                                       SOL00870
C      FORMULATE SEPERATE GEOMETRIC STIFFNESS MATRIX                    SOL00880
       IF (IOPT.EQ.2) THEN                                              SOL00890
           KGFORM=2                                                     SOL00900
           KGDATA=0                                                     SOL00910
           I2=2                                                         SOL00920
           CALL FORM(I2)                                                SOL00930
          IF (BTEST(IBUG,5)) CALL LMATRX(KG,MDKG,NDOF,IMDKG,            SOL00940
     &      ' GLOBAL GEOMETRIC STIFFNESS MATRIX')                       SOL00950
        ELSE                                                            SOL00960
          IF (BTEST(IBUG,5)) CALL LMATRX(MASS,MDMS,NDOF,IMDMS,          SOL00970
     &      ' GLOBAL MASS MATRIX')                                      SOL00980
        ENDIF                                                           SOL00990
C                                                                       SOL01000
C*******************************                                      SOL01010
C* CONDENSE OUT NON-FREE DOF ***                                       SOL01020
C*******************************                                      SOL01030
C... IF MCOND IS TRUE THE MASS MATRIX IS ALSO REDUCED BY GUYAN REDUCTIONSOL01040
C... IF KGCOND IS TRUE THEN KG IS ALSO REDUCED                          SOL01050
      IF (NCOND.GT.0) THEN                                              SOL01060
          I1=1                                                          SOL01070
          I1A=1                                                         SOL01080
          IF (IOPT.EQ.2) THEN                                           SOL01090
             MCOND1 =.FALSE.                                            SOL01100
             KCOND1 =KGCOND                                             SOL01110
          ELSE IF (IOPT.EQ.1) THEN                                      SOL01120
             KCOND1 =.FALSE.                                            SOL01130
             MCOND1 =MCOND                                              SOL01140
          ENDIF                                                         SOL01150
C             SET UP DUMMY ZERO LOAD VECTOR                             SOL01160
          DO 28 I=1,NDOF                                                SOL01170
  28          TMP(I)=0                                                  SOL01180
          CALL MSOLL(1,BTEST(IBUG,6),L4,IMD,NDOF,1,L1,L2,               SOL01190
     &                MD,STIFF,TMP,WORK,                                SOL01200
     &                MCOND1,MASS,MDMS,IMDMS,                           SOL01210
     &                KCOND1,KG   ,MDKG,IMDKG,ABORT)                    SOL01220
          IF (ABORT) RETURN                                             SOL01230
       ENDIF                                                            SOL01240
C                                                                       SOL01250
C*****************************************                              SOL01260
C* TRANSFER STIFFNESS INTO A MATRIX ******                             SOL01270
C*****************************************                              SOL01280
C  NOTE: IMSL ROUTINE DGVCSP IS USED TO SOLVE FOR EIGENVALUES,          SOL01290
C        THIS ROUTINE USES A 'FULL SYMMETRIC MATRIX'.                   SOL01300
C                                                                       SOL01310
C  NOTE: THE STORAGE METHOD USED FOR STIFFNESS, MASS AND                SOL01320
C        GEOMETRIC STIFFNESS ARE DIFFERENT.                             SOL01330
C                                                                       SOL01340
      QUIT=.FALSE.                                                      SOL01350
      DO 600 J=L3,L4,+1                                                 SOL01360
        DO 600 I=J,L3,-1                                                SOL01370
          IJ=MD(J)+J-I                                                  SOL01380
          IF(IJ .LT. MD(J+1) .AND. IJ .GE. MD(J) ) THEN                 SOL01390
              A(I-L3+1,J-L3+1)=STIFF(IJ)                                SOL01400
              A(J-L3+1,I-L3+1)=A(I-L3+1,J-L3+1)                         SOL01410
          ELSE IF( IJ .GE. MD(J+1) ) THEN                               SOL01420
              A(I-L3+1,J-L3+1)=0                                        SOL01430
              A(J-L3+1,I-L3+1)=0                                        SOL01440
          ENDIF                                                         SOL01450
C                                                                       SOL01460
          IF (BTEST(IBUG,5)) WRITE (6,29) I,J,A(I,J)                    SOL01470
   29     FORMAT (' I=',I5,' J=',I5,' A(I,J)=',1P,G15.5)                SOL01480
C                                                                       SOL01490
C--------- CHECK FOR ZERO'S ON MAIN DIAGONAL                            SOL01500
          IF (I.EQ.J .AND. A(I-L3+1,J-L3+1).LE.0 ) THEN                 SOL01510
              QUIT=.TRUE.                                               SOL01520
              WRITE (6,32) I,A(I-L3+1,J-L3+1)                           SOL01530
          ENDIF                                                         SOL01540
C                                                                       SOL01550
  600  CONTINUE                                                         SOL01560
   32 FORMAT (5X,'DOF #',I5,' IS .LE. 0 IN THE STIFFNESS MATRIX...',    SOL01570
     &        5X,'K(I,I)=',1P,G12.4,'  SOLN IS ABORTED')                SOL01580
C                                                                       SOL01590
C*************************************************************          SOL01600
C* TRANSFER MASS OR GEOMETRIC STIFFNESS INTO B MATRIX ***              SOL01610
C*************************************************************          SOL01620
      IF (IOPT.EQ.1) THEN                                               SOL01630
      DO 700 J=L3,L4,+1                                                 SOL01640
        DO 700 I=J,L3,-1                                                SOL01650
          IJ=MDMS(J)+J-I                                                SOL01660
          IF(IJ .LT. MDMS(J+1) .AND. IJ .GE. MDMS(J) ) THEN             SOL01670
              B(I-L3+1,J-L3+1)=MASS(IJ)                                 SOL01680
              B(J-L3+1,I-L3+1)=B(I-L3+1,J-L3+1)                         SOL01690
          ELSE IF( IJ .GE. MDMS(J+1) ) THEN                             SOL01700
              B(I-L3+1,J-L3+1)=0                                        SOL01710
              B(J-L3+1,I-L3+1)=0                                        SOL01720
          ENDIF                                                         SOL01730
          IF (BTEST(IBUG,5)) WRITE (6,39) I,J,B(I,J)                    SOL01740
   39     FORMAT (' I=',I5,' J=',I5,' B(I,J)=', 1P,G15.5)               SOL01750
C                                                                       SOL01760
C--------- CHECK FOR ZERO'S ON MAIN DIAGONAL                            SOL01770
          IF (I.EQ.J .AND. B(I-L3+1,J-L3+1).LE.0 ) THEN                 SOL01780
              QUIT=.TRUE.                                               SOL01790
              WRITE (6,42) I,B(I-L3+1,J-L3+1)                           SOL01800
          ENDIF                                                         SOL01810
C                                                                       SOL01820
  700 CONTINUE                                                          SOL01830
   42 FORMAT (5X,'DOF #',I5,' IS .LE. 0 IN THE MASS MATRIX...',         SOL01840
     &        5X,'M(I,I)=',1P,G12.4,'  SOLN IS ABORTED')                SOL01850
C                                                                       SOL01860
      ELSE                                                              SOL01870
      DO 800 J=L3,L4,+1                                                 SOL01880
        DO 800 I=J,L3,-1                                                SOL01890
          IJ=MDKG(J)+J-I                                                SOL01900
          IF(IJ .LT. MDKG(J+1) .AND. IJ .GE. MDKG(J) ) THEN             SOL01910
              B(I-L3+1,J-L3+1)=KG(IJ)                                   SOL01920
              B(J-L3+1,I-L3+1)=B(I-L3+1,J-L3+1)                         SOL01930
          ELSE IF( IJ .GE. MDKG(J+1) ) THEN                             SOL01940
              B(I-L3+1,J-L3+1)=0                                        SOL01950
              B(J-L3+1,I-L3+1)=0                                        SOL01960
          ENDIF                                                         SOL01970
          IF (BTEST(IBUG,5)) WRITE (6,49) I,J,B(I,J)                    SOL01980
   49     FORMAT (' I=',I5,' J=',I5,' B(I,J)=', 1P,G15.5)               SOL01990
C                                                                       SOL02000
C--------- CHECK FOR ZERO'S ON MAIN DIAGONAL                            SOL02010
          IF (I.EQ.J .AND. B(I-L3+1,J-L3+1).LE.0 ) THEN                 SOL02020
              QUIT=.TRUE.                                               SOL02030
              WRITE (6,42) I,B(I-L3+1,J-L3+1)                           SOL02040
          ENDIF                                                         SOL02050
C                                                                       SOL02060
  800 CONTINUE                                                          SOL02070
   52 FORMAT (5X,'DOF #',I5,                                            SOL02080
     &        ' IS .LE. 0 IN THE GEOMETRIC STIFFNESS MATRIX...',        SOL02090
     &        5X,'KG(I,I)=',1P,G12.4,'  SOLN IS ABORTED')               SOL02100
C                                                                       SOL02110
       ENDIF                                                            SOL02120
C                                                                       SOL02130
C---------- RETURN IF A MAIN DIAGONAL IS LE ZERO                        SOL02140
      IF (QUIT) RETURN                                                  SOL02150
C                                                                       SOL02160
C*********************************************                         SOL02170
C* SOLVE FOR EIGENVALUES AND EIGENVECTORS ****                         SOL02180
C*********************************************                         SOL02190
C--------- IMSL SU_BROUTINE DGVCSP ---------------------------------- SOL02200
C  EIGENVALUES AND EIGENVECTORS OF A*X-LAMBDA*B*X=0                     SOL02210
C  A  = STIFFNESS - REAL, FULL SYMMETRIC MATRIX                         SOL02220
C  B  = MASS OR GEOMETRIC STIFFNESS - REAL, FULL SYMMETRIC MATRIX       SOL02230
C  N  = NUMBER OF DOF OF A AND B                                        SOL02240
C  PI = PERFORMANCE INDEX                                               SOL02250
C         PI=WK(1),    PI<1    - ROUTINE PERFORMED WELL                SOL02260
C                  1<PI<100 - ROUTINE PERFORMED SATISFACTORILY          SOL02270
C                100<PI     - ROUTINE PERFORMED POORLY                 SOL02280
C  EVAL= EIGENVALUES (LAMBDA), N BY 1                                   SOL02290
C  EVEC= EIGENVECTORS, N BY N                                           SOL02300
C  LDA = LEADING DIMENSION OF MATRIX A EXACTLY AS SPECFIED IN THE       SOL02310
C        DIMENSION STATEMENT IN THE CALLING PROGRAM.                    SOL02320
C  LDB = LEADING DIMENSION OF MATRIX B EXACTLY AS SPECFIED IN THE       SOL02330
C        DIMENSION STATEMENT IN THE CALLING PROGRAM.                    SOL02340
CLDEVEC= LEADING DIMENSION OF MATRIX EVEC EXACTLY AS SPECFIED IN THE    SOL02350
C        DIMENSION STATEMENT IN THE CALLING PROGRAM.                    SOL02360
C---------------------------------------------------------------------- SOL02370
C                                                                       SOL02380
      LDEVEC=NFREE                                                      SOL02390
C     CALL IWKIN(100000)                                                SOL02400
C     CALL DGVCSP(NFREE,A,NFREE,B,NFREE,EVAL,EVEC,LDEVEC)               SOL02410
C     PI = DGPISP(NFREE,NFREE,A,NFREE,B,NFREE,EVAL,EVEC,LDEVEC)         SOL02420
C     WORK(1)=PI                                                        SOL02430
      CALL DG2CSP(NFREE,A,NFREE,B,NFREE,EVAL,EVEC,LDEVEC, A,WK2,WORK)   SOL02440
C     WRITE (6,*) ' PREFORMANCE INDEX=',WORK(1)                         SOL02450
C                                                                       SOL02460
C*********************************************                         SOL02470
C* NORMALIZE MODE SHAPE TO MAX VALUE OF ONE ***                         SOL02480
C*********************************************                         SOL02490
C                                                                       SOL02500
C NOTE: THE EIGENVECTOR IN IMSL DGVCSP HAS BEEN NORMALIZED              SOL02510
C                                                                       SOL02520
C-----------------------------------------                              SOL02530
C* PRINT EIGENVALUES AND EIGENVECTORS ***                              SOL02540
C-----------------------------------------                              SOL02550
      IST =1                                                            SOL02560
  100 IEND=MIN(IST+6,IPRT)                                              SOL02570
C                                                                       SOL02580
      WRITE (6,110) (I            ,I=IST,IEND)                        SOL02590
      IF (IOPT.EQ.2) THEN                                               SOL02600
          WRITE (6,120) (EVAL(I),I=IST,IEND)                            SOL02610
      ELSE IF (IOPT.EQ.1) THEN                                          SOL02620
          DO 125 I=IST,IEND                                             SOL02630
  125       EVAL(I)=SQRT( EVAL(I) )                                     SOL02640
          WRITE (6,126) (EVAL(I),I=IST,IEND)                            SOL02650
          DO 127 I=IST,IEND                                             SOL02660
  127       EVAL(I)= EVAL(I)  / 6.2831853                               SOL02670
          WRITE (6,128) (EVAL(I),I=IST,IEND)                            SOL02680
          DO 129 I=IST,IEND                                             SOL02690
            IF (EVAL(I).EQ.0) EVAL(I)=1.E-15                            SOL02700
  129       EVAL(I)= 1.00 / EVAL(I)                                     SOL02710
          WRITE (6,130) (EVAL(I),I=IST,IEND)                            SOL02720
      ENDIF                                                             SOL02730
      WRITE (6,135)                                                     SOL02740
  110 FORMAT('            MODE=',7(I10,5X))                             SOL02750
  120 FORMAT('     EIGENVALUE=',1P,7(G14.5,1X) )                        SOL02760
  126 FORMAT(' FREQ (RAD/SEC)=',1P,7(G14.5,1X) )                        SOL02770
  128 FORMAT(' FREQUENCY (HZ)=',1P,7(G14.5,1X) )                        SOL02780
  130 FORMAT('   PERIOD (SEC)=',1P,7(G14.5,1X) )                        SOL02790
  135 FORMAT('   EIGENVECTORS=')                                        SOL02800
C                                                                       SOL02810
C-------- EIGENVECTORS                                                 SOL02820
      DO 140 I=1,NFREE                                                  SOL02830
        II=I+NCOND                                                      SOL02840
  140   WRITE (6,150) II,(EVEC(I,J),J=IST,IEND)                         SOL02850
  150 FORMAT('      DOF(',I6,')',1P,7(G14.5,1X) )                       SOL02860
C                                                                       SOL02870
      IF (IEND.LT.IPRT ) THEN                                           SOL02880
          WRITE (6,10) TITLE(1),TITLE(2),OPTION,IPRT                    SOL02890
          IST=IEND+1                                                    SOL02900
          GO TO 100                                                     SOL02910
      ENDIF                                                             SOL02920
C                                                                       SOL02930
C*****************************************                              SOL02940
C* CALC EIGEN VECTORS FOR NONDYNAMIC DOF ***                           SOL02950
C*****************************************                              SOL02960
C                                                                       SOL02970
C------ TRANSFER EIGENVECTORS TO TEMP STORAGE                          SOL02980
C                                                                       SOL02990
      DO 500 IFREQ=1,IPRT2                                              SOL03000
C                                                                       SOL03010
C------ TRANSFER EIGENVECTORS TO TEMP STORAGE                          SOL03020
        DO 200 J=L1,L2                                                  SOL03030
  200     WORK(J)=0                                                     SOL03040
        DO 210 J=L3,L4                                                  SOL03050
  210     WORK(J)=EVEC(J-L2,IFREQ)                                      SOL03060
        DO 220 J=L5,L6                                                  SOL03070
  220     WORK(J)=0                                                     SOL03080
C                                                                       SOL03090
C------ CALC EIGENVECTORS FOR CONDENSED OUT DOF                        SOL03100
        CALL REACTN(I1,BTEST(IBUG,6),NNODE,L1,L2,L3,L4,TMP,WORK,        SOL03110
     &        MD,STIFF,IMD,NDOF,1,1.0,                                  SOL03120
     &        MAXNOD,IDOF,COORD,ID,TITLE,STEPID,.FALSE.,                SOL03130
     &        NCOS,COSINE,JTCOS,JTFLG,SUMRCT)                           SOL03140
        CALL MSOLL(3,BTEST(IBUG,6),L4,IMD,NDOF,I1,L1,L2,                SOL03150
     &             MD,STIFF,TMP,WORK,                                   SOL03160
     &            .FALSE.,MASS,MDMS,1,                                  SOL03170
     &            .FALSE.,KG   ,MDKG,1,ABORT)                           SOL03180
        IF (ABORT) RETURN                                               SOL03190
C                                                                       SOL03200
C------- NORMALIZE MODE SHAPE TO MAX VALUE OF ONE                       SOL03210
        AMIN=0                                                          SOL03220
        AMAX=0                                                          SOL03230
        DO 230 I=L1,L6                                                  SOL03240
          AMIN=MIN(AMIN,WORK(I))                                        SOL03250
  230     AMAX=MAX(AMAX,WORK(I))                                        SOL03260
        IF (-AMIN.GT.AMAX) AMAX=AMIN                                    SOL03270
C                                                                       SOL03280
        DO 240 I=L1,L6                                                  SOL03290
  240     WORK(I)=WORK(I)/AMAX                                          SOL03300
C                                                                       SOL03310
C------- PRINT EIGENVALUE AT EACH NODE                                  SOL03320
        IF (IOPT.EQ.2) WRITE (STEPID,250) IFREQ,EVAL(IFREQ)             SOL03330
          IF (EVAL(IFREQ).EQ.0) EVAL(IFREQ)=1.E-15                      SOL03340
        FREQ=1.00/EVAL(IFREQ)                                           SOL03350
        IF (IOPT.EQ.1) WRITE (STEPID,260) IFREQ,FREQ                    SOL03360
C                                                                       SOL03370
        CALL JOIDSP(1,MAXNOD,NDOF,NNODE,1,L3,L4,ID,                     SOL03380
     &        IDOF,CNST,WORK,                                           SOL03390
     &        TITLE,'EIGENVECTORS',STEPID,.TRUE.,                       SOL03400
     &        NCOS,COSINE,JTCOS,JTFLG)                                  SOL03410
  250   FORMAT ('MODE #',I3,'   EIGENVALUE=',1P,G15.6)                  SOL03420
  260   FORMAT ('MODE #',I3,'   FREQUENCY=',1P,G15.6)                   SOL03430
C                                                                       SOL03440
C------- PRINT EIGENVALUE TO DISK FOR PLOTTING                          SOL03450
```

```
C
      IF(BTEST(IBUG,2)) THEN
      DO 930 I=1,NNODE
C............. TRANSFER EIGENVECTOR TO LOCAL VECTOR .............
         DO 920 J=1,6
           IJ=IDOF(I,J)
  920      D(J)=WORK(IJ)
C
C............. ADD ROTATION EIGENVECTORS TO DISPL FOR CONSTRAINTS
C............. IF UNCONSTRAINED, CNST(I,J)=0
         D(1) = D(1) + D(5)*CNST(I,3) + D(6)*CNST(I,5)
         D(2) = D(2) + D(4)*CNST(I,1) + D(6)*CNST(I,6)
         D(3) = D(3) + D(4)*CNST(I,2) + D(5)*CNST(I,4)
C
C............. TRANSFER TO GLOBAL DISPLACEMENTS .....
         JCOS=JTCOS(I)
         DO 925 I1=1,3
           I2=I1+3
           G(I1)=0
           G(I2)=0
           DO 925 J1=1,3
              J2=J1+3
              G(I1)=G(I1)+COSINE(J1,I1,JCOS)*D(J1)
  925         G(I2)=G(I2)+COSINE(J1,I1,JCOS)*D(J2)
C
C............. WRITE DISPL
         WRITE (10+IFREQ,926) I, (G(K),K=1,6)
  926    FORMAT (I7,1P,6G12.4)
  930 CONTINUE
      ENDIF
C
  500 CONTINUE
C
      RETURN
C
      END
      SUBROUTINE SOL04
      LOGICAL BTEST,UNBAL
      INCLUDE (%COMM)
C
C=====================================
C= SET GEOMETRIC STIFFNESS FLAGS ===
C=====================================
      %(I%KGDT)=0.00
      KGLOAD=N%(I%KGDT+1)
      KGTYPE=N%(I%KGDT+2)
      KGFORM=N%(I%KGDT+3)
C
C=================================
C= READ INFO AND SET OPTIONS ===
C=================================
      READ  (5,*) MAXELD,IPRNT,IWRITE,UNBAL
      NLOAD=1
      IPRNT=MAX(IPRNT,1)
C
C==================
C= PRINT HEADER ===
C==================
      WRITE (6,101) TITLE(1),TITLE(2),IPRNT,IWRITE
  101 FORMAT ('1 STRUCTURE....: ',A,/
     &        '  SOLUTION.....: ',A,//
     &1X,' SOLUTION #4, STATIC NONLINEAR SOLUTION '/
     &1X,'========================================='/
     &1X,' INTERVAL FOR PRINTING DATA.............',I5/
     &1X,' INTERVAL FOR WRITING DATA TO FILE......',I5/)
C
      IF (KGFORM.EQ.2) WRITE (6,11)
   11 FORMAT (1X,' GEOMETRIC STIFFNESS IS NOT INCLUDED, ',
     &        ' USE KGDATA(3)=1 TO INCLUDE GEOMETRIC STIFFNESS.')
      IF (      UNBAL) WRITE (6,13)  ' ARE '
      IF (.NOT.UNBAL) WRITE (6,13)  ' ARE NOT '
   13 FORMAT(' UNBALANCED JOINT FORCES',A,'ADDED TO THE NEXT CYCLE')
C
C=======================================
C=INPUT FILE OUTPUT CONTROLL DATA ===
C=======================================
      IF (IWRITE.GT.0) THEN
         IOPT=1
         TO=0
         DT=0
         CALL DMPDAT(IOPT,IWRITE,TO,DT)
      ENDIF
C
C=============================================
C= INITILIZE STORAGE REQ'D FOR THIS SOLN ===
C=============================================
C     %(I%LOAD) = LOCATION OF THE TOTAL LOAD MATRIX
C     %(I%DISP) = LOCATION OF THE TOTAL DISPLACEMENT MATRIX
C     %(I%VEL ) = LOCATION OF THE TOTAL VELOCITY MATRIX
C     %(I%DLOA) = LOCATION OF THE INCRIMENTAL LOAD MATRIX
C     %(I%DDSP) = LOCATION OF THE INCRIMENTAL DISPLACEMENT MATRIX
C     %(I%DVEL) = LOCATION OF THE INCRIMENTAL VELOCITY MATRIX
C     %(I%Q   ) = LOCATION OF THE JOINT LOADS
C     %(I%R   ) = LOCATION OF THE ELEMENT LOADS
C
       IF (I%LOAD.EQ.0) THEN
           I%LOAD = I%
           I%     = I%+NDOF
           DO 1 I=1,NDOF
    1      %(I+I%LOAD-1) =0
           ENDIF
C
       IF (I%DISP.EQ.0) THEN
           I%DISP = I%
           I%     = I%+NDOF
           DO 2 I=1,NDOF
    2      %(I+I%DISP-1) =0
           ENDIF
C
       IF (I%VEL .EQ.0) THEN
           I%VEL  = I%
           I%     = I%+NDOF
           DO 3 I=1,NDOF
    3      %(I+I%VEL -1) =0
           ENDIF
C
      IF (I%DLOA.EQ.0) THEN
          I%DLOA = I%
          I%     = I%+NDOF
          DO 5 I=1,NDOF
    5     %(I+I%DLOA-1) =0
          ENDIF
C
      IF (I%DDSP.EQ.0) THEN
          I%DDSP = I%
          I%     = I%+NDOF
          DO 6 I=1,NDOF
    6     %(I+I%DDSP-1) =0
          ENDIF
C
      IF (I%DVEL.EQ.0) THEN
          I%DVEL = I%
          I%     = I%+NDOF
          DO 7 I=1,NDOF
    7     %(I+I%DVEL-1) =0
          ENDIF
C
          I%Q    = I%
          I%     = I%+NDOF
          DO 8 I=1,NDOF
    8     %(I+I%Q-1) =0
C
          I%R    = I%
          I%     = I%+NDOF
          DO 9 I=1,NDOF
    9     %(I+I%R-1) =0
C
      IF (I%FUB .EQ.0) THEN
          I%FUB  = I%
          I%     = I%+NDOF
          ENDIF
          DO 10 I=1,NDOF
   10     %(I+I%FUB-1) =0
C
          I%UBLD = I%
          I%     = I%+NDOF
          I%UBDP = I%
          I%     = I%+NDOF
          DO 12 I=1,NDOF
          %(I+I%UBLD-1) =0
   12     %(I+I%UBDP-1) =0
C
      IW=I%+1
C
C---- DETERMINE THE LENGTH OF THE STIFFNESS MATRIX
            IMD=N%(I%MD+NDOF) - NZ(I%MD) +1
C
C      ---- SET UP CONSTANTS ------------------
      L1=1
      L2=NCOND
      L3=L2+1
      L4=NCOND +NFREE
      L5=L4+1
      L6=NDOF
C
      NLOAD=1
C
      ELSTIC=.FALSE.
C
      IF (MAXELD.GT.0) THEN
           I%IELD=I%
           I%=I% + MAXELD*5
           I%ELD=I%
           I%=I% + MAXELD*12
      ENDIF
C
C=======================================
      I%FG   = I%
      I%     = I%+NDOF
      I%DFG  = I%
      I%     = I%+NDOF
      I%DMAX= I%
      I%     = I%+NDOF
      I%RMAX= I%
      I%     = I%+NDOF
      I%DRLO= I%
      I%     = I%+NDOF
      I%RLOA= I%
      I%     = I%+NDOF
      DO 27 I=0,NDOF-1
        %(I%FG  +I)=0
        %(I%DFG +I)=0
        %(I%DMAX+I)=0
        %(I%RMAX+I)=0
        %(I%RLOA+I)=0
   27   %(I%DRLO+I)=0
C
C=======================================
C
C------ CHECK MEMORY
      CALL CKSTOR(NDOF,IW)
C
C------ SET UP CONSTANTS
      IMD   =N%(I%MD   +NDOF)
      IMDMS =N%(I%MDMS+NDOF)
      IMDKG =N%(I%MDKG+NDOF)
      IMDMSL=1
      IMDS  =1
C
C=========================
C= ZERO GROUND MOTION ===
C=========================
      DO 18 I=1,6
   18   SUMRCT(I)=0.
C
C=========================
C= ZERO GROUND MOTION ===
C=========================
      DO 19 I=1,9
   19   GROUND(I)=0.
C
C=========================
C= CALC PRINTING INFO ===
C=========================
      IPDISP=6+NNODE
      IPRCTN=7
      DO 20 I=1,NNODE
        DO 15 J=0,5
        IF (BTEST(N%(I%JFLG-1+I),J)) THEN
          IPRCTN=IPRCTN+1
          GO TO 20
        ENDIF
   15   CONTINUE
   20 CONTINUE
      IPELEM=NELMT
      LSTTYP=0
      DO 26 I=1,NELMT
        J=N%(I%ELE+I)
        IF (N%(J).NE.LSTTYP) IPELEM=IPELEM+4
   26   LSTTYP=N%(J)
  111 FORMAT (0P,' %MATRX(',I6,')=',I15,1P,G30.10)
C
      CALL SOL04A
     & (L1,L2,L3,L4,L5,L6,IMD,IMDMS,IMDKG,IMDS,IMDMSL,ABORT,
     &  MAXELD   , IPRNT    , IWRITE   , UNBAL    ,
     &  NLOAD    , NNODE    , NCOS     , NELMT    ,
     &  I%DDSP   , I%DLOA   , IBUG     , MAXNOD   ,
     &  TITLE    , STEPID   , NDOF     , NFREE    , %(I%RLOA),
     &  KGFORM   , KGLOAD   , KGTYPE   , NCOND    , %(I%DRLO),
     &  NEWKG    , NEWK     , MCOND    , KGCOND   ,   SUMRCT ,
     &  IPDISP   , IPRCTN   , IPELEM   , %(I%DMAX), Z(I%RMAX),
     &  %(I%DISP), %(I%VEL ), %(I%LOAD), %(I%Q   ), %(I%FUB ),
     &  %(I%DDSP), %(I%DVEL), %(I%DLOA), %(I%R   ), %(IW    ),
     &  %(I%STIF), %(I%MASS), %(I%KG   ), %(I%ELD ), %(I%UBLD),
     & N%(I%MD  ),N%(I%MDMS),N%(I%MDKG),N%(I%ELD), %(I%UBDP),
     & N%(I%ID  ),N%(I%IDOF), %(I%CNST),N%(I%JFLG), %(I%CORD),
     &  %(I%COS ),N%(I%JCOS), EIE,ESE,PSE,EKE,EDD ,
     &  %(I%FG  ),N%(I%DFG ))
      RETURN
C
      END
C@PROCESS SDUMP GOSTMT XREF
C-------------------------------------------------------------------------------
      SUBROUTINE SOL04A
     & (L1,L2,L3,L4,L5,L6,IMD,IMDMS,IMDKG,IMDS,IMDMSL,ABORT,
     &  MAXELD   , IPRNT    , IWRITE   , UNBAL    ,
     &  NLOAD    , NNODE    , NCOS     , NELMT    ,
     &  I%DDSP   , I%DLOA   , IBUG     , MAXNOD   ,
     &  TITLE    , STEPID   , NDOF     , NFREE    , RLOAD    ,
     &  KGFORM   , KGLOAD   , KGTYPE   , NCOND    , DRLOA    ,
     &  NEWKG    , NEWK     , MCOND    , KGCOND   , SUMRC    ,
     &  IPDISP   , IPRCTN   , IPELEM   , DSPMAX   , RCTMAX   ,
     &  DISP     , VEL      , LOAD     , Q        , FUB      ,
     &  DDISP    , DVEL     , DLOAD    , R        , WORK     ,
```

```
     &  STIFF    , MASS     , KG       , SLD      , UBLOAD   ,
     &  MD       , MDMS     , MDKG     , IELD     , UBDISP   ,
     &  ID       , IDOF     , CNST     , JTFLG    , COORD    ,
     &  COSINE   , JTCOS    ,  EIE,ESE,PSE,EKE,EDD,
     &  FG       , DFG      )
C
      DOUBLE PRECISION  EIE,ESE,PSE,EDD,EKE
C
      COMMON /IDSTEP/ISTEP
      CHARACTER*(*) TITLE(2),STEPID
      LOGICAL PRINT,DSPFLG,HEAD,REPEAT,AXIAL,TEST,NEWKDT,STEST,ABORT
      LOGICAL NEWKG,NEWK,MCOND,KGCOND,UNBAL,MCND2,KGCND2,WRITE
      LOGICAL BUG3,BUG5,BUG6,BUG11
      CHARACTER*80 NAME
      CHARACTER*20 INTEG
      REAL LOAD,MASS,MASSSL,KG,MCOS
      INTEGER FDAMP,FMASS
      DIMENSION SUMRC(6),DSUMRC(6),SUMFD(6),DSUMFD(6),SUMRCM(6),SUMUB(6)
      DIMENSION DSPMAX(L6),RCTMAX(L6),DRLOA(L6),RLOAD(L6)
      DIMENSION RINPUT(100)
      DIMENSION STIFF(IMD) ,MASS(IMDMS),  KG(IMDKG)
      DIMENSION    MD(L6+1),MDMS(L6+1 ),MDKG(L6+1 )
      DIMENSION  LOAD(L6), DISP(L6), VEL(L6), Q(L6),FG(L6),DFG(L6)
      DIMENSION DLOAD(L6),DDISP(L6),DVEL(L6), R(L6)
      DIMENSION   FUB(L6),UBLOAD(L6),UBDISP(L6)
      DIMENSION WORK(2*L6)
      DIMENSION COORD(MAXNOD,3),ID(MAXNOD),
     &          IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),CNST(MAXNOD,6),
     &          JTFLG(MAXNOD),JTCOS(MAXNOD)

      DIMENSION IELD(1),ELD(1)
      DIMENSION  GA(3), GV(3), GD(3)
      DIMENSION GAT(3),GVT(3),GDT(3)
      LOGICAL NEWFAC
      CHARACTER*10 STEP,STEP2
C
      SUMUB(1)=0
      SUMUB(2)=0
      SUMUB(3)=0
      SUMUB(4)=0
      SUMUB(5)=0
      SUMUB(6)=0
      SUMFOR=0
      SUMMOM=0
      SUMMPO=0
      SUMMMO=0
C--------------------------------
      BUG3 =BTEST(IBUG, 3)
      BUG5 =BTEST(IBUG, 5)
      BUG6 =BTEST(IBUG, 6)
      BUG11=BTEST(IBUG,11)
C--------------------------------
C
C--------------------------------
C= INITIALIZE ENERGY CONSTANTS ===
C--------------------------------
      EIE=0
      ESE=0
      PSE=0
      EKE=0
      EDD=0
C
C-----------------------------------
C= SET GEOMETRIC STIFFNESS FLAGS ===
C-----------------------------------
      NEWKG=.TRUE.
      IF (KGLOAD.EQ.0 .OR. KGTYPE.EQ.0 .OR. KGFORM.EQ.0) NEWKG=.FALSE.
C
C------------------------
C= SET STIFFNESS FLAGS ===
C------------------------
      NEWK = .TRUE.
      KSAME= 0
C
C-----------------------
C= INPUT LOADING DATA ==
C-----------------------
C---------- MATRICIES Q AND R ARE ZEROED IN THE CALLING PGM.
C
C       ----- INPUT JOINT LOADS --------------------
C       JOINT LOADS AND SUPPORT DISPLACEMENTS ARE STORED IN Q
      CALL JOILOA(BUG3,MAXNOD,NDOF,NNODE,NLOAD,L4,
     &     ID,IDOF,CNST,Q, DSPFLG,JTFLG,TITLE,.FALSE.)
C
C
C       ----- INPUT ELEMENT LOADS ------------------
C       ELEMENT LOADS ARE STORED IN R
      IF (MAXELD.GT.0)
     &   CALL ELRLOA(BUG3,L5,NELMT,MAXELD,NELD,NDOF,NLOAD,
     &   R,IELD,ELD,I3DISP,I3LOAD,   NAME,TITLE,.FALSE.)
C
C
C--------------------------------------
C=WRITE INITIAL DATA TO OUTPUT FILE ===
C--------------------------------------
      IF (INRITE.GT.0) THEN
         IOPT=2
         CALL DMPDAT(IOPT,INWRITE,TO,DT)
      ENDIF
C
C
C=====================================================================
C= LOOP FOR EACH TIME STEP , ISTEP=CURRENT STEP  ====================
C=====================================================================
C... ISTEP = STEP NUMBER
C    FACTOR= CURRENT TARGET LOAD = FACTOR*(INPUT LOAD)
C    FACTL = PREVIOUS TRAGET LOAD
C    FACT2 = NEXT TRAGET LOAD
C    DF    = LOAD STEP
C    DFL   = PREVIOUS LOAD STEP
C    DFN   = NEXT LOAD STEP
C    PMAX  = MAXIMUN ALLOWABLE INCRIMENTAL LOAD / STEP
C    DMAX  = MAXIMUN ALLOWABLE INCRIMENTAL DISPLACMENT / STEP

      ISTEP=0
      FACTL=0
      DF=0
      READ(5,*) STEP2,FACT2,PMAX2,DMAX2,NUMB2
 1000 STEP  =STEP2
      IF (TEST(STEP,'END',.FALSE.)) GO TO 5000
      FACTOR=FACT2
      PMAX  =PMAX2
      DMAX  =DMAX2
      NUMB  =NUMB2
      NUMBDF=NUMB2
      READ(5,*,END=1010) STEP2,FACT2,PMAX2,DMAX2,NUMB2
      GO TO 1020
 1010 STEP2='END'
      FACT2=FACTOR
 1020 IF (TEST(STEP,'END',.FALSE.)) GO TO 5000
      IF (FACTOR.EQ.FACTL) GO TO 1000
C
 1100   ISTEP=ISTEP+1
        WRITE (STEPID,90) STEP,ISTEP,FACTOR
        IF (BUG3.OR.BUG5.OR.BUG6.OR.BUG11) THEN
          WRITE (6,91) STEP,ISTEP,FACTOR
          WRITE (6,*) 'NEWK: ',NEWK,'  NEWKG: ',NEWKG
        ENDIF
   90   FORMAT (A,' STEP:',I5,' FACTOR:',1P,G11.4)
   91   FORMAT ('1 ',A,' STEP:',I5,' FACTOR:',1P,G11.4)
C
C-----------------------------
C= CALC INCRIMENTAL LOADS  ==
C-----------------------------
C 1) ADD ELEMENT LOADS, AND JOINT LOADS ON FREE JOINTS
C    SAVE IN DLOA FOR CALCULATING UNBALANCED FORCES FOR THE NEXT STEP.
C 2) PUT JOINT DISPLACEMENTS IN DDISP FOR RESTRAINED JOINTS.
C 3) PUT THE NEGATIVE ELEMENT LOAD IN DLOA FOR RESTRAINED JOINTS, THIS
C    IS USED TO CALCULATE REACTIONS.
C 4) ZERO THE UNBALANCED LOAD VECTOR
C    IS USED TO CALCULATE REACTIONS.
         DFL=DF
         DF =FACTOR-FACTL
         DFN=FACT2 -FACTOR
         IF (PMAX.EQ.0 .AND. DMAX.EQ.0 .AND. NUMB.GT.0) DF=DF/NUMB
         IF (DF.EQ.0) GO TO 1000
C
           DO 106 I=L1,L4
  106        DLOAD(I)=(Q(I)+R(I))*DF
C
C       ...SWAP THE SIGN OF THE FEP AT REACTIONS
         DO 110 I=L5,L6
            DLOAD(I)=-R(I)*DF
  110       DDISP(I)= Q(I)*DF
C
C-----------------------------
C= FORMULATE STIFFNESS ======
C-----------------------------
C  FORMULATE THE STIFFNESS IF THE FLAG NEWK IS TRUE, THIS FLAG IS
C       INITIALIZED ABOVE, AND SET WHEN THE ELEMENT FORCES ARE CALC.
C       THE STIFFNESS IS ALSO FORMULATED IF KG IS TO BE FORMED
      REPEAT=.TRUE.
      KSAME=0
  100 IF (NEWK.OR.NEWKG) THEN
        I1=1
        CALL FORM(I1)
        IF (KGLOAD.EQ.1) NEWKG=.FALSE.
      ENDIF
C
C-----------------------------------------------
C= MODIFY LOADS FOR RESTRAINT DISPLACEMENTS ===
C-----------------------------------------------
      IF (DSPFLG) THEN
C          ----- ZERO INCRIMENTAL REACTION LOAD VECTOR ----
           DO 113 I=L1,L4
  113        DRLOA(I)=0
        CALL REACTN(1,BUG3,NNODE,L1,L4,L5,L6,DRLOA,DDISP,
     &  MD,STIFF,IMD,NDOF,1,1.0,
     &  MAXNOD,IDOF,COORD,ID,TITLE,STEPID,.FALSE.,
     &           NCOS,COSINE,JTCOS,JTFLG,SUMRC)
C          ----- ADD INCRIMENTAL REACTION LOADS TO OTHER LOADS
           DO 114 I=L1,L4
  114        DLOAD(I) = DLOAD(I) + DRLOA(I)
      ENDIF
C
C    SAVE IN LOADS IN DDSP FOR CALCULATING FREE DISPL FOR THIS STEP.
         DO 115 I=L1,L4
  115       DDISP(I)= DLOAD(I)
C
C-----------------------------
C= SOLVE FOR FREE DISPL =====
C-----------------------------
      IF (BUG5) THEN
         IF (NEWK .OR. NEWKG)
     &   CALL LMATRX(STIFF,MD,NDOF,IMD,'GLOBAL STIFFNESS DUMP=7')
         CALL WMATRX(DLOAD,NDOF,NLOAD,'LOAD MATRIX  DUMP=7')
      ENDIF
C
         IF (NEWK .OR. NEWKG) THEN
           CALL MSOLL(1,.FALSE.,L4,IMD,NDOF,NLOAD,L1,L4,
     &     MD,STIFF,DDISP,WORK,.FALSE.,MASS,MDMS,1,.FALSE.,KG,MDKG,1,
     &     ABORT)
           IF (ABORT) GO TO 5000
         ELSE
           CALL MSOLL(2,.FALSE.,L4,IMD,NDOF,NLOAD,L1,L4,
     &     MD,STIFF,DDISP,WORK,.FALSE.,MASS,MDMS,1,.FALSE.,KG,MDKG,1,
     &     ABORT)
           IF (ABORT) GO TO 5000
         ENDIF
         CALL MSOLL(3,.FALSE.,L4,IMD,NDOF,NLOAD,L1,L4,
     &     MD,STIFF,DDISP,WORK,.FALSE.,MASS,MDMS,1,.FALSE.,KG,MDKG,1,
     &     ABORT)
         IF (ABORT) GO TO 5000
C
       IF (BUG5)
     &    CALL WMATRX(DDISP,NDOF,NLOAD,'DISPLACEMENT  DUMP=7')
C
C-----------------------------------------
C= SCALE SOLN SUCH THAT PM LE PMAX, DM LE DMAX ==
C-----------------------------------------
      DM=0
      PM=0
      DO 130 I=L3,L4
         PM=MAX(DLOAD(I),-DLOAD(I),PM)
  130    DM=MAX(DDISP(I),-DDISP(I),DM)
      IF ((DM.GT.DMAX .AND. DMAX.NE.0).OR.
     &    (PM.GT.PMAX .AND. PMAX.NE.0) ) THEN
         F1=0
         F2=0
         IF (DMAX.NE.0) F1=DM/DMAX
         IF (PMAX.NE.0) F2=PM/PMAX
         F=MAX(F1,F2)
         DF=DF/F
         NEWFAC=.FALSE.
         DO 140 I=L1,L6
           DRLOA(I)=DRLOA(I)/F
           DLOAD(I)=DLOAD(I)/F
  140      DDISP(I)=DDISP(I)/F
           DFDFN=1
      ELSE IF (PMAX.EQ.0 .AND. DMAX.EQ.0 .AND. NUMB.GT.1) THEN
        NEWFAC=.FALSE.
        NUMB=NUMB-1
      ELSE
        NEWFAC=.TRUE.
        DFDFN=DF*DFN
      ENDIF
C
      AFACT=FACTL+DF
C
C-----------------------------------------
C= SOLVE FOR DISPL DUE TO UNBALANCED FORCES ===
C-----------------------------------------
C
C..... USE FIRST STIFFNESS TO CALCULATE UNBALANCED FORCES.
C       IF THE STEP IS REPEATED, DO NOT USE THE NEW STIFFNESS.
      IF (UNBAL.AND.REPEAT) THEN
        DO 104 I=L1,L4
          UBLOAD(I) = -FUB(I)
  104     UBDISP(I) = -FUB(I)
        DO 105 I=L5,L6
          UBLOAD(I) = -FUB(I)
  105     UBDISP(I) = 0.00
        IF (BUG5)
     &    CALL WMATRX(UBDISP,NDOF,NLOAD,'UNBALANCED LOAD MATRIX')
        CALL MSOLL(2,.FALSE.,L4,IMD,NDOF,NLOAD,L1,L4,
     &    MD,STIFF,UBDISP,WORK,.FALSE.,MASS,MDMS,1,.FALSE.,KG,MDKG,1,
     &    ABORT)
        IF (ABORT) GO TO 5000
        CALL MSOLL(3,.FALSE.,L4,IMD,NDOF,NLOAD,L1,L4,
     &    MD,STIFF,UBDISP,WORK,.FALSE.,MASS,MDMS,1,.FALSE.,KG,MDKG,1,
     &    ABORT)
        IF (ABORT) GO TO 5000
      IF (BUG5)
     &    CALL WMATRX(UBDISP,NDOF,NLOAD,'UNBALANCED DISPLACEMENT')
```

```
      ENDIF                                                             SOL02980
C                                                                       SOL02990
C..... ADD UNBALANCED FORCES AND DISPL TO STRUCTURAL LOAD AND DISPL.    SOL03000
      IF (UNBAL) THEN                                                   SOL03010
        DO 107 I=L1,L4                                                  SOL03020
          DDISP(I)=DDISP(I)+ UBDISP(I)                                  SOL03030
  107   CONTINUE                                                        SOL03040
        DO 109 I=L5,L6                                                  SOL03050
          DLOAD(I)=DLOAD(I)- UBLOAD(I)                                  SOL03060
  109   CONTINUE                                                        SOL03070
      ENDIF                                                             SOL03080
C                                                                       SOL03090
C..... CALCULATE THE NORM OF THE UNBALANCED FORCES                      SOL03100
      SUM=0                                                             SOL03110
      DO 108 I=L1,L6                                                    SOL03120
        SUM=SUM+FUB(I)**2                                               SOL03130
  108   FUB(I)=0.00                                                     SOL03140
        FNORM = SQRT(SUM/L6)                                            SOL03150
C                                                                       SOL03160
C=========================                                            SOL03170
C= CALC PSUDO-VELOCITY ===                                            SOL03180
C=========================                                            SOL03190
      IF (ISTEP.EQ.1) THEN                                              SOL03200
        DO 120 I=L1,L6                                                  SOL03210
          VEL(I)=Q(I)+R(I)                                              SOL03220
  120     DVEL(I)=Q(I)+R(I)                                             SOL03230
        PSUVEL=2                                                        SOL03240
        DVEL1 =1                                                        SOL03250
      ELSE IF (FACTOR.EQ.AFACT .AND.                                    SOL03260
     &     .NOT.((FACTL.LT.FACTOR .AND. FACTOR.LT.FACT2) .OR.           SOL03270
     &           (FACTL.GT.FACTOR .AND. FACTOR.GT.FACT2))) THEN         SOL03280
        PSUVEL=-PSUVEL                                                  SOL03290
        DO 121 I=L1,L6                                                  SOL03300
  121     DVEL(I)= PSUVEL * (Q(I)+R(I))                                 SOL03310
        DVEL1 =1                                                        SOL03320
      ELSE IF (DVEL1.NE.0) THEN                                         SOL03330
        DO 122 I=L1,L6                                                  SOL03340
  122     DVEL(I)=0                                                     SOL03350
          DVEL1 =0                                                      SOL03360
      ENDIF                                                             SOL03370
C                                                                       SOL03380
C=========================================                            SOL03390
C= CALCULATE INCRIMENTAL MEMBER FORCES ===                            SOL03400
C=========================================                            SOL03410
C ELE_LIB CALLS THE INDIVIDUAL ELEMENT LIBRARIES, EACH OF WHICH DETERMINSOL03420
C  1) DID THE STIFFNESS CHANGE?  IF SO; NEWK=.TRUE.                     SOL03430
C  2) SHOULD THE INCRIMENTAL DISPLACEMENT BE RECALCULATED?              SOL03440
C     IF SO; KSAME=KSAME+    (A VALUE DEPENDING ON ELEMENT TYPE)        SOL03450
C  3) THE INCRIMENTAL ELEMENT FORCES                                    SOL03460
C  4) THE TOTAL ELEMENT FORCES- STORED IN FUB                           SOL03470
C                                                                       SOL03480
      LSTTYP=0                                                          SOL03490
      PRINT=.FALSE.                                                     SOL03500
      HEAD =.FALSE.                                                     SOL03510
      NEWK=.FALSE.                                                      SOL03520
      DO 150 IELNO=1,NELMT                                              SOL03530
        IOPT=4                                                          SOL03540
        CALL ELELIB                                                     SOL03550
     &  (IOPT  ,LSTTYP ,PRINT,IREL ,HEAD  ,NAME ,EESE,EPSE,DAMAGE,      SOL03560
     &   DUCFAG,                                                        SOL03570
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)    SOL03580
      IOPT=IOPT                                                         SOL03590
  150 CONTINUE                                                          SOL03600
C                                                                       SOL03610
C============================                                          SOL03620
C= REPEAT STEP IF NESSARY ===                                         SOL03630
C============================                                          SOL03640
C  IF AN ELEMENT STIFFNESS CHANGE DICTATED THAT THE STEP BE REANALIZED, SOL03650
C     REFORMULATE THE STIFFNESS AND REANALIZE                           SOL03660
C     THE VARIABLE REPEAT IS SET UP TO SUPRESS THE REANALIZING OF A STEPSOL03670
C     CURENTLY, EACH STEP MAY ONLY BE REANALIZED ONCE, THIS PREVENTS    SOL03680
C     AN ENDLESS LOOP.                                                  SOL03690
C                                                                       SOL03700
      IF (KSAME.NE.0 .AND. REPEAT) THEN                                 SOL03710
          REPEAT=.FALSE.                                                SOL03720
          GO TO 100                                                     SOL03730
      ENDIF                                                             SOL03740
C                                                                       SOL03750
C=========================                                            SOL03760
C= SOLVE FOR REACTIONS ===                                            SOL03770
C=========================                                            SOL03780
C                                                                       SOL03790
      IOPT=2                                                            SOL03800
      CALL REACTN(2,BUG3,NNODE,L1,L4,L5,L6,DLOAD,DDISP,                 SOL03810
     &  MD,STIFF,IMD,NDOF,1,1.0,                                        SOL03820
     &  MAXNOD,IDOF,COORD,ID,TITLE,STEPID,.FALSE.,                      SOL03830
     &          NCOS,COSINE,JTCOS,JTFLG,SUMRC)                          SOL03840
C                                                                       SOL03850
C=========================================================            SOL03860
C= SOLVE FOR INCRIMENTAL GEOMETRIC STIFFNESS FORCE, DFG ===           SOL03870
C=========================================================            SOL03880
C                                                                       SOL03890
C------- INCRIMENTAL GEOMETRIC STIFFNESS FORCE                          SOL03900
      IF (KGFORM.EQ.2) THEN                                             SOL03910
        IF (BUG5) WRITE(6,*) '*----- INCRIMENTAL FORCE DUE TO KG '      SOL03920
        CALL TMPLT(BUG5,.TRUE.,IMD,L4,L3,L4,L3,L4,                      SOL03930
     &           MDKG,KG,DFG,DDISP)                                     SOL03940
CC      SCALE FORCE TO INLCUDE GROUND ACCELERATION ...                  SOL03950
      ENDIF                                                             SOL03960
C                                                                       SOL03970
C======================                                                SOL03980
C= PRINT STEP SIZE ===                                                SOL03990
C======================                                                SOL04000
      ISTEP=ISTEP                                                       SOL04010
      IPRNT=IPRNT                                                       SOL04020
      WRITE=(MOD(ISTEP,IPRNT).EQ.0)                                     SOL04030
      IF (WRITE) THEN                                                   SOL04040
        WRITE (6,245) TITLE(1),TITLE(2),STEPID,FACTOR,AFACT,DF,FACTL,   SOL04050
     &            PMAX,DMAX,FNORM                                       SOL04060
  245   FORMAT ('1 STRUCTURE....: ',A,/                                 SOL04070
     &        ' SOLUTION.....: ',A,//,2X,A/                             SOL04080
     &  '    TARGET FACTOR...:',1P,G12.4,                               SOL04090
     &  ' FACTOR....:',1P,G12.4,'  INC. FACTOR....:',G12.4,             SOL04100
     &  ' PREV. FACTOR.:',G12.4/                                        SOL04110
     &  '    PMAX.....:',1P,G12.4,                                      SOL04120
     &  ' DMAX.....:',G12.4,                                            SOL04130
     &  ' NORM OF UNBALANCED FORCE:',G12.4)                             SOL04140
        LINE=6                                                          SOL04150
      ENDIF                                                             SOL04160
C                                                                       SOL04170
C=======================================                               SOL04180
C= CALC AND ENERGIES AND DUCTILITIES ===                              SOL04190
C=======================================                               SOL04200
C                                                                       SOL04210
      IF (WRITE)  LINE=LINE+9                                           SOL04220
C---------- PULL JOINT DISPL FORCES OUT OF LOAD MATRIX ...              SOL04230
      IF (DSPFLG) THEN                                                  SOL04240
        DO 137 I=L1,L4                                                  SOL04250
          LOAD(I)  = LOAD(I)  - RLOAD(I)                                SOL04260
  137     DLOAD(I) = DLOAD(I) - DRLOA(I)                                SOL04270
      ENDIF                                                             SOL04280
C                                                                       SOL04290
      CALL ENERGY                                                       SOL04300
     & (L1,L2,L3,L4,L5,L6,NDOF,MAXNOD,NCOS,BUG11,.FALSE.,              SOL04310
     &  IMD,IMDMS,IMDKG,MD,MDMS,MDKG,.FALSE.,WRITE,ABORT,               SOL04320
     &  STIFF,KG,MASS,                                                  SOL04330
     &  EIE,ESE,PSE,EKE,EDD,                                            SOL04340
     &  ALPHA,BETA,KGFORM,NELMT,NNODE,DT,SUMUB,                         SOL04350
     &   SUMRC,DSUMRC, SUMFD,DSUMFD,FDAMP,DFDAMP, FI,                  SOL04360
     &  GA,GV,GD,GAT,GVT,GDT,   WORK,WORK(L6+1),ZERO,ZERO,              SOL04370
     &  DACC,DVEL,DDISP,ACC,VEL,DISP, LOAD,DLOAD,                       SOL04380
     &  ID      , IDOF    , CNST    , JTFLG    , COORD    ,            SOL04390
     &  COSINE  , JTCOS    )                                           SOL04400
      IF (ABORT) RETURN                                                 SOL04410
C                                                                       SOL04420
C=======================================================              SOL04430
C= GET TOTAL GLOBAL LOADS AND DISPL                    ==             SOL04440
C=======================================================              SOL04450
      DO 239 I=L1,L4                                                    SOL04460
        LOAD(I)=LOAD(I)+DLOAD(I) + RLOAD(I)                             SOL04470
        RLOAD(I) = RLOAD(I) + DRLOA(I)                                  SOL04480
        DISP(I)=DISP(I)+DDISP(I)                                        SOL04490
        FG(I)  =FG(I)  +DFG(I)                                          SOL04500
  239 CONTINUE                                                          SOL04510
      DO 240 I=L5,L6                                                    SOL04520
        LOAD(I)=LOAD(I)+DLOAD(I)                                        SOL04530
        DISP(I)=DISP(I)+DDISP(I)                                        SOL04540
        FG(I)  =FG(I)  +DFG(I)                                          SOL04550
  240 CONTINUE                                                          SOL04560
C                                                                       SOL04570
C..... GET MAXIMUN LOADS AND DISPL                                      SOL04580
      DO 1007 I=L1,L6                                                   SOL04590
        IF (ABS(DISP(I)).GT.ABS(DSPMAX(I))) DSPMAX(I)=DISP(I)           SOL04600
        IF (ABS(LOAD(I)).GT.ABS(RCTMAX(I))) RCTMAX(I)=LOAD(I)           SOL04610
 1007   CONTINUE                                                        SOL04620
C                                                                       SOL04630
C=====================================                                SOL04640
C= PRINT TOTAL GLOBAL DISPLACEMENTS ===                               SOL04650
C=====================================                                SOL04660
      IF (WRITE) THEN                                                   SOL04670
        IF (IPDISP+LINE.LE.60 ) THEN                                    SOL04680
          LINE=IPDISP+LINE                                              SOL04690
        ELSE                                                            SOL04700
          LINE=IPDISP+4                                                 SOL04710
          WRITE (6,250) TITLE(1),TITLE(2),     ISTEP,STEPID             SOL04720
        ENDIF                                                           SOL04730
C                                                                       SOL04740
  250 FORMAT ('1 STRUCTURE....: ',A,/                                   SOL04750
     &        ' SOLUTION.....: ',A,//                                   SOL04760
     &        2X,  ' LOADING #',I5,5X,A)                                SOL04770
        CALL JOIDSP(1,MAXNOD,NDOF,NNODE,1,L5,L6,ID,                     SOL04780
     &              IDOF,CNST,DISP,                                     SOL04790
     &              TITLE,'DISPLACEMENTS',STEPID,.FALSE.,               SOL04800
     &              NCOS,COSINE,JTCOS,JTFLG)                            SOL04810
      ENDIF                                                             SOL04820
C                                                                       SOL04830
C                                                                       SOL04840
C=====================================                                SOL04850
C= PRINT TOTAL GLOBAL REACTIONS ===                                   SOL04860
C=====================================                                SOL04870
      IF (WRITE) THEN                                                   SOL04880
        IF (IPRCTN+LINE.LE.60 ) THEN                                    SOL04890
          LINE=IPRCTN+LINE+4                                            SOL04900
        ELSE                                                            SOL04910
          LINE=IPRCTN+8                                                 SOL04920
          WRITE (6,250) TITLE(1),TITLE(2),     ISTEP,STEPID             SOL04930
        ENDIF                                                           SOL04940
      ENDIF                                                             SOL04950
C                                                                       SOL04960
      CALL REACTN(3,WRITE,NNODE,L1,L4,L5,L6,LOAD,DISP,                  SOL04970
     &  MD,STIFF,IMD,NDOF,1,1.0,                                        SOL04980
     &  MAXNOD,IDOF,COORD,ID,TITLE,STEPID,.FALSE.,                      SOL04990
     &  NCOS,COSINE,JTCOS,JTFLG,SUMRC)                                  SOL05000
      SUMFOR=SQRT(SUMRC(1)**2+SUMRC(2)**2+SUMRC(3)**2)                  SOL05010
      SUMMOM=SQRT(SUMRC(4)**2+SUMRC(5)**2+SUMRC(6)**2)                  SOL05020
      SUMMFO=MAX(SUMFOR,SUMMFO)                                         SOL05030
      SUMMMO=MAX(SUMMOM,SUMMMO)                                         SOL05040
      IF (WRITE) WRITE (6,313) SUMFOR,SUMMOM                            SOL05050
  313 FORMAT (/4X,'RESULTANT OF REACTIONS,    FORCE=',1P,G12.4,         SOL05060
     &                                      '  MOMENT=',G12.4)         SOL05070
C                                                                       SOL05080
      DO 1008 I=1,6                                                     SOL05090
        IF (ABS(SUMRC(I)).GT.ABS(SUMRCM(I))) SUMRCM(I)=SUMRC(I)         SOL05100
 1008 CONTINUE                                                          SOL05110
C                                                                       SOL05120
C=============================================                         SOL05130
C= CALCULATE AND PRINT TOTAL MEMBER FORCES ===                        SOL05140
C=============================================                         SOL05150
      IOPT=5                                                            SOL05160
      LSTTYP=0                                                          SOL05170
      HEAD =.FALSE.                                                     SOL05180
      IF (WRITE) THEN                                                   SOL05190
          PRINT=.TRUE.                                                  SOL05200
          IF (IPELEM+LINE.LE.60 ) THEN                                  SOL05210
            LINE=IPELEM+LINE                                            SOL05220
          ELSE                                                          SOL05230
            LINE=IPELEM+4                                               SOL05240
            WRITE (6,250) TITLE(1),TITLE(2),     ISTEP,STEPID           SOL05250
          ENDIF                                                         SOL05260
      ELSE                                                              SOL05270
          PRINT=.FALSE.                                                 SOL05280
      ENDIF                                                             SOL05290
      DO 300 IELNO=1,NELMT                                              SOL05300
  300   CALL ELELIB                                                     SOL05310
     &  (IOPT  ,LSTTYP ,PRINT,IREL ,HEAD  ,NAME ,EESE,EPSE,DAMAGE,      SOL05320
     &   DUCFAG,                                                        SOL05330
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)    SOL05340
C                                                                       SOL05350
C=============================================                         SOL05360
C= PRINT MEMBER DUCTILITY AND EXCURSION RATIOS ===                    SOL05370
C=============================================                         SOL05380
      IOPT=11                                                           SOL05390
      LSTTYP=0                                                          SOL05400
      HEAD =.FALSE.                                                     SOL05410
      IF (WRITE) THEN                                                   SOL05420
          PRINT=.TRUE.                                                  SOL05430
          IF (IPELEM+LINE.LE.60 ) THEN                                  SOL05440
            LINE=IPELEM+LINE                                            SOL05450
          ELSE                                                          SOL05460
            LINE=IPELEM+4                                               SOL05470
            WRITE (6,250) TITLE(1),TITLE(2),     ISTEP,STEPID           SOL05480
          ENDIF                                                         SOL05490
      ELSE                                                              SOL05500
          PRINT=.FALSE.                                                 SOL05510
      ENDIF                                                             SOL05520
      DO 310 IELNO=1,NELMT                                              SOL05530
  310   CALL ELELIB                                                     SOL05540
     &  (IOPT  ,LSTTYP ,PRINT,IREL ,HEAD  ,NAME ,EESE,EPSE,DAMAGE,      SOL05550
     &   DUCFAG,                                                        SOL05560
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)    SOL05570
C                                                                       SOL05580
C                                                                       SOL05590
C=================================                                     SOL05600
C= WRITE DATA TO OUTPUT FILES  ===                                    SOL05610
C=================================                                     SOL05620
      IF (IWRITE.GT.0) THEN                                             SOL05630
        IF (MOD(ISTEP,INWRITE).EQ.0) THEN                               SOL05640
          IOPT=2                                                        SOL05650
          CALL DMPDAT(IOPT,IWRITE,TO,DT)                                SOL05660
        ENDIF                                                           SOL05670
      ENDIF                                                             SOL05680
C                                                                       SOL05690
C=====================================================================  SOL05700
C= EITHER GET AN NEW FACTOR, OR KEEP WORKING ON THE CURRENT ONE ==     SOL05710
C=====================================================================  SOL05720
      FACTL=FACTL+DF                                                    SOL05730
      IF (NEWFAC) GO TO 1000                                            SOL05740
      GO TO 1100                                                        SOL05750
C                                                                       SOL05760
C=========================                                            SOL05770
C= CLOSE OUTPUT FILES  ===                                            SOL05780
C=========================                                            SOL05790
 5000 IF (IWRITE.GT.0) THEN                                             SOL05800
        IOPT=3                                                          SOL05810
```

```
         CALL DMPDAT(IOPT,IWRITE,TO,DT)
      ENDIF
C
C-----------------------------------
C= PRINT MAXIMUM  DISPLACEMENTS ===
C-----------------------------------
      LINE=IPDISP+4
      WRITE (6,250) TITLE(1),TITLE(2),0,'MAXIMUM DISPLACEMENTS'
      CALL JOIDSP(2,MAXNOD,NDOF,NNODE,1,L5,L6,ID,
     &            IDOF,CNST,DSPMAX,
     &            TITLE,'DISPLACEMENTS','MAXIMUM VALUES',.FALSE.,
     &            NCOS,COSINE,JTCOS,JTFLG)
C
C-----------------------------------
C= PRINT MAXIMUM GLOBAL REACTIONS ===
C-----------------------------------
      WRITE (6,250) TITLE(1),TITLE(2),0,'MAXIMUM REACTIONS'
        CALL REACTN(4,.TRUE.,NNODE,L1,L4,L5,L6,RCTMAX,DSPMAX,
     &       MD,STIFF,IMD,NDOF,1,1.0,
     &       MAXNOD,IDOF,COORD,ID,TITLE,'MAXIMUM VALUES',.FALSE.,
     &       NCOS,COSINE,JTCOS,JTFLG,SUMRCM)
      WRITE (6,5111) SUMMFO,SUMMMO
 5111 FORMAT (/4X,'MAXIMUM RESULTANT OF REACTIONS,   FORCE=',1P,G12.4,
     &                                              MOMENT=',G12.4)
C
C-----------------------------------
C= PRINT MAXIMUM MEMBER FORCES ===
C-----------------------------------
      LSTTYP=0
      HEAD =.FALSE.
      PRINT=.TRUE.
      WRITE (6,250) TITLE(1),TITLE(2),0,'PEAK ELEMENT FORCES'
      DO 5300 IELNO=1,NELMT
 5300   CALL ELELIB
     &  (12     ,LSTTYP ,PRINT,IREL ,HEAD  ,NAME ,EESE,EPSE,DAMAGE,
     &   DUCFAG,
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)
C
C-----------------------------------------------
C= PRINT DUCTILITIES AND EXCURSION RATIOS ===
C-----------------------------------------------
      LSTTYP=0
      HEAD =.FALSE.
      PRINT=.TRUE.
      WRITE (6,250) TITLE(1),TITLE(2),0,
     &    'PEAK DUCTILITIES AND EXCURSION RATIO''S'
      DO 5310 IELNO=1,NELMT
 5310   CALL ELELIB
     &  (11     ,LSTTYP ,PRINT,IREL ,HEAD  ,NAME ,EESE,EPSE,DAMAGE,
     &   DUCFAG,
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)
C
C-----------------------------
C= PRINT DAMAGE INDICIES  ===
C-----------------------------
      LSTTYP=0
      HEAD =.FALSE.
      PRINT=.TRUE.
      WRITE (6,250) TITLE(1),TITLE(2),0,
     &    'DAMAGE INDEX '
      DAMAGE=0
      DO 5320 IELNO=1,NELMT
        CALL ELELIB
     &  (14     ,LSTTYP ,PRINT,IREL ,HEAD  ,NAME ,EESE,EPSE,EDAM,
     &   DUCFAG,
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)
 5320   DAMAGE=DAMAGE+EDAM
      IF( (ESE+PSE) .EQ. 0 ) RETURN
      DAMAGE=DAMAGE/(ESE+PSE)
      WRITE (6,5330) DAMAGE
 5330 FORMAT (/10X,'STRUCTURE DAMAGE INDEX=',1P,G15.5)
      RETURN
C
      END
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP
C-------------------------------------------------------------
C         DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT
C         ENDBUG
C-------------------------------------------------------------
      SUBROUTINE SOL05
      LOGICAL PRINT,HEAD,BTEST,TEST,QUIT
      COMMON /RSA/ ICOMN, RSAF(12)
      REAL ICOMN
      DIMENSION RINPUT(100)
      CHARACTER*20 ICOM
      INCLUDE (ZCOMN)
C
C----------------
C= PRINT HEADER =
C----------------
      WRITE(6,10)TITLE(1),TITLE(2)
   10 FORMAT('1 STRUCTURE....: ',A,/
     &       ' SOLUTION.....: ',A,/)
      DO 555 I=1,20
  555   WRITE(6,*)
      WRITE(6,557)
  557   FORMAT(40X,'** RESPONSE SPECTRUM ANALYSIS **',///
     &           40X,'   NOTE: MODAL COMBINATION APPROACH CAN BE SRSS ',
     &               'OR CQC')
C
C----------------
C= INPUT DATA ===
C----------------
      READ(5,*)ICOM,NMODE,DAMP,NA,NAP,IPRT
      READ(5,*)NSTEP,MEIG
C NOTE: NSTEP =1 : CALCULATE EIGENPAIRS ,STORE EIGENPAIRS IN DISK MEIG
C       NSTEP =2 : READ EIGENPAIR FROM 'MEIG' AND DO SPECTRUM ANALYSIS
C       NSTEP =3 : CALCULATE EIGENPAIRS AND PERFORM SPECTRUM ANALYSIS,
C                  EIGENPAIRS ARE NOT STORED IN UNIT MEIG
C
      WRITE(6,10)TITLE(1),TITLE(2)
      IF( TEST(ICOM,'SRSS' ,.FALSE.)) THEN
        ICOMN=1
        WRITE(6,11)'SRSS MODAL COMBINATION METHOD',
     &             '============================'
      ELSE IF( TEST(ICOM,'CQC' ,.FALSE.)) THEN
        ICOMN=2
        WRITE(6,11)'CQC MODAL COMBINATION METHOD',
     &             '=========================='
      ELSE
        WRITE(6,*)'INVALID MODAL COMBINATION METHOD'
        WRITE(6,11) ICOM,'====================='
        RETURN
      ENDIF
C
      WRITE(6,12)NMODE,DAMP,NA,NAP
   11 FORMAT(1X,'SOLUTION #5, ',A,/
     &       1X,'============',A,/)
   12 FORMAT(1X,' NUMBER OF MODES CONSIDERED .............',OP,I4,/
     &       1X,' DAMPING RATIO FOR ALL MODES ...........',OP,G12.3,/
     &       1X,' NO. OF SEISMIC COMPONENTS CONSIDERED ..',OP,I4,/
     &       1X,' NO. OF INPUT POINTS FOR SPECTRUM ......',OP,I4)
C
      IF(NSTEP .EQ. 1 ) THEN
      WRITE(6,13)NSTEP,MEIG
   13 FORMAT(1X,' NSTEP ..............................',OP,I4,/
     &       1X,' EIGEN-SOLUTION IS STORE IN DISK UNIT ..',OP,I4)
      ELSE IF(NSTEP .EQ. 2 ) THEN
      WRITE(6,14)NSTEP,MEIG
   14 FORMAT(1X,' NSTEP ..............................',OP,I4,/
     &       1X,' EIGEN-SOLUTION IS READ FROM DISK UNIT ..',OP,I4)
      ENDIF
C                                                                      SOL00690
C     ---- SET UP CONSTANTS --------------                             SOL00700
      L1=1                                                             SOL00710
      L2=NCOND                                                         SOL00720
      L3=L2+1                                                          SOL00730
      L4=NCOND +NFREE                                                  SOL00740
      L5=L4+1                                                          SOL00750
      L6=NDOF                                                          SOL00760
      ELSTIC=.TRUE.                                                    SOL00770
C                                                                      SOL00780
C     ---- INITIALIZE STORAGE                                          SOL00790
C     Z(IZEVAL) = LOCATION OF THE EIGENVALUES                          SOL00800
C     Z(IZEVEC) = LOCATION OF THE EIGENVECTORS                         SOL00810
C     Z(IZA   ) = LOCATION OF THE SYMMETRIC STIFFNESS                  SOL00820
C     Z(IZB   ) = LOCATION OF THE SYMMETRIC MASS                       SOL00830
C     Z(IZLOAD) = LOCATION OF THE TOTAL LOAD MATRIX                    SOL00840
C     Z(IZDISP) = LOCATION OF THE TOTAL DISPLACEMENT MATRIX            SOL00850
C                                                                      SOL00860
      IZEVAL = I%                                                      SOL00870
      I%     = I%+NFREE                                                SOL00880
      IZEVEC = I%                                                      SOL00890
      I%     = IZ+NFREE*NFREE                                          SOL00900
      IZA    = I%                                                      SOL00910
      I%     = IZ+NFREE*NFREE                                          SOL00920
      IZB    = I%                                                      SOL00930
      I%     = IZ+NFREE*NFREE                                          SOL00940
      IF(IZLOAD .EQ. 0 ) THEN                                          SOL00950
         IZLOAD=I%                                                     SOL00960
         I%=I% + NDOF                                                  SOL00970
         CALL CKSTOR(0,0)                                              SOL00980
         DO 1 I=1,NDOF                                                 SOL00990
    1      Z(I+IZLOAD-1)=0                                             SOL01000
      ENDIF                                                            SOL01010
C                                                                      SOL01020
      IF(IZDISP .EQ. 0 ) THEN                                          SOL01030
         IZDISP=I%                                                     SOL01040
         I%=I% + NDOF                                                  SOL01050
         CALL CKSTOR(0,0)                                              SOL01060
         DO 2 I=1,NDOF                                                 SOL01070
    2      Z(I+IZDISP-1)=0                                             SOL01080
      ENDIF                                                            SOL01090
C                                                                      SOL01100
      IF(I%FUB .EQ. 0 ) THEN                                           SOL01110
         IZFUB=I%                                                      SOL01120
         I%=I% + NDOF                                                  SOL01130
         CALL CKSTOR(0,0)                                              SOL01140
         DO 3 I=1,NDOF                                                 SOL01150
    3      Z(I+IZFUB-1)=0                                              SOL01160
      ENDIF                                                            SOL01170
C                                                                      SOL01180
C--------DEFINE STORAGE FOR THE MODAL COMBINATION                      SOL01190
      IZRA=I%                                                          SOL01200
      I%     = I%+NMODE*3                                              SOL01210
      IZRD=I%                                                          SOL01220
      I%     = I%+NMODE*3                                              SOL01230
      IZFLU=I%                                                         SOL01240
      I%     = I%+NFREE*3                                              SOL01250
      IZCMC=I%                                                         SOL01260
      I%     = I%+NMODE*NMODE                                          SOL01270
      IZGM =I%                                                         SOL01280
      I%     = I%+NMODE                                                SOL01290
      IZELF=I%                                                         SOL01300
      I%     = I%+NMODE*3                                              SOL01310
      IZCQC=I%                                                         SOL01320
      I%     = I%+L4*NMODE                                             SOL01330
      IZDLD=I%                                                         SOL01340
      I%     = I%+NDOF*NA                                              SOL01350
      IZELEF=I%                                                        SOL01360
      I%     = I%+NELMT*12*(NMODE+NA)                                  SOL01370
      IZRCT=I%                                                         SOL01380
      I%     = I%+6*(NMODE+NA)                                         SOL01390
      IZNAP=I%                                                         SOL01400
      I%     = I%+2*NAP                                                SOL01410
      IZTMP2=I%                                                        SOL01420
      I%     = I%+NFREE                                                SOL01430
      IZBEF=IZ                                                         SOL01440
      I%     = IZ+NFREE*NA                                             SOL01450
      IZCQ2=I%                                                         SOL01460
      IZ     = I%+NFREE*NMODE                                          SOL01470
C                                                                      SOL01480
       IZTMP= I%                                                       SOL01490
       IZ   = I%+NDOF                                                  SOL01500
       IW   =IZ                                                        SOL01510
C                                                                      SOL01520
      CALL CKSTOR(NDOF,IW)                                             SOL01530
C                                                                      SOL01540
      IMD  =N%(I%MD  +NDOF)                                            SOL01550
      IMDMS=N%(I%MDMS+NDOF)                                            SOL01560
      IMDKG=N%(I%MDKG+NDOF)                                            SOL01570
C                                                                      SOL01580
C---------- FORM MASS MATRIX -----                                     SOL01590
      DO 400 I=1,IMDMS                                                 SOL01600
  400    Z(I+IZMASS-1)=0.                                              SOL01610
C                                                                      SOL01620
      CALL MFORM(2,INMASS,NMASS,Z(IZMASS),NZ(IZMDMS),MD,NDOF,IMDMS,    SOL01630
     &           TITLE,NNODE,MAXNOD,NCOS,BTEST(IBUG,5),IBUG,           SOL01640
     &           NZ(I%ID),Z(I%COS),NZ(I%IDOF),Z(IZCNST),NZ(IZJPLG),    SOL01650
     &           NZ(IZJCOS),Z(IZMDAT),Z(IW),NZ(IW+21),Z(IW+27),        SOL01660
     &            Z(IW+ 63),Z(IW+ 99))                                 SOL01670
C                                                                      SOL01680
      CALL SOL05A                                                      SOL01690
     & (L1,L2,L3,L4,L5,L6,IMD,IMDMS,IMDKG,      KGCOND,MCOND,ABORT,   SOL01700
     &  Z(IZKGDT),NZ(IZKGDT+1),NZ(IZKGDT+2),NZ(IZKGDT+3),              SOL01710
     &  IZDISP,IZLOAD,MSTOR,                                           SOL01720
     &  NDOF   , NCOND   , NFREE   , TITLE   , STEPID  ,               SOL01730
     &  Z(IZSTIF), Z(IZMASS), Z(I%KG  ), Z(IZA   ), Z(IZB   ),         SOL01740
     & NZ(IZMD  ),NZ(IZMDMS),NZ(I%MDKG), Z(IZDISP), Z(IZTMP ),         SOL01750
     &  NNODE  , NCOS    , NELMT   , MAXNOD  ,IBUG   ,                 SOL01760
     &  NA,NAP   , NMODE   , DAMP    , IPRT    ,NSTEP,MEIG,           SOL01770
     & NZ(IZID  ),NZ(IZIDOF), Z(IZCNST),NZ(IZJPLG), Z(IZCORD),         SOL01780
     &  Z(IZCOS ),NZ(IZJCOS), Z(IZEVAL), Z(IZEVEC),                    SOL01790
     &  Z(IZRA  ),                                                     SOL01800
     &  Z(IZRD  ),Z(IZFLU ), Z(IZCMC ), Z(IZGM  ),                     SOL01810
     &  Z(IZELF ),Z(IZLOAD), Z(IZCQC ), Z(IZDLD ),                     SOL01820
     &  Z(IZELEF),Z(IZRCT ), Z(IZNAP ),                                SOL01830
     &  Z(IZTMP2),Z(IZBEF ), Z(IZCQ2 ), Z(IW    ))                     SOL01840
C                                                                      SOL01850
      RETURN                                                           SOL01860
      END                                                              SOL01870
C@PROCESS SDUMP OPT(0) GOSTMT XREF MAP                                 SOL01880
C=======================================================================SOL00010
C           DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                   SOL00020
C           END DEBUG                                                  SOL00030
C=======================================================================SOL00040
      SUBROUTINE SOL05A                                                SOL00050
     & (L1,L2,L3,L4,L5,L6,IMD,IMDMS,IMDKG,      KGCOND,MCOND,ABORT,   SOL00060
     &  KGDATA , KGLOAD , KGTYPE , KGFORM,                             SOL00070
     &  IZDISP , IZLOAD , MSTOR  ,                                     SOL00080
     &  NDOF   , NCOND   , NFREE   , TITLE   , STEPID  ,               SOL00090
     &  STIFF  , MASS    , KG      , A       , B       ,               SOL00100
     &  MD     , MDMS    , MDKG    , DISP    , TMP     ,               SOL00110
     &  NNODE  , NCOS    , NELMT   , MAXNOD  , IBUG    ,               SOL00120
     &  NA,NAP , NMODE   , DAMP    , IPRT    , NSTEP   , MEIG ,        SOL00130
     &  ID     , IDOF    , CNST    , JTFLG   , COORD   ,               SOL00140
     &  COSINE , JTCOS   , EVAL    , EVEC    ,                         SOL00150
     &  RA     ,                                                       SOL00160
     &  RD     , FLU     , CMC     , GM      ,                         SOL00170
     &  ELF    , LOAD    , CQCM    , DLOAD   ,                         SOL00180
     &  ELEF   , RCT     , RSNP    ,                                   SOL00190
     &  TMP2   , BEF     , CQCM2   , WORK    )                         SOL00200
                                                                       SOL00210
                                                                       SOL00220
```

```
      COMMON/RSA/ICOMN,RSAF(12)                                         SOL00230
      REAL ICOMN                                                        SOL00240
C-- COMMON BLOCK FOR IMSL SUBROUTINE DGVCSP                            SOL00250
C     COMMON/WORKSP/RWKSP                                               SOL00260
C     REAL RWKSP(100000)                                                SOL00270
      LOGICAL TEST,QUIT,BTEST,MCOND,KGCOND,MCOND1,KCOND1,ABORT,PRINT    SOL00280
      LOGICAL HEAD, AXIAL                                               SOL00290
      REAL LOAD, MASS, KG, KGDATA                                       SOL00300
      DIMENSION STIFF(IMD),MASS(IMDMS),KG(IMDKG),RINPUT(100)            SOL00310
      DIMENSION A(NFREE,NFREE),B(NFREE,NFREE)                           SOL00320
      DIMENSION MD(L6+1),MDMS(L6+1),MDKG(L6+1),DISP(L6),TMP(L6)         SOL00330
      DIMENSION EVAL(NFREE),EVEC(NFREE,NFREE),SUMRCT(6)                 SOL00340
      DIMENSION COORD(MAXNOD,3),ID(MAXNOD),                             SOL00350
     &          IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),CNST(MAXNOD,6),       SOL00360
     &          JTFLG(MAXNOD),JTCOS(MAXNOD)                            SOL00370
      DIMENSION G(6),D(6),TMP2(NFREE),EEP(NFREE,NA),CQCM2(NFREE,NMODE)  SOL00380
      DIMENSION V1(3),V2(3),V3(3)                                       SOL00390
      DIMENSION RA(NMODE,3),LOAD(L6)                                    SOL00400
      DIMENSION RD(NMODE,3),DLOAD(L6,NA)                                SOL00410
      DIMENSION FLU(NFREE,3),CMC(NMODE,NMODE)                           SOL00420
      DIMENSION GM(NMODE),ELF(NMODE,3),WORK(L4),CQCM(L4,NMODE)          SOL00430
      DIMENSION ELEF(NELMT,12,NMODE+NA),RCT(6,NMODE+NA),RSNP(2,NAP)     SOL00440
      DIMENSION WK2(80, 80)                                             SOL00450
C NOTE: WK2 DIMENSION SHOULD BE GREATER THAN NFREE*NFREE                SOL00460
C                                                                       SOL00470
      CHARACTER*1   NAME                                                SOL00480
      CHARACTER*40  OPTION                                              SOL00490
      CHARACTER(*) TITLE(2),STEPID                                      SOL00500
C                                                                       SOL00510
C-------------------------                                              SOL00520
C= FORMULATE STIFFNESS ====                                            SOL00530
C-------------------------                                              SOL00540
      I1=1                                                              SOL00550
      CALL FORM(I1)                                                     SOL00560
        IF (BTEST(IBUG,5)) CALL LMATRI(STIFF,MD,NDOF ,IMD,              SOL00570
     &    'GLOBAL STIFFNESS MATRIX')                                    SOL00580
C                                                                       SOL00590
C--------------------------------                                       SOL00600
C= CONDENSE OUT NON-FREE DOF ===                                       SOL00610
C--------------------------------                                       SOL00620
C... IF MCOND IS TRUE THE MASS MATRIX IS ALSO REDUCED BY GUYAN REDUCTIONSOL00630
      IF (NCOND.GT.0) THEN                                              SOL00640
          I1=1                                                          SOL00650
          I1A=1                                                         SOL00660
          KCOND1 =.FALSE.                                               SOL00670
          MCOND1 =MCOND                                                 SOL00680
C             SET UP ZERO LOAD VECTOR                                   SOL00690
          DO 28 I=1,NDOF                                                SOL00700
  28          TMP(I)=0                                                  SOL00710
          CALL MSOLL(1,BTEST(IBUG,6),L4,IMD,NDOF,1,L1,L2,               SOL00720
     &                 MD,STIFF,TMP,WORK,                               SOL00730
     &                 MCOND1,MASS,MDMS,IMDMS,                          SOL00740
     &                 KCOND1,KG  ,MDKG,IMDKG,ABORT)                    SOL00750
          IF (ABORT) RETURN                                             SOL00760
      ENDIF                                                             SOL00770
C                                                                       SOL00780
C-----------------------------------------                              SOL00790
C= TRANSFER STIFFNESS INTO A MATRIX ===                                 SOL00800
C-----------------------------------------                              SOL00810
C  NOTE: IMSL ROUTINE DGVCSP IS USED TO SOLVE FOR EIGENVALUES,          SOL00820
C        THIS ROUTINE USES A 'FULL SYMMETRIC MATRIX'.                   SOL00830
C                                                                       SOL00840
      QUIT=.FALSE.                                                      SOL00850
      DO 600 J=L3,L4,+1                                                 SOL00860
        DO 600 I=J,L3,-1                                                SOL00870
          IJ=MD(J)+J-I                                                  SOL00880
          IF(IJ .LT. MD(J+1) .AND. IJ .GE. MD(J) ) THEN                 SOL00890
              A(I-L3+1,J-L3+1)=STIFF(IJ)                                SOL00900
              A(J-L3+1,I-L3+1)=A(I-L3+1,J-L3+1)                         SOL00910
          ELSE IF( IJ .GE. MD(J+1) ) THEN                               SOL00920
              A(I-L3+1,J-L3+1)=0                                        SOL00930
              A(J-L3+1,I-L3+1)=0                                        SOL00940
          ENDIF                                                         SOL00950
C                                                                       SOL00960
          IF (BTEST(IBUG,5)) WRITE (6,29) I,J,A(I-L3+1,J-L3+1)          SOL00970
   29     FORMAT (' I=',I5,' J=',I5,' A(I-L3+1,J-L3+1)=',1P,G15.5)      SOL00980
C                                                                       SOL00990
C--------- CHECK FOR ZERO'S ON MAIN DIAGONAL                            SOL01000
          IF (I.EQ.J .AND. A(I-L3+1,J-L3+1).LE.0 ) THEN                 SOL01010
              QUIT=.TRUE.                                               SOL01020
              WRITE (6,32) I,A(I-L3+1,J-L3+1)                           SOL01030
          ENDIF                                                         SOL01040
C                                                                       SOL01050
  600  CONTINUE                                                         SOL01060
   32 FORMAT (5X,'DOF #',I5,' IS .LE. 0 IN THE STIFFNESS MATRIX...',    SOL01070
     &        5X,'K(I,I)=',1P,G12.4,'  SOLN IS ABORTED')                SOL01080
C                                                                       SOL01090
C---------------------------------                                      SOL01100
C= TRANSFER MASS INTO B MATRIX ===                                      SOL01110
C---------------------------------                                      SOL01120
      DO 700 J=L3,L4,+1                                                 SOL01130
        DO 700 I=J,L3,-1                                                SOL01140
          IJ=MDMS(J)+J-I                                                SOL01150
          IF(IJ .LT. MDMS(J+1) .AND. IJ .GE. MDMS(J) ) THEN             SOL01160
              B(I-L3+1,J-L3+1)=MASS(IJ)                                 SOL01170
              B(J-L3+1,I-L3+1)=B(I-L3+1,J-L3+1)                         SOL01180
          ELSE IF( IJ .GE. MDMS(J+1) ) THEN                             SOL01190
              B(I-L3+1,J-L3+1)=0                                        SOL01200
              B(J-L3+1,I-L3+1)=0                                        SOL01210
          ENDIF                                                         SOL01220
          IF (BTEST(IBUG,5)) WRITE (6,39) I,J,B(I-L3+1,J-L3+1)          SOL01230
   39     FORMAT (' I=',I5,' J=',I5,' B(I-L3+1,J-L3+1)=', 1P,G15.5)     SOL01240
C                                                                       SOL01250
C--------- CHECK FOR ZERO'S ON MAIN DIAGONAL                            SOL01260
          IF (I.EQ.J .AND. B(I-L3+1,J-L3+1).LE.0 ) THEN                 SOL01270
              QUIT=.TRUE.                                               SOL01280
              WRITE (6,42) I,B(I-L3+1,J-L3+1)                           SOL01290
          ENDIF                                                         SOL01300
C                                                                       SOL01310
  700 CONTINUE                                                          SOL01320
   42 FORMAT (5X,'DOF #',I5,' IS .LE. 0 IN THE MASS MATRIX...',         SOL01330
     &        5X,'M(I,I)=',1P,G12.4,'  SOLN IS ABORTED')                SOL01340
C                                                                       SOL01350
C                                                                       SOL01360
C---------- RETURN IF A MAIN DIAGONAL IS LE ZERO                        SOL01370
      IF (QUIT) RETURN                                                  SOL01380
C                                                                       SOL01390
C-----------------------------------------                              SOL01400
C= SOLVE FOR EIGENVALUES AND EIGENVECTORS ===                           SOL01410
C-----------------------------------------                              SOL01420
C---------- IMSL SU_BROUTINE DGVCSP ---------------------------------- SOL01430
C  EIGENVALUES AND EIGENVECTORS OF A*X-LAMBDA*B*X=0                     SOL01440
C  A  = STIFFNESS - REAL, FULL SYMMETRIC MATRIX                         SOL01450
C  B  = MASS - REAL, FULL SYMMETRIC MATRIX                              SOL01460
C  N  = NUMBER OF DOF OF A AND B                                        SOL01470
C  PI = PERFORMANCE INDEX                                               SOL01480
C          PI=MK(1),    PI<1    - ROUTINE PERFORMED WELL                SOL01490
C                     1<PI<100  - ROUTINE PERFORMED SATISFACTORILY      SOL01500
C                   100<PI      - ROUTINE PERFORMED POORLY              SOL01510
C  EVAL= EIGENVALUES (LAMBDA), N BY 1                                   SOL01520
C  EVEC= EIGENVECTORS, N BY N                                           SOL01530
C  LDA = LEADING DIMENSION OF MATRIX A EXACTLY AS SPECFIED IN THE       SOL01540
C        DIMENSION STATEMENT IN THE CALLING PROGRAM.                    SOL01550
C  LDB = LEADING DIMENSION OF MATRIX B EXACTLY AS SPECFIED IN THE       SOL01560
C        DIMENSION STATEMENT IN THE CALLING PROGRAM.                    SOL01570
CLDEVEC= LEADING DIMENSION OF MATRIX EVEC EXACTLY AS SPECFIED IN THE    SOL01580
C        DIMENSION STATEMENT IN THE CALLING PROGRAM.                    SOL01590
C---------------------------------------------------------------------- SOL01600
C                                                                       SOL01610
      IF(NSTEP .EQ. 1 .OR. NSTEP .EQ. 3) THEN                           SOL01620
        LDEVEC=NFREE                                                    SOL01630
C       CALL IWKIN(100000)                                              SOL01640
C       CALL DGVCSP(NFREE,A,NFREE,B,NFREE,EVAL,EVEC,LDEVEC)             SOL01650
        CALL DG2CSP(NFREE,A,NFREE,B,NFREE,EVAL,EVEC,LDEVEC, A,WK2,WORK) SOL01660
C                                                                       SOL01670
C---------------------------------------------                          SOL01680
C= NORMALIZE MODE SHAPE TO MAX VALUE OF ONE ===                         SOL01690
C---------------------------------------------                          SOL01700
C                                                                       SOL01710
C NOTE: THE EIGENVECTOR IN IMSL DGVCSP HAS BEEN NORMALIZED              SOL01720
C                                                                       SOL01730
C-----WRITE EIGENVALUES AND EIGENVECTORS TO DISK MEIG                   SOL01740
        IF( NSTEP .EQ. 1 .OR. NSTEP .EQ. 3) THEN                        SOL01750
          WRITE(MEIG,175)(EVAL(I),I=1,IPRT)                             SOL01760
          DO 176 I=1,NFREE                                              SOL01770
  176     WRITE(MEIG,175)(EVEC(I,J),J=1,IPRT)                           SOL01780
  175   FORMAT(5(G14.5,1X))                                             SOL01790
        ENDIF                                                           SOL01800
C                                                                       SOL01810
        ELSE IF(NSTEP .EQ. 2 ) THEN                                     SOL01820
          READ(MEIG,175)(EVAL(I),I=1,IPRT)                              SOL01830
          DO 178 I=1,NFREE                                              SOL01840
  178     READ(MEIG,175)(EVEC(I,J),J=1,IPRT)                            SOL01850
        ENDIF                                                           SOL01860
C                                                                       SOL01870
C---------------------------------------                                SOL01880
C= PRINT EIGENVALUES AND EIGENVECTORS ===                               SOL01890
C---------------------------------------                                SOL01900
        IST =1                                                          SOL01910
  100   IEND=MIN(IST+6,IPRT)                                            SOL01920
C                                                                       SOL01930
        WRITE (6,110) (I             ,I=IST,IEND)                       SOL01940
        DO 125 I=IST,IEND                                               SOL01950
  125     EVAL(I)=SQRT( EVAL(I) )                                       SOL01960
        WRITE (6,126) (EVAL(I),I=IST,IEND)                              SOL01970
        DO 127 I=IST,IEND                                               SOL01980
  127     EVAL(I)= EVAL(I)  / 6.2831853                                 SOL01990
        WRITE (6,128) (EVAL(I),I=IST,IEND)                              SOL02000
        DO 129 I=IST,IEND                                               SOL02010
          IF (EVAL(I).EQ.0) EVAL(I)=1.E-15                              SOL02020
  129     EVAL(I)= 1.00 / EVAL(I)                                       SOL02030
        WRITE (6,130) (EVAL(I),I=IST,IEND)                              SOL02040
        WRITE (6,135)                                                   SOL02050
  110 FORMAT('          MODE:',7(I10,5X))                               SOL02060
  120 FORMAT('     EIGENVALUE:',1P,7(G14.5,1X) )                        SOL02070
  126 FORMAT(' FREQ (RAD/SEC):',1P,7(G14.5,1X) )                        SOL02080
  128 FORMAT(' FREQUENCY (HZ):',1P,7(G14.5,1X) )                        SOL02090
  130 FORMAT('   PERIOD (SEC):',1P,7(G14.5,1X) )                        SOL02100
  135 FORMAT('   EIGENVECTORS:')                                        SOL02110
C                                                                       SOL02120
C-------- EIGENVECTORS                                                 SOL02130
        DO 140 I=1,NFREE                                                SOL02140
          II=I+NCOND                                                    SOL02150
  140     WRITE (6,150) II,(EVEC(I,J),J=IST,IEND)                       SOL02160
  150   FORMAT('      DOF(',I6,')',1P,7(G14.5,1X) )                     SOL02170
C                                                                       SOL02180
        IF (IEND.LT.IPRT ) THEN                                         SOL02190
          WRITE (6,10) TITLE(1),TITLE(2)                                SOL02200
          IST=IEND+1                                                    SOL02210
          GO TO 100                                                     SOL02220
        ENDIF                                                           SOL02230
   10 FORMAT('1 STRUCTURE.....: ',A,/                                   SOL02240
     &       ' SOLUTION.....: ',A,//)                                   SOL02250
C                                                                       SOL02260
        IF(NSTEP .EQ. 1 ) THEN                                          SOL02270
          ABORT=.TRUE.                                                  SOL02280
          RETURN                                                        SOL02290
        ENDIF                                                           SOL02300
C                                                                       SOL02310
C-----ZERO GLOBAL BASE ACCELERATION AND DISPLACEMENT SPECTRUM VALUES    SOL02320
        DO 771 I=1,NMODE                                                SOL02330
          RA(I,1)=0                                                     SOL02340
          RA(I,2)=0                                                     SOL02350
          RA(I,3)=0                                                     SOL02360
          RD(I,1)=0                                                     SOL02370
          RD(I,2)=0                                                     SOL02380
  771     RD(I,3)=0                                                     SOL02390
C                                                                       SOL02400
C---------------------------------------------------------------------- SOL02410
C-- INPUT NUMBER OF SEISMIC COMPONENTS, SCALING FACTOR ,PRINT CONTROL = SOL02420
C---------------------------------------------------------------------- SOL02430
C                                                                       SOL02440
        READ(5,*)ASCALE,OPTION,PRINT                                    SOL02450
C                                                                       SOL02460
C-----INPUT DIRECTION VECTORS V1 AND V2                                 SOL02470
        READ(5,*)V1,V2                                                  SOL02480
C                                                                       SOL02490
C------- CALCULATE V3 = V1 X V2                                         SOL02500
        CALL CROSS(V3,V1,V2,0)                                          SOL02510
C                                                                       SOL02520
C--------CALCULATE V2 = V3 X V1, AND NORMALIZE ALL VECTORS...           SOL02530
        CALL CROSS(V2,V3,V1,1)                                          SOL02540
C---------------------------------                                      SOL02550
C-- INPUT RESPONSE SPECTRUM VALUES --                                   SOL02560
C---------------------------------                                      SOL02570
        WRITE(6,10)TITLE(1),TITLE(2)                                    SOL02580
        DO 30 I=1,NA                                                    SOL02590
          READ (5,*) IDIR                                               SOL02600
C                                                                       SOL02610
          IF (IDIR.GE.1 .AND. IDIR.LE.3) THEN                           SOL02620
            IF(TEST(OPTION,'ACC'..FALSE. )) THEN                        SOL02630
              WRITE (6,107) I                                           SOL02640
  107         FORMAT (///1X,'** ACCELERATION SPECTRUM VALUES :',        SOL02650
     &          'INPUTING TRANSLATIONAL #',I2,/)                        SOL02660
            ELSE IF(TEST(OPTION,'DISP',.FALSE. )) THEN                  SOL02670
              WRITE (6,1070) I                                          SOL02680
 1070         FORMAT (///1X,'** DISPLACEMENT SPECTRUM VALUES :',        SOL02690
     &          'INPUTING TRANSLATIONAL #',I2,/)                        SOL02700
            ENDIF                                                       SOL02710
          ELSE                                                          SOL02720
            WRITE (6,*) 'INVALID DIRECTION:',IDIR,' SET 0 < IDIR < 4'   SOL02730
            GO TO 30                                                    SOL02740
          ENDIF                                                         SOL02750
C                                                                       SOL02760
          IF (IDIR.EQ.1) THEN                                           SOL02770
            CX=V1(1)                                                    SOL02780
            CY=V1(2)                                                    SOL02790
            CZ=V1(3)                                                    SOL02800
            WRITE (6,92) V1                                             SOL02810
          ELSE IF (IDIR.EQ.2) THEN                                      SOL02820
            CX=V2(1)                                                    SOL02830
            CY=V2(2)                                                    SOL02840
            CZ=V2(3)                                                    SOL02850
            WRITE (6,92) V2                                             SOL02860
          ELSE IF (IDIR.EQ.3) THEN                                      SOL02870
            CX=V3(1)                                                    SOL02880
            CY=V3(2)                                                    SOL02890
            CZ=V3(3)                                                    SOL02900
            WRITE (6,92) V3                                             SOL02910
          ENDIF                                                         SOL02920
C                                                                       SOL02930
          DO 909 II=1,NAP                                               SOL02940
  909       READ(5,*)RSNP(1,II),RSNP(2,II)                             SOL02950
          WRITE(6,640)                                                  SOL02960
  640     FORMAT(30X,'PERIOD          ',10X,'SPECTRUM VALUE',/          SOL02970
     &      30X,'______________',10X,'_______________')                 SOL02980
          DO 908 JJ=1,NAP                                               SOL02990
  908       WRITE(6,907)RSNP(1,JJ),RSNP(2,JJ)                           SOL03000
  907     FORMAT(30X,G14.6,10X,G14.6)                                   SOL03010
C                                                                       SOL03020
          DO 95 II=1,NMODE                                              SOL03030
            TI=EVAL(II)                                                 SOL03040
            CALL INTPC(TI,TMP(II),NAP,RSNP)                             SOL03050
   95     CONTINUE                                                      SOL03060
```

```
C
        IF(TEST(OPTION,'ACC',.FALSE. ) ) THEN
          DO 20 J=1,NMODE
            ACC=TMP(J)*ASCALE
            DIS=TMP(J)*ASCALE/((6.2831853/EVAL(J))**2)
            RA(J,1)=RA(J,1) + ACC*CX
            RA(J,2)=RA(J,2) + ACC*CY
            RA(J,3)=RA(J,3) + ACC*CZ
            RD(J,1)=RD(J,1) + DIS*CX
            RD(J,2)=RD(J,2) + DIS*CY
            RD(J,3)=RD(J,3) + DIS*CZ
   20     CONTINUE
        ELSE IF(TEST(OPTION,'DISP'..FALSE. ) ) THEN
          DO 22 J=1,NMODE
            DIS=TMP(J)*ASCALE
            ACC=TMP(J)*ASCALE*((6.2831853/EVAL(J))**2)
            RA(J,1)=RA(J,1) + ACC*CX
            RA(J,2)=RA(J,2) + ACC*CY
            RA(J,3)=RA(J,3) + ACC*CZ
            RD(J,1)=RD(J,1) + DIS*CX
            RD(J,2)=RD(J,2) + DIS*CY
            RD(J,3)=RD(J,3) + DIS*CZ
   22     CONTINUE
        ENDIF
C
C
   30 CONTINUE
      IF (PRINT) THEN
        WRITE (6,31) 'X'
        WRITE (6,93) (RA(J,1),J=1,NMODE)
        WRITE (6,31) 'Y'
        WRITE (6,93) (RA(J,2),J=1,NMODE)
        WRITE (6,31) 'Z'
        WRITE (6,93) (RA(J,3),J=1,NMODE)
   31   FORMAT(/1X,'GLOBAL BASE ACCE.-SPECTRUM VALUES IN THE ',A,
     &     ' DIRECTION ',/)
        WRITE (6,319) 'X'
        WRITE (6,93) (RD(J,1),J=1,NMODE)
        WRITE (6,319) 'Y'
        WRITE (6,93) (RD(J,2),J=1,NMODE)
        WRITE (6,319) 'Z'
        WRITE (6,93) (RD(J,3),J=1,NMODE)
  319   FORMAT(/1X,'GLOBAL BASE DISP.-SPECTRUM VALUES IN THE ',A,
     &     ' DIRECTION ',/)
      ENDIF
C
C-------------------------------------------------
C== CALCULATE IUNFLUENCE COEFFICIENT VECTORS ==
C-------------------------------------------------
      DO 440 I=1,NDOF
      DO 440 J=1,3
  440  FLU(I,J)=0
C
      DO 451 K=1,3
C
C..... OMIT ALL DOF THAT ARE CONSTRAINED TO A MASTER DOF...
        DO 554 NODE=1,NNODE
             ICOS=JTCOS(NODE)
             DO 650 JDIR=1,3
                  IF (.NOT. BTEST(JTFLG(NODE),JDIR+11)) THEN
                       II=IDOF(NODE,JDIR)
                       JJ=II-NCOND
                       IF(JJ .GE. 1 .AND. JJ .LE. NFREE ) THEN
                          FLU(JJ,K)=COSINE(JDIR,K,ICOS)
                       ENDIF
                  ENDIF
  650        CONTINUE
  554   CONTINUE
  451 CONTINUE
C
      IF(BTEST(IBUG,5))CALL WMATRX(FLU,NFREE,3,'INFLUENCE COEF. MATRIX')
C
   92 FORMAT(1X,'DIRECTION OF COSINE:',
     &    4X,5E,F9.5,' I',SP,F9.5,' J',F9.5,' K  ',SS,/)
   93 FORMAT (1X,1P,10G12.4)
   94 FORMAT (/1X,'INPUT RESPONSE SPECTRUM VALUES:',/)
  929 FORMAT(1X,'DIRECTION OF DISPLACEMENT:',
     &    4X,5E,F9.5,' I',SP,F9.5,' J',F9.5,' K  ',SS)
C-------------------------------------------------------
C== CALCULATE CROSS-MODAL COEFFICIENTS FOR CQC APPROACH ==
C-------------------------------------------------------
      IF( ICOMN .EQ. 2 ) THEN
        DO 809 I=1,NMODE
          CMC(I,I)= 1
        DO 809 J=I+1,NMODE
          WI=EVAL(I)
          WJ=EVAL(J)
          WJI=WJ/WI
          COEF1=8*(DAMP**2)*(1+WJI)*(WJI**1.5)
          COEF2=(1-WJI**2)**2 + 4*(DAMP*2)*WJI*(1+WJI)**2
          CMC(I,J)=COEF1/COEF2
          CMC(J,I)=CMC(I,J)
  809   CONTINUE
C
      WRITE(6,10)TITLE(1),TITLE(2)
      CALL WMATRX(CMC,NMODE,NMODE,'CROSS-MODAL COF. MATRIX')
      ENDIF
C-----------------------------------------------
C== CALCULATE GENERALIZED MASS, GM=EVEC^T*E*EVEC ==
C-----------------------------------------------
      DO 450 I=1,NMODE
          GM(I)=0
      DO 460 II=1,NFREE
          DISP(II)=0
      DO 470 JJ=1,NFREE
          IF( E(II,JJ) .EQ. 0 ) GO TO 470
          DISP(II)=DISP(II) + E(II,JJ)*EVEC(JJ,I)
  470 CONTINUE
  460 CONTINUE
      DO 480 KK=1,NFREE
          GM(I)=GM(I)+EVEC(KK,I)*DISP(KK)
  480 CONTINUE
  450 CONTINUE
      IF(BTEST(IBUG,5))
     & CALL WMATRX(GM,NMODE,1,'GENERIZED MASS MATRIX')
C-------------------------------------------------------------
C== CALCULATE EFFECTIVE-LOAD FACTOR FOR EACH MODE, ELF=EVEC^T*E*FLU ==
C-------------------------------------------------------------
      DO 455 I=1,NMODE
      DO 457 K=1,3
          ELF(I,K)=0
      DO 465 II=1,NFREE
          DISP(II)=0
      DO 475 JJ=1,NFREE
          IF( E(II,JJ) .EQ. 0 ) GO TO 475
          DISP(II)=DISP(II) + E(II,JJ)*FLU(JJ,K)
  475 CONTINUE
  465 CONTINUE
      DO 485 KK=1,NFREE
          ELF(I,K)=ELF(I,K)+EVEC(KK,I)*DISP(KK)
  485 CONTINUE
  457 CONTINUE
  455 CONTINUE
      IF(BTEST(IBUG,5))
     & CALL WMATRX(ELF,NMODE,3,'EFFECTIVE-LOAD MATRIX')
C-------------------------------------------------------------
C== CALCULATE EFFECTIVE MASS FOR EACH MODE IN EACH INDIVIDUAL ==
C   SEISMIC INPUT DIRECTION
C-------------------------------------------------------------
C
C-----READ EFFECTIVE MASS RATIO LIMIT IN X, Y, Z DIRECION
C     NOTE: IF NO. OF MODE BASED ON THE LIMIT IS LESS THAN NMODE.
C           NO. OF MODES BASED ON THIS LIMIT IS USED.
C
      READ(5,*)XLIM,YLIM,ZLIM
C
      WRITE(6,10)TITLE(1),TITLE(2)
      WRITE(6,487)
  487 FORMAT(1X,'EFFECTIVE MASS IN GCS' TRANSLATIONAL DIRECTIONS',/
     &       1X,'-----------------------------------------',/)
C
      WRITE(6,987)'X',XLIM
      WRITE(6,987)'Y',YLIM
      WRITE(6,987)'Z',ZLIM
C
C....CALCULATE TOTAL STRUCTURAL TRANSLATIONAL MASS
      TOTX=0
      TOTY=0
      TOTZ=0
      DO 705 J=L3,L4
        IJ=MDMS(J)
        DO 706 I=1,NNODE
            IF( .NOT. BTEST(JTFLG(I),12) ) THEN
              NDOFX=IDOF(I,1)
              IF(J .EQ. NDOFX)TOTX=TOTX + MASS(IJ)
            ENDIF
            IF( .NOT. BTEST(JTFLG(I),13) ) THEN
              NDOFY=IDOF(I,2)
              IF(J .EQ. NDOFY)TOTY=TOTY + MASS(IJ)
            ENDIF
            IF( .NOT. BTEST(JTFLG(I),14) ) THEN
              NDOFZ=IDOF(I,3)
              IF(J .EQ. NDOFZ)TOTZ=TOTZ + MASS(IJ)
            ENDIF
  706   CONTINUE
  705 CONTINUE
      WRITE(6,388) 'X',TOTX
      WRITE(6,388) 'Y',TOTY
      WRITE(6,388) 'Z',TOTZ
  388 FORMAT(1X,'NOTE: TOTAL STRUCTURAL MASS IN ',A,' DIRECTION IS ',
     &       0P,G15.6)
      WRITE(6,386)
  386 FORMAT(1X,'   MODE     ','GCS X-DIRECTION(%)   ',
     &                        'GCS Y-DIRECTION(%)   ',
     &                        'GCS Z-DIRECTION(%)   ')
  987 FORMAT(1X,'EFFECTIVE MASS LIMIT IN GCS' ',A,
     &          ' DIRECTION IS ',G15.6 )
C
      TXLIM=0
      TYLIM=0
      TZLIM=0
      MODEX=0
      MODEY=0
      MODEZ=0
      FXME=0
      FYME=0
      FZME=0
      DO 701 I=1,NMODE
        IF (TOTX .EQ. 0 ) GO TO 707
            FXME=(ELF(I,1)**2/GM(I))/TOTX
            TXLIM=TXLIM + FXME
            IF( XLIM .NE. 0 .AND. MODEX .EQ. 0 ) THEN
                IF( TXLIM .GE. XLIM ) MODEX=I
            ENDIF
  707   IF (TOTY .EQ. 0 ) GO TO 708
            FYME=(ELF(I,2)**2/GM(I))/TOTY
            TYLIM=TYLIM + FYME
            IF( YLIM .NE. 0 .AND. MODEY .EQ. 0 ) THEN
                IF( TYLIM .GE. YLIM ) MODEY=I
            ENDIF
  708   IF (TOTZ .EQ. 0 ) GO TO 709
            FZME=(ELF(I,3)**2/GM(I))/TOTZ
            TZLIM=TZLIM + FZME
            IF( ZLIM .NE. 0 .AND. MODEZ .EQ. 0 ) THEN
                IF( TZLIM .GE. ZLIM ) MODEZ=I
            ENDIF
  709   WRITE(6,389)I,FXME,FYME,FZME
  389   FORMAT(5X,I3,5X,3(G15.4,6X))
  701 CONTINUE
C
      IMO=MAX(MODEX,MODEY,MODEZ)
      IF( IMO .NE. 0 .AND. IMO .LT. NMODE) THEN
        NMODE=IMO
        WRITE(6,*)' ** BASED ON EFFECTIVE MASS LIMIT, FINAL NO. OF ',
     &            'MODES USED IS ',NMODE,' **'
      ELSE
        NMODE=NMODE
      ENDIF
C-----------------------------------------
C== CALCULATE MAXIMUM DISPLACEMENT RESPONSES ==
C-----------------------------------------
C
      DO 560 NN=1,NA
      DO 540 I=1,L6
  540   TMP(I)=0
      DO 541 I=1,NFREE
  541   TMP2(I)=0
      DO 550 J=1,NMODE
        DO 559 I=1,NFREE
          DISP(I)=0
  559 CONTINUE
        DO 570 II=1,NFREE
          DISP(II)=EVEC(II,J)*ELF(J,NN)*RD(J,NN)/GM(J) + DISP(II)
  570   CONTINUE
C
C
        IF(ICOMN .EQ. 2 ) THEN
          DO 815 I=1,NFREE
  815       CQCM(I+NCOND,J)=DISP(I)
        ENDIF
C
        IF(BTEST(IBUG,5)) THEN
          WRITE(6,*)'MODAL DISPLACEMENTS FOR MODE ',J
          DO 9999 I=1,NFREE
 9999       WRITE(6,*)'DISP(',I,')= ',DISP(I)
        ENDIF
C
        IF( ICOMN .EQ. 1 ) THEN
          DO 580 KK=1,NFREE
            TMP(KK+NCOND)=DISP(KK)**2 + TMP(KK+NCOND)
  580     CONTINUE
        ENDIF
C
        DO 500 LL=1,NFREE
  500     DLOAD(LL,1)=DISP(LL)
        DO 501 LL=L1,L2
  501     DISP(LL)=0
        DO 502 LL=L3,L4
  502     DISP(LL)=DLOAD(LL-L2,1)
        DO 503 LL=L5,L6
  503     DISP(LL)=0
C
C-----------------------------------------
C== CLACULATE EQUIVALENT EARTHQUAKE FORCES ==
C-----------------------------------------
      DO 753 I=L5,L6
  753   LOAD(I)=0
C
      DO 735 I=1,NFREE
        LOAD(I+NCOND)=0
      DO 735 JJ=1,NFREE
        LOAD(I+NCOND)=LOAD(I+NCOND)+A(I,JJ)*DISP(JJ+NCOND)
  735 CONTINUE
      IF(BTEST(IBUG,5) ) THEN
```

```
      WRITE(6,10)TITLE(1),TITLE(2)
      WRITE(6,605)J,NN
  605 FORMAT(1X,'TOTAL EQUIVALENT EARTHQUAKE FORCES FOR MODE ',I3,
     &       ' IN THE SEISMIC COMPONENT # ',I2/)
       WRITE(6,913)
       WRITE(6,212) (I+NCOND,LOAD(I+NCOND),I=1,NFREE)
      ENDIF
C
      IF( ICOMN .EQ. 1 ) THEN
        DO 581 KK=1,NFREE
  581     TMP2(KK)=LOAD(KK+NCOND)**2 +TMP2(KK)
      ELSE IF( ICOMN .EQ. 2 ) THEN
        DO 816 I=1,NFREE
  816     CQCM2(I,J)=LOAD(I+NCOND)
      ENDIF
C
C*****************************************
C== CALC RESPONSE FOR CONDENSED OUT DOF ==
C*****************************************
          II=1
          CALL REACTN(I1,BTEST(IBUG,6),NNODE,L1,L2,L3,L4,DISP,DISP,
     &         MD,STIFF,IMD,NDOF,1,1.0,
     &         MAXNOD,IDOF,COORD,ID,TITLE,STEPID,.FALSE.,
     &           NCOS,COSINE,JTCOS,JTFLG,SUMRCT)
          CALL MSOLL(3,BTEST(IBUG,6),L4,IMD,NDOF,I1,L1,L2,
     &               MD,STIFF,DISP,WORK,
     &             .FALSE.,MASS,MDMS,1,
     &             .FALSE.,KG   ,MDKG,1,ABORT)
          IF (ABORT) RETURN
C
      IF(ICOMN .EQ. 1 ) THEN
        DO 675 I=L1,L2
  675     TMP(I)=DISP(I)**2 + TMP(I)
      ELSE IF(ICOMN .EQ. 2 ) THEN
        DO 676 I=L1,L2
  676     CQCM(I,J)=DISP(I)
      ENDIF
C
C*************************************************
C= PRINT GLOBAL EQUIVALENT EARTHQUAKE FORCE ==
C*************************************************
C
C     CALL JOIDSP(1,MAXNOD,L6,NNODE,1,L3,L4,ID,
C    &       IDOF,CNST,LOAD,
C    &       TITLE,'TOTAL EQUIVALENT EARTHQUAKE FORCES',STEPID,.TRUE.,
C    &       NCOS,COSINE,JTCOS,JTFLG)
C
C*****************************************
C= CALCULATE MEMBER FORCES ===
C*****************************************
      IOPT=4
      LSTTYP=0
      PRINT=BTEST(IBUG,5)
      HEAD =.FALSE.
      DO 358 IELNO=1,NELMT
        CALL ELELIB
     &  (IOPT  ,LSTTYP ,PRINT ,IREL,HEAD,   NAME ,EESE,EPSE,DAMAGE,
     &   DUCFAG,
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)
        DO 359 I=1,12
  359     ELEF(IELNO,I,J)=RSAF(I)
  358 CONTINUE
C
C*************************
C= SOLVE FOR REACTIONS ===
C*************************
C
      CALL REACTN(2,BTEST(IBUG,5),NNODE,L1,L4,L5,L6,LOAD,DISP,
     &       MD,STIFF,IMD,NDOF,1,1.0,
     &       MAXNOD,IDOF,COORD,ID,TITLE,STEPID,.FALSE.,
     &       NCOS,COSINE,JTCOS,JTFLG,SUMRCT)
C
C*********************************
C= SOLVE FOR GLOBAL REACTIONS ===
C*********************************
C
        DO 366 I=1,6
  366     SUMRCT(I)=0
        CALL REACTN(2,BTEST(IBUG,5),NNODE,L1,L4,L5,L6,LOAD,DISP,
     &       MD,STIFF,IMD,NDOF,1,1.0,
     &       MAXNOD,IDOF,COORD,ID,TITLE,STEPID,.FALSE.,
     &       NCOS,COSINE,JTCOS,JTFLG,SUMRCT)
        DO 360 I=1,6
  360     RCT(I,J)=SUMRCT(I)
C
       IF( BTEST(IBUG,5)) THEN
         WRITE(6,10)TITLE(1),TITLE(2)
         WRITE(6,607)J,NN
  607    FORMAT(1X,'REACTION FOR MODE ',I3,' IN SEISMIC COMP # ',I2/)
         WRITE(6,323)SUMRCT
       ENDIF
C
  550   CONTINUE
C
      IF( ICOMN .EQ. 1 ) THEN
        DO 590 JJ=1,L4
  590     TMP(JJ)=SQRT(TMP(JJ))
        DO 547 JJ=1,NFREE
  547     TMP2(JJ)=SQRT(TMP2(JJ))
        DO 591 I=1,NELMT
        DO 591 II=1,12
        DO 592 J=1,MMODE
  592     ELEF(I,II,MMODE+NN)=ELEF(1,II,MMODE+NN)+ELEF(I,II,J)**2
          ELEF(I,II,MMODE+NN)=SQRT(ELEF(I,II,MMODE+NN) )
  591   CONTINUE
        DO 593 I=1,6
        DO 594 J=1,MMODE
  594     RCT(I,MMODE+NN)=RCT(I,MMODE+NN)+RCT(I,J)**2
          RCT(I,MMODE+NN)=SQRT(RCT(I,MMODE+NN) )
  593   CONTINUE
C
      ELSE IF(ICOMN .EQ. 2 ) THEN
          DO 826 L=1,L4
            SUM=0
            DO 828 J=1,MMODE
            DO 828 JJ=1,MMODE
  828         SUM=SUM+CQCM(L,J)*CMC(J,JJ)*CQCM(L,JJ)
            SUM=ABS(SUM)
            TMP(L)=SQRT(SUM)
  826     CONTINUE
C
          DO 837 L=1,NFREE
            SUM=0
            DO 838 J=1,MMODE
            DO 838 JJ=1,MMODE
  838         SUM=SUM+CQCM2(L,J)*CMC(J,JJ)*CQCM2(L,JJ)
            SUM=ABS(SUM)
            TMP2(L)=SQRT(SUM)
  837     CONTINUE
C
          DO 595 I=1,NELMT
          DO 595 II=1,12
            SUM=0
          DO 596 J=1,MMODE
          DO 596 JJ=1,MMODE
  596       SUM=SUM+ELEF(I,II,J)*CMC(J,JJ)*ELEF(I,II,JJ)
          SUM=ABS(SUM)
          ELEF(I,II,MMODE+NN)=SQRT(SUM)
  595     CONTINUE
          DO 597 I=1,6
            SUM=0
          DO 598 J=1,MMODE
SOL05910
SOL05920
SOL05930
SOL05940
SOL05950
SOL05960
SOL05970
SOL05980
SOL05990
SOL06000
SOL06010
SOL06020
SOL06030
SOL06040
SOL06050
SOL06060
SOL06070
SOL06080
SOL06090
SOL06100
SOL06110
SOL06120
SOL06130
SOL06140
SOL06150
SOL06160
SOL06170
SOL06180
SOL06190
SOL06200
SOL06210
SOL06220
SOL06230
SOL06240
SOL06250
SOL06260
SOL06270
SOL06280
SOL06290
SOL06300
SOL06310
SOL06320
SOL06330
SOL06340
SOL06350
SOL06360
SOL06370
SOL06380
SOL06390
SOL06400
SOL06410
SOL06420
SOL06430
SOL06440
SOL06450
SOL06460
SOL06470
SOL06480
SOL06490
SOL06500
SOL06510
SOL06520
SOL06530
SOL06540
SOL06550
SOL06560
SOL06570
SOL06580
SOL06590
SOL06600
SOL06610
SOL06620
SOL06630
SOL06640
SOL06650
SOL06660
SOL06670
SOL06680
SOL06690
SOL06700
SOL06710
SOL06720
SOL06730
SOL06740
SOL06750
SOL06760
SOL06770
SOL06780
SOL06790
SOL06800
SOL06810
SOL06820
SOL06830
SOL06840
SOL06850
SOL06860
SOL06870
SOL06880
SOL06890
SOL06900
SOL06910
SOL06920
SOL06930
SOL06940
SOL06950
SOL06960
SOL06970
SOL06980
SOL06990
SOL07000
SOL07010
SOL07020
SOL07030
SOL07040
SOL07050
SOL07060
SOL07070
SOL07080
SOL07090
SOL07100
SOL07110
SOL07120
SOL07130
SOL07140
SOL07150
SOL07160
SOL07170
SOL07180
SOL07190
SOL07200
SOL07210
SOL07220
SOL07230
SOL07240
SOL07250
SOL07260
SOL07270
SOL07280
SOL07290
SOL07300
SOL07310
SOL07320
          DO 598 JJ=1,MMODE
  598       SUM=SUM+RCT(I,J)*CMC(J,JJ)*RCT(I,JJ)
          SUM=ABS(SUM)
          RCT(I,MMODE+NN)=SQRT(SUM)
  597     CONTINUE
C       ENDIF
C
        IF( NA .GT. 1 ) THEN
        WRITE(6,10)TITLE(1),TITLE(2)
        WRITE(6,489)NN
  489 FORMAT(1X,'MAXIMUM STRUCTURAL RESPONSES ',
     &         'DUE TO SEISMIC COMPONENT # ',I2,/
     &       1X,'------------------------------',
     &         '------------------------------',/)
        WRITE(6,913)
  913 FORMAT(3(' | D.O.F.    DISPL     |'),/)
        WRITE(6,212) (I+NCOND,TMP(I+NCOND),I=1,NFREE)
C
        WRITE(6,10)TITLE(1),TITLE(2)
C       WRITE(6,489)NN
C       DO 193 J=L5,L6
C 193     TMP(J)=0
C
C       CALL JOIDSP(1,MAXNOD,L6,NNODE,1,L3,L4,ID,
C    &       IDOF,CNST,TMP,
C    &       TITLE,'MAXIMUM DISPLACEMENTS',STEPID,.FALSE.,
C    &       NCOS,COSINE,JTCOS,JTFLG)
C
C----CALCULATE TOTAL ELEMENT FORCES
          WRITE(6,10)TITLE(1),TITLE(2)
          WRITE(6,984)NN
  984     FORMAT(1X,'MAXIMUM ELEMENT FORCES ',
     &           'DUE TO SEISMIC COMPONENT # ',I2,/
     &         1X,'--------------------------',
     &           '--------------------------'/)
          IOPT=5
          LSTTYP=0
          PRINT=.TRUE.
          HEAD=.FALSE.
        DO 905 I=1,NELMT
          IELNO=I
          DO 959 II=1,12
  959       RSAF(II)=ELEF(I,II,MMODE+NN)
          CALL ELELIB
     &  (IOPT, LSTTYP ,PRINT ,IREL,HEAD,   NAME ,EESE,EPSE,DAMAGE,
     &   DUCFAG,
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)
  905 CONTINUE
C
C----CALCULATE TOTAL EQUIVALENT EARTHQUAKE FORCES
          WRITE(6,10)TITLE(1),TITLE(2)
          WRITE(6,1841)NN
  184     FORMAT(1X,'MAXIMUM EQUIVALENT EARTHQUAKE FORCES ',
     &           'DUE TO SEISMIC COMPONENT # ',I2,/
     &         1X,'----------------------------------------',
     &           '------------------------------'/)
          WRITE(6,912)
          WRITE(6,212)(I+NCOND,TMP2(I),I=1,NFREE)
C
C----CALCULATE TOTAL REACTION AT GCS' ORIGIN
        DO 696 I=1,6
  696     SUMRCT(I)=RCT(I,MMODE+NN)
        WRITE(6,10)TITLE(1),TITLE(2)
        WRITE(6,232)NN
  232   FORMAT(///' TOTAL REACTIONS AT GCS' ORIGIN ',
     &           'DUE TO SEISMIC COMPONENT # ',I2,/
     &           ' ------------------------------ ',
     &           '------------------------------'/)
     &    17X,'FX',13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',/)
        WRITE(6,323)SUMRCT
      ENDIF
C
C------ STORE RESPONSE TO DISP ARRAY
          DO 200 J=L1,L2
  200       DLOAD(J,NN)=TMP(J)
          DO 210 J=L3,L4
  210       DLOAD(J,NN)=TMP(J)
          DO 220 J=L5,L6
  220       DLOAD(J,NN)=0
C
C------ STORE EQUIVALENT EARTHQUAKE FORCES IN EEF ARRAY
          DO 291 J=1,NFREE
  291       EEF(J,NN)=TMP2(J)
C
  560 CONTINUE
C
C-----CALCULATE TOTAL RESPONSES
        DO 237 J=L1,L6
          DISP(J)=0
        DO 237 I=1,NA
  237   DISP(J)=DISP(J)+DLOAD(J,I)
        WRITE(6,10)TITLE(1),TITLE(2)
        WRITE(6,608)
  608 FORMAT(1X,'TOTAL MAXIMUM STRUCTURAL RESPONSES',/
     &         1X,'================================',/)
         WRITE(6,913)
         WRITE(6,212) (I+NCOND,DISP(I+NCOND),I=1,NFREE)
C
C-----CALCULATE TOTAL EQUIVALENT EARTHQUAKE FORCE
        DO 202 J=1,NFREE
          TMP2(J)=0
        DO 202 I=1,NA
  202   TMP2(J)=TMP2(J)+EEF(J,I)
        WRITE(6,10)TITLE(1),TITLE(2)
        WRITE(6,678)
  678 FORMAT(1X,'TOTAL MAXIMUM EQUIVALENT EARTHQUAKE FORCES',/
     &         1X,'------------------------------------',/)
         WRITE(6,912)
         WRITE(6,212) (I+NCOND,TMP2(I),I=1,NFREE)
C
C*********************************
C= PRINT GLOBAL DISPLACEMENTS ==
C*********************************
C     CALL JOIDSP(1,MAXNOD,L6,NNODE,1,L3,L4,ID,
C    &       IDOF,CNST,DISP,
C    &       TITLE,'TOTAL MAXIMUM DISPLACEMENTS',STEPID,.TRUE.,
C    &       NCOS,COSINE,JTCOS,JTFLG)
C
C----CALCULATE TOTAL ELEMENT FORCES
          WRITE(6,10)TITLE(1),TITLE(2)
          WRITE(6,980)
  980   FORMAT(1X,'TOTAL MAXIMUM ELEMENT FORCES:',/
     &         1X,'---------------------------',/)
          IOPT=5
          LSTTYP=0
          PRINT=.TRUE.
          HEAD=.FALSE.
        DO 509 I=1,NELMT
          IELNO=I
          DO 599 II=1,12
            RSAF(II)=0
          DO 599 J=1,NA
  599       RSAF(II)=RSAF(II)+ELEF(I,II,MMODE+J)
          CALL ELELIB
     &  (IOPT, LSTTYP ,PRINT ,IREL,HEAD,   NAME ,EESE,EPSE,DAMAGE,
     &   DUCFAG,
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)
  509 CONTINUE
C
C----CALCULATE TOTAL REACTION AT GCS' ORIGIN
        DO 699 I=1,6
SOL07330
SOL07340
SOL07350
SOL07360
SOL07370
SOL07380
SOL07390
SOL07400
SOL07410
SOL07420
SOL07430
SOL07440
SOL07450
SOL07460
SOL07470
SOL07480
SOL07490
SOL07500
SOL07510
SOL07520
SOL07530
SOL07540
SOL07550
SOL07560
SOL07570
SOL07580
SOL07590
SOL07600
SOL07610
SOL07620
SOL07630
SOL07640
SOL07650
SOL07660
SOL07670
SOL07680
SOL07690
SOL07700
SOL07710
SOL07720
SOL07730
SOL07740
SOL07750
SOL07760
SOL07770
SOL07780
SOL07790
SOL07800
SOL07810
SOL07820
SOL07830
SOL07840
SOL07850
SOL07860
SOL07870
SOL07880
SOL07890
SOL07900
SOL07910
SOL07920
SOL07930
SOL07940
SOL07950
SOL07960
SOL07970
SOL07980
SOL07990
SOL08000
SOL08010
SOL08020
SOL08030
SOL08040
SOL08050
SOL08060
SOL08070
SOL08080
SOL08090
SOL08100
SOL08110
SOL08120
SOL08130
SOL08140
SOL08150
SOL08160
SOL08170
SOL08180
SOL08190
SOL08200
SOL08210
SOL08220
SOL08230
SOL08240
SOL08250
SOL08260
SOL08270
SOL08280
SOL08290
SOL08300
SOL08310
SOL08320
SOL08330
SOL08340
SOL08350
SOL08360
SOL08370
SOL08380
SOL08390
SOL08400
SOL08410
SOL08420
SOL08430
SOL08440
SOL08450
SOL08460
SOL08470
SOL08480
SOL08490
SOL08500
SOL08510
SOL08520
SOL08530
SOL08540
SOL08550
SOL08560
SOL08570
SOL08580
SOL08590
SOL08600
SOL08610
SOL08620
SOL08630
SOL08640
SOL08650
SOL08660
SOL08670
SOL08680
SOL08690
SOL08700
SOL08710
SOL08720
SOL08730
SOL08740
```

```
        SUMRCT(I)=0                                                     SOL08750
      DO 699 J=1,NA                                                     SOL08760
        SUMRCT(I)=SUMRCT(I)+RCT(I,MMODE+J)                              SOL08770
  699 CONTINUE                                                          SOL08780
      WRITE(6,10)TITLE(1),TITLE(2)                                      SOL08790
      WRITE(6,322)                                                      SOL08800
  322 FORMAT(//' TOTAL REACTIONS AT GCS' ORIGIN ',/                     SOL08810
     &       ' ============================== ',/                      SOL08820
     & 17X,'FX',13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',/)         SOL08830
      WRITE(6,323)SUMRCT                                                SOL08840
  323 FORMAT(' GCS SUMM.',6G15.7)                                       SOL08850
C                                                                       SOL08860
C*****************************************************                SOL08870
C** CALCULATE MAXIMUM INERTIA EARTHQUAKE FORCES **                     SOL08880
C*****************************************************                SOL08890
C                                                                       SOL08900
      DO 207 NN=1,NA                                                    SOL08910
      DO 898 I=1,NFREE                                                  SOL08920
  898   TMP(I)=0                                                        SOL08930
      DO 204 J=1,MMODE                                                  SOL08940
        DO 201 I=1,NFREE                                                SOL08950
  201     DISP(I)=0                                                     SOL08960
        DO 205 II=1,NFREE                                               SOL08970
          WORK(II)=0                                                    SOL08980
          DO 206 JJ=1,NFREE                                             SOL08990
            IF( B(II,JJ) .EQ. 0 ) GO TO 206                             SOL09000
            WORK(II)=WORK(II)+B(II,JJ)*EVEC(JJ,J)                       SOL09010
  206     CONTINUE                                                      SOL09020
  205   CONTINUE                                                        SOL09030
C                                                                       SOL09040
        DO 208 II=1,NFREE                                               SOL09050
          DISP(II)=WORK(II)*ELF(J,NN)*RA(J,NN)/GM(J) + DISP(II)         SOL09060
  208   CONTINUE                                                        SOL09070
C                                                                       SOL09080
      IF(ICOMN .EQ. 2 ) THEN                                            SOL09090
        DO 915 I=1,NFREE                                                SOL09100
  915     CQCM(I,J)=DISP(I)                                             SOL09110
      ENDIF                                                             SOL09120
C                                                                       SOL09130
      IF ( BTEST(IBUG,5) ) THEN                                         SOL09140
        WRITE(6,*)'MODAL FORCE FOR MODE ',J                             SOL09150
        DO 8888 I=1,NFREE                                               SOL09160
 8888     WRITE(6,*)'DISP(',I,')= ',DISP(I)                             SOL09170
      ENDIF                                                             SOL09180
C                                                                       SOL09190
      IF(ICOMN .EQ. 1 ) THEN                                            SOL09200
        DO 209 KK=1,NFREE                                               SOL09210
          TMP(KK)=DISP(KK)**2 +TMP(KK)                                  SOL09220
  209   CONTINUE                                                        SOL09230
      ENDIF                                                             SOL09240
  204 CONTINUE                                                          SOL09250
C                                                                       SOL09260
      IF(ICOMN .EQ. 1 ) THEN                                            SOL09270
        DO 213 JJ=1,NFREE                                               SOL09280
         TMP(JJ)=SQRT(TMP(JJ))                                          SOL09290
  213   CONTINUE                                                        SOL09300
      ELSE IF(ICOMN .EQ. 2 ) THEN                                       SOL09310
        DO 926 L=1,NFREE                                                SOL09320
          SUM=0                                                         SOL09330
          DO 928 J=1,MMODE                                              SOL09340
          DO 928 JJ=1,MMODE                                             SOL09350
  928       SUM=SUM+CQCM(L,J)*CMC(J,JJ)*CQCM(L,JJ)                      SOL09360
          SUM=ABS(SUM)                                                  SOL09370
          TMP(L)=SQRT(SUM)                                              SOL09380
  926   CONTINUE                                                        SOL09390
      ENDIF                                                             SOL09400
C                                                                       SOL09410
     IF( NA .GT. 1 ) THEN                                               SOL09420
     WRITE(6,10)TITLE(1),TITLE(2)                                       SOL09430
     WRITE(6,488)NN                                                     SOL09440
  488 FORMAT(1X,'MAXIMUM INERTIA FORCES ',                              SOL09450
     &          'DUE TO SEISMIC COMPONENT # ',I2./                     SOL09460
     &       1X,'------------------------',                            SOL09470
     &          '------------------------------',/)                   SOL09480
      WRITE(6,912)                                                      SOL09490
  912 FORMAT(3(' | D.O.F.    FORCE     |'),/)                           SOL09500
      WRITE(6,212) (I+NCOND,TMP(I),I=1,NFREE)                           SOL09510
  212   FORMAT(3(3X,I6,1X,G15.4))                                       SOL09520
      ENDIF                                                             SOL09530
C                                                                       SOL09540
C....STORE EARTHQUAKE FORCES TO LOAD ARRAY                              SOL09550
C                                                                       SOL09560
        DO 233 J=L1,L2                                                  SOL09570
  233     DLOAD(J,NN)=0                                                 SOL09580
        DO 234 J=L3,L4                                                  SOL09590
  234     DLOAD(J,NN)=TMP(J-L2)                                         SOL09600
        DO 235 J=L5,L6                                                  SOL09610
  235     DLOAD(J,NN)=0                                                 SOL09620
  207 CONTINUE                                                          SOL09630
C                                                                       SOL09640
C-----CALCULATE TOTAL FORCES                                            SOL09650
      DO 236 J=L1,L6                                                    SOL09660
        LOAD(J)=0                                                       SOL09670
      DO 236 I=1,NA                                                     SOL09680
  236   LOAD(J)=LOAD(J)+DLOAD(J,I)                                      SOL09690
      WRITE(6,10)TITLE(1),TITLE(2)                                      SOL09700
      WRITE(6,588)                                                      SOL09710
  588 FORMAT(1X,'TOTAL MAXIMUM INERTIA FORCES'./                        SOL09720
     &       1X,'============================',/)                      SOL09730
      WRITE(6,912)                                                      SOL09740
      WRITE(6,212) (I+NCOND,LOAD(I+NCOND),I=1,NFREE)                    SOL09750
C------------------------                                               SOL09760
C= PRINT GLOBAL FORCES ==                                               SOL09770
C------------------------                                               SOL09780
C                                                                       SOL09790
C     CALL JOIDSP(1,MAXNOD,L6,NNODE,1,L3,L4,ID,                         SOL09800
C    &       IDOF,CNST,LOAD,                                            SOL09810
C    &       TITLE,'TOTAL MAXIMUM FORCES',STEPID,.TRUE.,                SOL09820
C    &       NCOS,COSINE,JTCOS,JTFLG)                                   SOL09830
C                                                                       SOL09840
      RETURN                                                            SOL09850
      END                                                               SOL09860
C*********************************************************************STR00010
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                      STR00020
C         END DEBUG                                                     STR00030
C*********************************************************************STR00040
      SUBROUTINE STRENG (ESE,PSE,PSEOLD,P,PL,D,DL,S,SE)                 STR00050
C                                                                       STR00060
      IMPLICIT REAL(A-H,O-Z)                                            STR00070
      DIMENSION ESE(3)                                                  STR00080
C---------------------------------------------------------------------STR00090
C VARIABLES:                                                            STR00100
C---------------------------------------------------------------------STR00110
C TSE    = TOTAL ELEMENT STRAIN ENERGY                                  STR00120
C ESE(1) = ELASTIC ELEMENT STRAIN ENERGY                                STR00130
C ESE(2) = MAXIMUM ELASTIC ELEMENT STRAIN ENERGY, POSITIVE LOADING      STR00140
C ESE(3) = MAXIMUM ELASTIC ELEMENT STRAIN ENERGY, NEGATIVE LOADING      STR00150
C PSE    = PLASTIC ELEMENT STRAIN ENERGY                                STR00160
C PSEOLD = PLASTIC ELEMENT STRAIN ENERGY AT LAST ZERO CROSSING          STR00170
C P      = CURRENT LOAD                                                 STR00180
C PL     = PREVIOUS LOAD                                                STR00190
C D      = CURRENT DISPLACEMENT                                         STR00200
C DL     = PREVIOUS DISPLACEMENT                                        STR00210
C S      = CURENT STIFFNESS                                             STR00220
C SE     = UNLOADING STIFFNESS                                          STR00230
C---------------------------------------------------------------------STR00240
C                                                                       STR00250
      PSEL  = PSE                                                       STR00260
C           PREVENT UNDERFLOW FOR VERY SMALL PSUM OR DDIF               STR00270
            PSUM=P+PL                                                   STR00280
            DDIF=D-DL                                                   STR00290
            TOL=1.00E-25                                                STR00300
            IF (ABS(PSUM).LE.TOL .OR. ABS(DDIF).LE.TOL) THEN            STR00310
              DTSE=0                                                    STR00320
            ELSE                                                        STR00330
                DTSE  = PSUM*DDIF/2.00                                  STR00340
            ENDIF                                                       STR00350
      TSE   = ESE(1)+ PSE + DTSE                                        STR00360
C           PREVENT UNDERFLOW FOR VERY SMALL P, OR  SE=0                STR00370
            IF (ABS(P).LE.TOL) THEN                                     STR00380
              ESE(1)=0.                                                 STR00390
            ELSE IF (SE.EQ.0) THEN                                      STR00400
              ESE(1)=TSE                                                STR00410
            ELSE                                                        STR00420
                ESE(1)= P*P/(2*SE)                                      STR00430
            ENDIF                                                       STR00440
      ESE(1)= MIN(TSE,ESE(1))                                           STR00450
      PSE   = TSE-ESE(1)                                                STR00460
C                                                                       STR00470
C........ CALC PLACTIC STRAIN ENERGY AT ZERO CROSSING ...               STR00480
      IF (P*PL.LE.0) THEN                                               STR00490
        DF=PL-P                                                         STR00500
        IF (DF.NE.0) THEN                                               STR00510
          IF( (PSEL - PSE) .LT. TOL) THEN                               STR00520
            PSEOLD=PSE                                                  STR00530
          ELSE                                                          STR00540
            PSEOLD=PSE+ (PSEL-PSE)*P/(PL-P)                             STR00550
          ENDIF                                                         STR00560
        ELSE                                                            STR00570
            PSEOLD=PSE                                                  STR00580
        ENDIF                                                           STR00590
      ENDIF                                                             STR00600
C                                                                       STR00610
C........ DETERMING MAXIMUM ELASTIC STRAIN ENERGY/SIDE...               STR00620
      IF (P.GE.0) THEN                                                  STR00630
        ESE(2)=MAX(ESE(2),ESE(1))                                       STR00640
      ELSE                                                              STR00650
        ESE(3)=MAX(ESE(3),ESE(1))                                       STR00660
      ENDIF                                                             STR00670
C                                                                       STR00680
      RETURN                                                            STR00690
      END                                                               STR00700
@PROCESS SDUMP GOSTMT XREF                                              STR00010
C*********************************************************************STR00020
C         DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                      STR00030
C         END DEBUG                                                     STR00040
C*********************************************************************STR00050
      SUBROUTINE STRUCT                                                 STR00060
      DIMENSION NOINP(8,2),RINPUT(100),GINPUT(100),DUCT(3),EXCR(3)      STR00070
      CHARACTER*80 NAME,TYPE                                            STR00080
      DIMENSION SE(100)                                                 STR00090
      LOGICAL FIRST,PT,HEAD,TEST,AXIAL,HEAD2,BTEST                      STR00100
      INCLUDE (ZCOMN)                                                   STR00110
      WRITE (6,5) TITLE(1),TITLE(2)                                     STR00120
    5 FORMAT ('1 STRUCTURE....: ',A,/                                   STR00130
     &        '  SOLUTION.....: ',A,/)                                  STR00140
      WRITE (6,6)                                                       STR00150
    6 FORMAT (' *** PROGRAM FEM ***          DOUBLE PRECISION VERSION'//)STR00160
C                                                                       STR00170
C------- INPUT # OF NODES, # OF SUPTS, # OF CONST., AND SCALE FACTOR    STR00180
      READ (5,*) NNODE,NCOS,NSUPT,NCOND,NCONST,SCALE                    STR00190
      MAXNOD=NNODE                                                      STR00200
C                                                                       STR00210
      NLOAD=1                                                           STR00220
C------------------------                                               STR00230
C= INPUT NODE DATA =====                                                STR00240
C------------------------                                               STR00250
C       ---- NODE DATA STORAGE --------------------                     STR00260
C            N%(I%ID  ) = ID NUMBERS FOR EACH NODE                      STR00270
C            N%(I%IDOF) = DEGREE OF FREEDOM FOR EACH NODE               STR00280
C             %(I%CORD) = GLOBAL COORDINATE OF EACH NODE (X,Y,Z)        STR00290
C             %(I%CNST) = GLOBAL CONSTRAINT MATRIX FOR EACH NODE        STR00300
C            N%(I%WK  ) = TEMPORARY STORAGE                             STR00310
C                                                                       STR00320
      I%ID   = I%                                                       STR00330
      I%     = I%+MAXNOD                                                STR00340
      I%IDOF = I%                                                       STR00350
      I%     = I%+MAXNOD*6                                              STR00360
      I%CORD = I%                                                       STR00370
      I%     = I%+MAXNOD*3                                              STR00380
      I%COS  = I%                                                       STR00390
      I%     = I%+MAXNOD*9                                              STR00400
      I%CNST = I%                                                       STR00410
      I%     = I%+MAXNOD*6                                              STR00420
      I%JFLG = I%                                                       STR00430
      I%     = I%+MAXNOD                                                STR00440
      I%JCOS = I%                                                       STR00450
      I%     = I%+MAXNOD                                                STR00460
      I%WK   = I%                                                       STR00470
      I%WK2  = I%+MAXNOD*6                                              STR00480
C                                                                       STR00490
      CALL NODDOF(NDOF,NCOND,NFREE,NREST,MAXNOD,NNODE,NCOS,             STR00500
     &     SCALE,NSUPT,NCONST,                                          STR00510
     &     N%(I%ID),N%(I%IDOF),N%(I%WK2),%(I%CORD),%(I%COS),%(I%CNST),  STR00520
     &     N%(I%JFLG),N%(I%JCOS),BTEST(IBUG,0),N%(I%WK) )               STR00530
C                                                                       STR00540
C------- SET UP TEMPORARY ARRAY FOR MAIN DIAGONAL INDICIES...           STR00550
         I%MD =I%                                                       STR00560
         I%   =I%+NDOF+1                                               STR00570
         CALL GKTLIN(0,1,NDOF,IBUG,TITLE,N%(I%MD),                      STR00580
     &          IELDOF,N%(I%LM),%(I%KE),MD,'GLOBAL STIFFNESS')         STR00590
C                                                                       STR00600
C                                                                       STR00610
C------------------------                                               STR00620
C= INPUT MATERIAL DATA =                                                STR00630
C------------------------                                               STR00640
C     WRITE (6,5) TITLE(1),TITLE(2)                                     STR00650
      READ (5,*) NMAT                                                   STR00660
C     ---- MATERIAL DATA STORAGE --------------------                  STR00670
C          N%(I%MAT     )=        # OF MATERIAL PORPERTIES (NMAT)      STR00680
C          N%(I%MAT+    1)=        ABSOLUTE ADDRESS OF MATL #1         STR00690
C          N%(I%MAT+    2)=        ABSOLUTE ADDRESS OF MATL #2         STR00700
C          N%(I%MAT+    3)=        ABSOLUTE ADDRESS OF MATL #3         STR00710
C                  .                          .                       STR00720
C                  .                          .                       STR00730
C                  .                          .                       STR00740
C          N%(I%MAT+NMAT )=        ABSOLUTE ADDRESS FO MATL # NMAT     STR00750
C           %(N%(I%MAT+1))=        MATERIAL DATA FOR MATERIAL #1       STR00760
C                  .                          .                       STR00770
C                  .                          .                       STR00780
C                  .                          .                       STR00790
C           %(N%(I%MAT+2))=        MATERIAL DATA FOR MATERIAL #2       STR00800
C                  .                          .                       STR00810
C                  .                          .                       STR00820
C                  .                          .                       STR00830
C                                                                      STR00840
C                                                                      STR00850
      ICPT=1                                                           STR00860
      I%MAT =I%                                                        STR00870
      I%    =I% +NMAT+1                                                STR00880
      N%(I%MAT)=NMAT                                                   STR00890
      PT=.TRUE.                                                        STR00900
      FIRST=.TRUE.                                                     STR00910
      HEAD=.TRUE.                                                      STR00920
      MAT   =0                                                         STR00930
      LSTTYP=0                                                         STR00940
C------------- LOOP FOR EACH MATERIAL ----------------------------     STR00950
  110 MAT   =MAT   +1                                                  STR00960
          READ (5,*) TYPE                                              STR00970
          BACKSPACE (5)                                                STR00980
C                                                                      STR00990
          IF (TEST(TYPE,'3D-BEAM',.FALSE.)) THEN                       STR01000
              MATTYP= 1                                                STR01010
          ELSE IF (TEST(TYPE,'AXLMOD' ,.FALSE.)) THEN                  STR01020
```

```
            MATTYP= 2                                                   STR01030
         ELSE IF (TEST(TYPE,'BEND1'  ,.FALSE.)) THEN                    STR01040
            MATTYP= 3                                                   STR01050
         ELSE IF (TEST(TYPE,'SHEAR1' ,.FALSE.)) THEN                    STR01060
            MATTYP= 4                                                   STR01070
         ELSE IF (TEST(TYPE,'ROMBERG',.FALSE.)) THEN                    STR01080
            MATTYP= 5                                                   STR01090
         ELSE IF (TEST(TYPE,'TAKEDA' ,.FALSE.)) THEN                    STR01100
            MATTYP= 6                                                   STR01110
         ELSE IF (TEST(TYPE,'ELSPLS' ,.FALSE.)) THEN                    STR01120
            MATTYP= 7                                                   STR01130
         ELSE IF (TEST(TYPE,'BILINEAR',.FALSE.)) THEN                   STR01140
            MATTYP= 7                                                   STR01150
         ELSE IF (TEST(TYPE,'BRACE',.FALSE.)) THEN                      STR01160
            MATTYP= 8                                                   STR01170
         ELSE IF (TEST(TYPE,'LONG_ONJG',.FALSE.)) THEN                  STR01180
            MATTYP= 9                                                   STR01190
         ELSE IF (TEST(TYPE,'IA_BILN',.FALSE.)) THEN                    STR01200
            MATTYP= 10                                                  STR01210
         ELSE IF (TEST(TYPE,'SHORT_ONJG',.FALSE.)) THEN                 STR01220
            MATTYP= 11                                                  STR01230
         ELSE IF (TEST(TYPE,'STABILITY',.FALSE.)) THEN                  STR01240
            MATTYP= 12                                                  STR01250
         ELSE IF (TEST(TYPE,'LOCBKL',.FALSE.)) THEN                     STR01260
            MATTYP= 13                                                  STR01270
         ELSE IF (TEST(TYPE,'SHEAR2',.FALSE.)) THEN                     STR01280
            MATTYP= 14                                                  STR01290
C                                                                       STR01300
C.......... TO ADD ADDITIONAL MATERIALS, COPY THE NEXT TWO LINES        STR01310
C         ELSE IF (TEST(TYPE,'(1)',.FALSE.)) THEN                       STR01320
C            MATTYP= (4)                                                STR01330
C..                  CHANGE THE FOLLOWING VALUES                       STR01340
C..                  (1) YOUR MATERIAL NAME                            STR01350
C..                  (2) MATERIAL ID NUMBER = XX                       STR01360
C..                  ADD CALL MATXX TO SUBPROGRAM   MATLIB             STR01370
C..                  ADD SUBPROGRAM MATXX                              STR01380
C..........                                                             STR01390
         ELSE IF (TEST(TYPE,'END',.FALSE.)) THEN                        STR01400
            NMAT=MAT-1                                                  STR01410
            NZ(IZMAT)=NMAT                                              STR01420
            GO TO 120                                                   STR01430
         ELSE                                                           STR01440
            WRITE (6,*) 'INVALID MATERIAL TYPE: ',HEAD                  STR01450
            GO TO 110                                                   STR01460
         ENDIF                                                          STR01470
C                                                                       STR01480
C.......RESERVE STORAGE                                                STR01490
        NZ(IZMAT+MAT    )=IZ                                            STR01500
        IZWK=IZ+200                                                     STR01510
C                                                                       STR01520
C.......CALL MATERIAL LIBRARY TO INITIALIZE MATERIAL DATA               STR01530
        CALL MATLIB                                                     STR01540
     &  (IOPT  ,LSTTYP ,PT    ,IREL,HEAD,NAME,EESE,EPSE,DUCT,EXCR,      STR01550
     &   P     ,PL     ,D     ,DL    ,V     ,VL    ,LELEM ,LMAT  ,     STR01560
     &   MAT   ,MATTYP,SE,-1,DAMAGE)                                    STR01570
C                                                                       STR01580
C.......RESERVE STORAGE                                                 STR01590
        IZ=IZ+LMAT+1                                                    STR01600
C                                                                       STR01610
      IF (MAT   .LT.NMAT) GO TO 110                                     STR01620
  120 CONTINUE                                                          STR01630
C                                                                       STR01640
C=====================================                                  STR01650
C= INPUT GEOMETRIC STIFFNESS DATA ===                                   STR01660
C=====================================                                  STR01670
C   Z(IZKGDT+0) = VERTICAL GROUND ACCELERATION                          STR01680
C  NZ(IZKGDT+1) = TYPE OF AXIAL LOAD USED                               STR01690
C               = 0, GEOMETRIC STIFFNESS IS NOT CONSIDERED              STR01700
C               = 1, LOAD= F(INPUT)*(1.0 + ACCEL)                       STR01710
C               = 2, PREVIOUS LOAD STEP'S AXIAL LOAD IS USED            STR01720
C  NZ(IZKGDT+2) = TYPE OF ELEMENT GEOMETRIC STIFFNESS TO BE USED        STR01730
C               = 0, GEOMETRIC STIFFNESS IS NOT CONSIDERED              STR01740
C               = 1, LUMPED FORMULATION                                 STR01750
C               = 2, CONSISTENT FORMULATION                             STR01760
C  NZ(IZKGDT+3) = FORM OF SOLUTION                                      STR01770
C               = 0, GEOMETRIC STIFFNESS IS NOT CONSIDERED              STR01780
C               = 1, KG IS SUBTRACTED FROM ELEMENT STIFFNESS            STR01790
C               = 2, SEPERATE GLOBAL KG IS FORMED                       STR01800
C  KGCOND       = LOGICAL FLAG, TRUE IF KG IS TO BE CONDENSED           STR01810
C                                                                       STR01820
      IZKGDT=IZ                                                         STR01830
      IZ    =IZ +4                                                      STR01840
      READ (5,*)  (NZ(IZKGDT+I),I=1,3) ,KGCOND                         STR01850
C     WRITE(6,*) 'KGDATA',(NZ(IZKGDT+I),I=1,3)                          STR01860
      I=IZKGDT                                                          STR01870
      IF (NZ(I+1).EQ.0 .OR. NZ(I+2).EQ.0 .OR. NZ(I+3).EQ.0) THEN        STR01880
         NZ(I+1)=0                                                      STR01890
         NZ(I+2)=0                                                      STR01900
         NZ(I+3)=0                                                      STR01910
       ELSE                                                             STR01920
         WRITE (6,150) TITLE(1),TITLE(2)                                STR01930
         IF (NZ(I+1).EQ.1)  WRITE (6,160)                               STR01940
     &    'KGDATA(1)= 1, LOAD= F(INPUT)*(1.00+ACCEL)',                  STR01950
     &    '                 F(INPUT) IS POSITIVE IN COMPRESSION',       STR01960
     &    '                 ACCEL IS POSITIVE GLOBAL Z-AXIS ACCELERATION' STR01970
         IF (NZ(I+1).EQ.2)  WRITE (6,160)                               STR01980
     &       'KGDATA(1)= 2, PREVIOUS LOAD STEP'S AXIAL LOAD IS USED'    STR01990
C                                                                       STR02000
         IF (NZ(I+2).EQ.1)  WRITE (6,160)                               STR02010
     &       'KGDATA(2)= 1, LUMPED FORMULATION   '                      STR02020
         IF (NZ(I+2).EQ.2)  WRITE (6,160)                               STR02030
     &       'KGDATA(2)= 2, CONSISTENT FORMULATION '                    STR02040
C                                                                       STR02050
C                                                                       STR02060
         IF (NZ(I+3).EQ.1)  WRITE (6,160)                               STR02070
     &       'KGDATA(3)= 1, KG IS SUBTRACTED FROM ELEMENT STIFFNESS '   STR02080
         IF (NZ(I+3).EQ.2)  WRITE (6,160)                               STR02090
     &       'KGDATA(3)= 2, SEPERATE GLOBAL KG IS FORMED'               STR02100
C                                                                       STR02110
         WRITE (6,160) '  '                                             STR02120
         IF (    KGCOND)  WRITE (6,160)                                 STR02130
     &      '- THE GEOMETRIC STIFFNESS MATRIX IS CONDENSED '//          STR02140
     &                  'WITH THE STRUCTURAL STIFFNESS MATRIX.'         STR02150
         IF (.NOT.KGCOND)  WRITE (6,160)                                STR02160
     &      '- THE GEOMETRIC STIFFNESS MATRIX IS NOT CONDENSED '//      STR02170
     &                  'WITH THE STRUCTURAL STIFFNESS MATRIX.'         STR02180
       ENDIF                                                            STR02190
C                                                                       STR02200
  150 FORMAT ('1 STRUCTURE....: ',A,/                                   STR02210
     &        '  SOLUTION.....: ',A,//                                 STR02220
     &     5X,' GEOMETRIC STIFFNESS DATA '/                             STR02230
     &     5X,'--------------------------'/)                            STR02240
  160 FORMAT(5X,A)                                                      STR02250
C                                                                       STR02260
C                                                                       STR02270
C=================================================                      STR02280
C= SET UP STORAGE FOR GEOMETRIC STIFFNESS DATA ===                      STR02290
C=================================================                      STR02300
C      ---- GLOBAL STIFFNESS STORGE ------------------                  STR02310
C           NZ(IZMDKG) = INDICIES OF THE MAIN DIAGONAL TERMS            STR02320
C             Z(IZKG  ) = GLOBAL GEOMETRIC STIFFNESS MATRIX             STR02330
      IF (NZ(IZKGDT+3).EQ.2) THEN                                       STR02340
         IZMDKG=IZ                                                      STR02350
         IZ    =IZ +NDOF + 1                                            STR02360
         CALL SKYLIN(0,1,NDOF,IBUG,TITLE,NZ(IZMDKG),                    STR02370
     &   KGDOF,NZ(IZLMKG),Z(IZKEKG),MD,'GLOBAL GEOMETRIC STIFFNESS')    STR02380
       ELSE                                                             STR02390
C        DUMMY LOCATION FOR KG TO AVOID ADDRESSING EXCEPTIONS           STR02400
C        KG DATA IS NOT STORED IN THESE LOCATIONS...                    STR02410
         IZMDKG=IZKGDT                                                  STR02420
         IZKG  =IZKGDT                                                  STR02430
      ENDIF                                                             STR02440
C                                                                       STR02450
C-----------------------                                                STR02460
C= INPUT ELEMENT DATA ==                                                STR02470
C-----------------------                                                STR02480
      READ (5,*) NELMT                                                  STR02490
C     ---- ELEMENT STORAGE ----------------------                       STR02500
C          NZ(IZELE     )=        # OF MATERIAL PORPERTIES (NELMT)     STR02510
C          NZ(IZELE+    1)=       ABSOLUTE ADDRESS OF ELEMENT #1       STR02520
C          NZ(IZELE+    2)=       ABSOLUTE ADDRESS OF ELEMENT #2       STR02530
C          NZ(IZELE+    3)=       ABSOLUTE ADDRESS OF ELEMENT #3       STR02540
C                  .                           .                      STR02550
C                  .                           .                      STR02560
C                  .                           .                      STR02570
C          NZ(IZMAT+NELMT)=      ABSOLUTE ADDRESS FOR ELEM # NELMT     STR02580
C           Z(NZ(IZELE+1))=      MATERIAL DATA FOR ELEMENT #1          STR02590
C                  .                           .                      STR02600
C                  .                           .                      STR02610
C                  .                           .                      STR02620
C           Z(NZ(IZELE+2))=      MATERIAL DATA FOR ELEMENT #1          STR02630
C                  .                           .                      STR02640
C                  .                           .                      STR02650
C                  .                           .                      STR02660
C                                                                     STR02670
      IOPT=1                                                            STR02680
      IZELE =IZ                                                         STR02690
      IZ    =IZ +NELMT+1                                                STR02700
      NZ(IZELE)=NELMT                                                   STR02710
      FIRST=.TRUE.                                                      STR02720
      HEAD=.TRUE.                                                       STR02730
      IGEN=0                                                            STR02740
      IELNO=0                                                           STR02750
      LSTTYP=0                                                          STR02760
C------------ LOOP FOR EACH ELEMENT -------------------------------     STR02770
  210 IELNO=IELNO+1                                                     STR02780
        IF (IGEN.LE.0) THEN                                             STR02790
          READ (5,*) TYPE                                               STR02800
C                                                                       STR02810
          IF (TEST(TYPE,'3D-BEAM',.FALSE.)) THEN                        STR02820
             NINPUT= 10                                                 STR02830
             NIN%  = 437                                                STR02840
             NELTYP= 1                                                  STR02850
C                                                                       STR02860
          ELSE IF (TEST(TYPE,'SPRING',.FALSE.)) THEN                    STR02870
             NINPUT= 10                                                 STR02880
             NIN%  = 133                                                STR02890
             NELTYP= 2                                                  STR02900
C                                                                       STR02910
          ELSE IF (TEST(TYPE,'SHEAR WALL',.FALSE.)) THEN                STR02920
             NINPUT= 9                                                  STR02930
             NIN%  = 769                                                STR02940
             NELTYP= 3                                                  STR02950
C                                                                       STR02960
          ELSE IF (TEST(TYPE,'BRACE',.FALSE.)) THEN                     STR02970
             NINPUT= 10                                                 STR02980
             NIN%  = 133                                                STR02990
             NELTYP= 8                                                  STR03000
C                                                                       STR03010
          ELSE IF (TEST(TYPE,'IE3DBEAM',.FALSE.)) THEN                  STR03020
             NINPUT= 14                                                 STR03030
             NIN%  = 614                                                STR03040
             NELTYP= 9                                                  STR03050
C                                                                       STR03060
          ELSE IF (TEST(TYPE,'STABILITY',.FALSE.)) THEN                 STR03070
             NINPUT= 8                                                  STR03080
             NIN%  = 618                                                STR03090
             NELTYP= 12                                                 STR03100
          ELSE IF (TEST(TYPE,'SHEAR_OPEN',.FALSE.)) THEN                STR03110
             NINPUT= 9                                                  STR03120
             NIN%  = 775                                                STR03130
             NELTYP= 14                                                 STR03140
C.......... TO ADD ADDITIONAL ELEMENTS, COPY THE NEXT FOUR LINES        STR03150
          ELSE IF (TEST(TYPE,'(1)',.FALSE.)) THEN                       STR03160
             NINPUT= (2)                                                STR03170
             NIN%  = (3)                                                STR03180
             NELTYP= (4)                                                STR03190
C..                    CHANGE THE FOLLOWING VALUES                     STR03200
C..                    (1) YOUR ELEMENT NAME                           STR03210
C..                    (2) # OF VALUES TO BE INPUT                     STR03220
C..                    (3) # OF STORAGE LOCATIONS REQD FOR ELEMENT     STR03230
C..                    (4) ELEMENT ID NUMBER = XX                      STR03240
C..                    ADD CALL ELEXX TO SUBPROBRAM   ELE_LIB          STR03250
C..                    ADD SUBPROGRAM ELEXX                            STR03260
C............                                                           STR03270
          ELSE IF (TEST(TYPE,'END',.FALSE.)) THEN                       STR03280
             NELMT=IELNO-1                                              STR03290
             NZ(IZELE)=NELMT                                            STR03300
             GO TO 300                                                  STR03310
          ELSE                                                          STR03320
             WRITE (6,*) 'INVALID ELEMENT TYPE: ',HEAD                  STR03330
             GO TO 210                                                  STR03340
          ENDIF                                                         STR03350
          BACKSPACE (5)                                                 STR03360
          READ (5,*) TYPE,NAME,       ( RINPUT(I),I=1,NINPUT),IGEN      STR03370
          IF (IGEN.GT.0)  READ (5,*)  ( GINPUT(I),I=1,NINPUT)          STR03380
        ELSE                                                            STR03390
          IGEN=IGEN-1                                                   STR03400
          NAME='GENERATED'                                              STR03410
          DO 220 I=1,NINPUT                                             STR03420
  220       RINPUT(I)=RINPUT(I)+GINPUT(I)                               STR03430
        ENDIF                                                           STR03440
C     WRITE (6,*) 'BEFORE STORAGE IS RESERVED IN STRUCT'                STR03450
CDBG    WRITE(6,111) (I,NZ(I),Z(I),I=68,91)                             STR03460
  111 FORMAT (OP,' ZMATRX(',I6,'):',I15,1P,G20.10)                      STR03470
C.......RESERVE STORAGE                                                 STR03480
        NZ(IZELE+IELNO)=IZ                                              STR03490
        IC=IZ                                                           STR03500
        IZWK=IZ                                                         STR03510
C                                                                       STR03520
C.......STORE ELEMENT TYPE                                              STR03530
        NZ(IC) =NELTYP                                                  STR03540
C                                                                       STR03550
C.......CALL ELEMENT LIBRARY TO DETERMINE THE STORAGE FOR MATL DATA     STR03560
        IOPT=9                                                          STR03570
        HEAD2=.FALSE.                                                   STR03580
        LSTTP2=0                                                        STR03590
        CALL ELELIB                                                     STR03600
     &  (IOPT  ,LSTTP2 ,.FALSE.,IREL,HEAD2, NAME , EESE,EPSE,DAMAGE,   STR03610
     &   DUCFAG,                                                        STR03620
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)    STR03630
C                                                                       STR03640
C.......RESERVE STORAGE                                                 STR03650
        IZ=IZ+NINZ+MSTOR                                                STR03660
        IZWK=IZ                                                         STR03670
C                                                                       STR03680
C.......DU_MP ELEMENT DATA IF DEBU_GING                                 STR03690
        IF (BTEST(IBUG,8)) THEN                                         STR03700
         WRITE (6,*) 'IZLM=',IZLM,' IZKE=',IZKE                         STR03710
         JJ=-1                                                          STR03720
         DO 7100 I=IC,(IZWK-1-IREL)                                     STR03730
          JJ=JJ+1                                                       STR03740
 7100     IF (NZ(I).NE.0) WRITE (6,7010) IELNO,JJ,I,NZ(I),Z(I)          STR03750
        ENDIF                                                           STR03760
C                                                                       STR03770
C.......CALL ELEMENT LIBRARY TO INITIALIZE ELEMENT DATA AND RETURN      STR03780
C.......INFO ON ELEMENT DOF...                                         STR03790
        IOPT=1                                                          STR03800
        CALL ELELIB                                                     STR03810
     &  (IOPT ,LSTTYP ,.TRUE.,IREL,HEAD,NAME , EESE,EPSE,DAMAGE,       STR03820
     &   DUCFAG,                                                        STR03830
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)    STR03840
C                                                                       STR03850
C.......RELEASE UNUSED STORAGE .....                                   STR03860
```

```
        I3=I3-IREL                                                      STR03870
C                                                                       STR03880
C.......DU_MP ELEMENT DATA IF DEBU_GING                                 STR03890
        IF (BTEST(IBUG,8)) THEN                                         STR03900
         WRITE (6,*) 'IZLM:',IZLM,' IZKE:',IZKE                         STR03910
         JJ=-1                                                          STR03920
         DO 7000 I=IC,(I3WK-1-IREL)                                     STR03930
          JJ=JJ+1                                                       STR03940
 7000     IF (N3(I).NE.0) WRITE (6,7010) IELNO,JJ,I,N3(I),Z(I)          STR03950
 7010    FORMAT (' IELNO:',I3,'  #',I6,'    Z(',I6,')=',I10,1P,G18.8)   STR03960
        ENDIF                                                           STR03970
C                                                                       STR03980
C--------- RESERVE SPACE IN THE SKYLINE FOR THE STIFFNESS               STR03990
        JFLAG=0                                                         STR04000
        IF( TEST(TYPE,'STABILITY ', .FALSE. )) JFLAG=1                  STR04010
        CALL SKYLIN(JFLAG,2,NDOF,IBUG,TITLE,N2(IZMD),                   STR04020
     &            IELDOF,N3(IZLM),Z(IZKE),MD,'GLOBAL STIFFNESS')        STR04030
C                                                                       STR04040
C--------- RESERVE SPACE IN THE SKYLINE FOR THE GEOMETRIC STIFFNESS     STR04050
      IF (N2(IZKGDT+3).EQ.1 .AND. AXIAL) THEN                           STR04060
        CALL SKYLIN(0,2,NDOF,IBUG,TITLE,N2(IZMD),                       STR04070
     &   KGDOF,N2(IZLMKG),Z(IZKEKG),MD,'GLOBAL GEOMETRIC STIFFNESS')    STR04080
      ELSE IF (N2(IZKGDT+3).EQ.2 .AND. AXIAL) THEN                      STR04090
        CALL SKYLIN(0,2,NDOF,IBUG,TITLE,N2(IZMDKG),                     STR04100
     &   KGDOF,N2(IZLMKG),Z(IZKEKG),MD,'GLOBAL GEOMETRIC STIFFNESS')    STR04110
      ENDIF                                                             STR04120
C     WRITE (6,*) 'AFTER SKYLIN IN STRUCT'                              STR04130
CDBG     WRITE(6,111) (I,N2(I),Z(I),I=68,91)                            STR04140
C                                                                       STR04150
      IF (IELNO.LT.NELMT) GO TO 210                                     STR04160
  300 CONTINUE                                                          STR04170
C                                                                       STR04180
C                                                                       STR04190
C*******************************************************              STR04200
C* MODIFY GEOMETRIC STIFFNESS MATRIX FOR CONDENSATION  ***              STR04210
C*******************************************************              STR04220
C THE CONDENSATION PROCESS MAY EXPAND THE GEOMETRIC STIFFNESS MATRIX'S  STR04230
C SKYLINE.  THE MAXIMUN SKYLINE OF BOTH THE GEOMETRIC STIFFNESS AND     STR04240
C THE STIFFNESS MATRIX IS USED.                                         STR04250
      IF (N2(IZKGDT+3).EQ.2 .AND. KGCOND .AND. NCOND.GT.0)THEN          STR04260
        DO 310 I=0,NDOF                                                 STR04270
          J=I+1                                                         STR04280
          MDKG1=N2(IZMDKG+I)                                            STR04290
C         N2(IZMDKG+I)=MAX( N2(IZMDKG+I),N2(IZMD+I) )                   STR04300
          N2(IZMDKG+I)=MIN( N2(IZMDKG+I),N2(IZMD+I) )                   STR04310
CCC       WRITE (6,*) 'J,MDKG1,MD,MDKG',J,MDKG1,N2(IZMD+I),N2(IZMDKG+I) STR04320
  310     CONTINUE                                                      STR04330
      ENDIF                                                             STR04340
C                                                                       STR04350
C                                                                       STR04360
C*************************************                                  STR04370
C* SET UP GLOBAL STIFFNESS STORAGE **                                   STR04380
C*************************************                                  STR04390
C             N2(IZMD)   = INDICIES OF THE MAIN DIAGONAL TERMS          STR04400
C              Z(IZSTIF) = GLOBAL STIFFNESS MATRIX                      STR04410
C                                                                       STR04420
      CALL SKYLIN(0,3,NDOF,IBUG,TITLE,N2(IZMD),                         STR04430
     &            IELDOF,N2(IZLM),Z(IZKE),MD,'GLOBAL STIFFNESS')        STR04440
      IZSTIF=IZ                                                         STR04450
      IZ     = IZ+N2(IZMD+NDOF) +1                                      STR04460
C                                                                       STR04470
C*************************************************                      STR04480
C* SET UP STORAGE FOR GEOMETRIC STIFFNESS DATA ***                      STR04490
C*************************************************                      STR04500
C             N2(IZMDKG) = INDICIES OF THE MAIN DIAGONAL TERMS          STR04510
C              Z(IZKG  ) = GLOBAL GEOMETRIC STIFFNESS MATRIX            STR04520
      IF (N2(IZKGDT+3).EQ.2) THEN                                       STR04530
        CALL SKYLIN(0,3,NDOF,IBUG,TITLE,N2(IZMDKG),                     STR04540
     &   KGDOF,N2(IZLMKG),Z(IZKEKG),MD,'GLOBAL GEOMETRIC STIFFNESS')    STR04550
        IZKG  =IZ                                                       STR04560
        IZ    =IZ +N2(IZMDKG+NDOF)                                      STR04570
      ENDIF                                                             STR04580
C                                                                       STR04590
C                                                                       STR04600
C------------------------                                               STR04610
C* INPUT MASS MATRIX ***                                                STR04620
C------------------------                                               STR04630
C     NMASS = NUMBER OF NODAL MASSES TO BE INPUT                        STR04640
C     FMASS = 0, NO MASS MATRIX                                         STR04650
C             1, MASS MATRIX DUE TO NODE MASSES ONLY                    STR04660
C     MCOND = LOGICAL FLAG, TRUE IF MASS IS TO BE CONDENSED...          STR04670
C                                                                       STR04680
      READ (5,*) NMASS,FMASS,MCOND                                      STR04690
C                                                                       STR04700
      CALL MASSIN(NMASS,FMASS,MCOND,BUG,IBUG,TITLE,MAXI,MAXD1,          STR04710
     &    MAXNOD,NCOS,NNODE,I1,IW,I3ID,I,N1,D1,                         STR04720
     &    I2MDM5,I2MASS,IZLM,I3KE,I3MDAT,                               STR04730
     &    NDOF,NMASS,I2COS,I3IDOF,I2CNST,I3JFLG,I2JCOS,'GLOBAL MASS',   STR04740
     &    NCOND,NFREE,I2MD)                                             STR04750
      IF (NMASS.GT.0) THEN                                              STR04760
        WRITE (6,160) ' '                                               STR04770
        IF (MCOND) THEN                                                 STR04780
          WRITE (6,160) '- THE MASS MATRIX IS ''//                      STR04790
     &      'CONDENSED WITH THE STRUCTURAL STIFFNESS MATRIX.'           STR04800
        ELSE                                                            STR04810
          WRITE (6,160) '- THE MASS MATRIX IS NOT ''//                  STR04820
     &      'CONDENSED WITH THE STRUCTURAL STIFFNESS MATRIX.'           STR04830
        ENDIF                                                           STR04840
      ELSE                                                              STR04850
        WRITE (6,499)                                                   STR04860
  499   FORMAT(//5X,' ZERO MASS MATRIX '/                               STR04870
     &         5X,'------------------')                                 STR04880
      ENDIF                                                             STR04890
C                                                                       STR04900
C------------------------                                               STR04910
C* INPUT DAMPING DATA ***                                               STR04920
C------------------------                                               STR04930
C     NDAMP = NUMBER OF NODAL DAMPING COIEF TO BE INPUT =0              STR04940
C     FDAMP = 0, NO DAMPING MATRIX                                      STR04950
C             1, ALPHA AND BETA INPUT                                   STR04960
C                                                                       STR04970
      NDAMP=0                                                           STR04980
      FDAMP=1                                                           STR04990
C     READ (5,*) NDAMP,FDAMP                                            STR05000
      IF (FDAMP.EQ.1) THEN                                              STR05010
        I3DAMP=I3                                                       STR05020
        I3=I3 + 2                                                       STR05030
        READ (5,*) ALPHA,BETA                                           STR05040
        Z(IZDAMP)   =ALPHA                                              STR05050
        Z(IZDAMP+1) =BETA                                               STR05060
        WRITE(6,500) ALPHA,BETA                                         STR05070
      ENDIF                                                             STR05080
  500 FORMAT(//5X,' PROPORTIONAL DAMPING COIEFICIENTS'/                 STR05090
     &         5X,'-----------------------------------'/,               STR05100
     &         5X,'ALPHA=',1P,G13.5,'   BETA=',G13.5/)                  STR05110
C                                                                       STR05120
C-----------------------------------                                    STR05130
C* ZERO LOAD AND DISPL INDICIES ***                                     STR05140
C-----------------------------------                                    STR05150
      IZLOAD=0                                                          STR05160
      IZDISP=0                                                          STR05170
      IZVEL =0                                                          STR05180
      IZACC =0                                                          STR05190
      IZDLOA=0                                                          STR05200
      IZDDSP=0                                                          STR05210
      IZDVEL=0                                                          STR05220
      IZDACC=0                                                          STR05230
C                                                                       STR05240
      RETURN                                                            STR05250
      END                                                               STR05260
C*********************************************************************  TES00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE                                 TES00020
C         END DEBUG                                                     TES00030
      LOGICAL FUNCTION TEST(OPTION,TARGET,DEFLT)                        TES00040
      CHARACTER*(*) OPTION,TARGET                                       TES00050
      LOGICAL DEFLT,BTEST                                               TES00060
         J= INDEX(OPTION,TARGET)                                        TES00070
         IF (DEFLT) THEN                                                TES00080
              IF (J.LT.2) THEN                                          TES00090
                   TEST=.TRUE.                                          TES00100
              ELSE IF (OPTION((J-2):(J-1)).NE.'NO') THEN                TES00110
                   TEST=.TRUE.                                          TES00120
              ELSE                                                      TES00130
                   TEST=.FALSE.                                         TES00140
              ENDIF                                                     TES00150
         ELSE                                                           TES00160
              IF (J.EQ.0) THEN                                          TES00170
                   TEST=.FALSE.                                         TES00180
              ELSE IF (J.LT.2 ) THEN                                    TES00190
                   TEST=.TRUE.                                          TES00200
              ELSE IF (OPTION((J-2):(J-1)).EQ.'NO') THEN                TES00210
                   TEST=.FALSE.                                         TES00220
              ELSE                                                      TES00230
                   TEST=.TRUE.                                          TES00240
              ENDIF                                                     TES00250
         ENDIF                                                          TES00260
      RETURN                                                            TES00270
      END                                                               TES00280
C*********************************************************************  TMP00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT                            TMP00020
C         END DEBUG                                                     TMP00030
C*********************************************************************  TMP00040
C   TRIANGLAR MATRIX MULTIPLICATION ....                                TMP00050
C   MATRIX A IS AN UPPER TRIANGULAR MATRIX, WITH MAIN DIAGONAL ADDRESS  TMP00060
C   MATRIX MD.  X IS A VECTOR.  P=AX                                    TMP00070
C   THE MULTIPLICATION IS CARRIED OUT FROM ROW IR1 TO IR2  AND FROM     TMP00080
C   COLUMN JC1 TO JC2.                                                  TMP00090
C*********************************************************************  TMP00100
      SUBROUTINE TMPLY(PRINT,ZERO,IMD,N,IR1,IR2,JC1,JC2,MD,A,P,X)       TMP00110
      DIMENSION A(IMD),X(N),P(N),MD(N+1)                                TMP00120
      LOGICAL BTEST,PRINT,ZERO                                          TMP00130
C                                                                       TMP00140
      IJI(I,J)=MD(J)+J-I                                                TMP00150
      IF (PRINT) THEN                                                   TMP00160
        WRITE (6,500) 'TRIANGULAR MULTIPLACATION',                      TMP00170
     &                IR1,IR2,JC1,JC2,N                                 TMP00180
        CALL LMATRX(A,MD,N,MD(N+1),'MATRIX A ')                         TMP00190
C       CALL WMATRX(X,N ,1,'MATRIX X')                                  TMP00200
      ENDIF                                                             TMP00210
C       ---- ZERO P ----                                                TMP00220
      IF (ZERO) THEN                                                    TMP00230
        DO 320 J=IR1,IR2                                                TMP00240
  320     P(J)=0                                                        TMP00250
      ENDIF                                                             TMP00260
C---- OLD METHOD OF MULTIPLICATION, TOO SLOW                            TMP00270
C       DO 140 I=IR1,IR2                                                TMP00280
C       DO 140 K=JC1,JC2                                                TMP00290
C         IF (I.LE.K) THEN                                              TMP00300
C           IK=MD(K)+K-I                                                TMP00310
C           MDK=MD(K+1)                                                 TMP00320
C         ELSE                                                          TMP00330
C           IK=MD(I)+I-K                                                TMP00340
C           MDK=MD(I+1)                                                 TMP00350
C         ENDIF                                                         TMP00360
C         IF (IK.GE.MDK)  P(I)=P(I)+A(IK)*X(K)                          TMP00370
C 140   CONTINUE                                                        TMP00380
C       IF (PRINT) CALL WMATRX(P,N ,1,'RESULT = A*X ')                  TMP00390
C                                                                       TMP00400
C--------- MULTIPLY TERMS ABOVE AND INCLUDING THE MAIN DIAGONAL         TMP00410
        DO 20 J=JC1,JC2                                                 TMP00420
          IRC=J-MD(J+1)+MD(J)+1                                         TMP00430
          MDJ=MD(J)+J                                                   TMP00440
          I1=MAX(IRC,IR1)                                               TMP00450
          I2=MIN(J,IR2)                                                 TMP00460
          IF (I2.GE.I1) THEN                                            TMP00470
            DO 10 I=I1,I2                                               TMP00480
   10         P(I)=P(I)+A(MDJ-I)*X(J)                                  TMP00490
          ENDIF                                                         TMP00500
   20   CONTINUE                                                       TMP00510
C                                                                       TMP00520
C--------- MULTIPLY TERMS BELOW THE MAIN DIAGONAL                       TMP00530
        DO 30 I=IR1,IR2                                                 TMP00540
          IRC=I-MD(I+1)+MD(I)+1                                         TMP00550
          MDI=MD(I)+I                                                   TMP00560
          J1=MAX(IRC,JC1)                                               TMP00570
          J2=MIN(I-1,JC2)                                               TMP00580
          IF (J2.GE.J1) THEN                                            TMP00590
            DO 40 J=J1,J2                                               TMP00600
   40         P(I)=P(I)+A(MDI-J)*X(J)                                  TMP00610
          ENDIF                                                         TMP00620
   30   CONTINUE                                                       TMP00630
C                                                                       TMP00640
CC      IF (PRINT) CALL WMATRX(P,N ,1,'RESULT = A*X ')                  TMP00650
C                                                                       TMP00660
  500 FORMAT (/2X,A/,2X,' IR1:',I5,' IR2:',I5,                          TMP00670
     &        ' JC1:',I5,' JC2:',I5,' N:',I5)                           TMP00680
      IF (PRINT) THEN                                                   TMP00690
        DO 515 I=IR1,IR2                                                TMP00700
  515     WRITE(6,520) I,X(I),P(I)                                      TMP00710
        WRITE (6,*) ' '                                                 TMP00720
  520 FORMAT (2X,'DOF:',I6,'  X(DOF)',1P,G14.5,' A*X=P(DOF):',G14.5)    TMP00730
      ENDIF                                                             TMP00740
      RETURN                                                            TMP00750
      END                                                               TMP00760
C$PROCESS SDUMP OPT(0) GOSTMT XREF MAP                                  TMP00770
C*********************************************************************  UTM00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE,INIT                            UTM00020
C         END DEBUG                                                     UTM00030
C*********************************************************************  UTM00040
C   UPPER TRIANGLAR MATRIX MULTIPLICATION ....                          UTM00050
C   MATRIX A IS AN UPPER TRIANGULAR MATRIX, WITH MAIN DIAGONAL ADDRESS  UTM00060
C   MATRUX MD.  X IS A VECTOR.  P=AX                                    UTM00070
C   THE MULTIPLICATION IS CARRIED OUT FROM ROW L1 TO L2  AND FROM       UTM00080
C  COLUMN L1 TO N.                                                      UTM00090
C*********************************************************************  UTM00100
      SUBROUTINE UTMPLY(PRINT,ZERO,IMD,N,IR1,IR2,JC1,JC2,MD,A,P,X)      UTM00110
C     SUBROUTINE UTMPLY(PRINT,ZERO,N,IMD,L1,L2,MD,A,P,X)                UTM00120
      DIMENSION A(IMD),X(N),P(N),MD(N+1)                                UTM00130
      LOGICAL BTEST,PRINT,ZERO                                          UTM00140
C                                                                       UTM00150
      IJI(I,J)=MD(J)+J-I                                                UTM00160
      IF (PRINT) THEN                                                   UTM00170
        WRITE (6,500) 'UPPER TRIANGULAR MULTIPLACATION',                UTM00180
     &                IR1,IR2,JC1,JC2,N                                 UTM00190
        CALL LMATRX(A,MD,N,MD(N+1),'MATRIX A ')                         UTM00200
C       CALL WMATRX(X,N ,1,'MATRIX X')                                  UTM00210
      ENDIF                                                             UTM00220
C       ---- ZERO P ----                                                UTM00230
      IF (ZERO) THEN                                                    UTM00240
        DO 320 J=L1,L2                                                  UTM00250
  320     P(J)=0                                                        UTM00260
      ENDIF                                                             UTM00270
C                                                                       UTM00280
C-------- MULTIPLY TERMS ABOVE AND INCLUDING THE MAIN DIAGONAL          UTM00290
        DO 20 J=JC1,JC2                                                 UTM00300
          IRC=J-MD(J+1)+MD(J)+1                                         UTM00310
          MDJ=MD(J)+J                                                   UTM00320
          I1=MAX(IRC,IR1)                                               UTM00330
          I2=MIN(J,IR2)                                                 UTM00340
          IF (I2.GE.I1) THEN                                            UTM00350
            DO 10 I=I1,I2                                               UTM00360
   10         P(I)=P(I)+A(MDJ-I)*X(J)                                  UTM00370
          ENDIF                                                         UTM00380
                                                                        UTM00390
```

```
   20  CONTINUE                                                          UTM00400
C                                                                        UTM00410
C------------ MULTIPLY S_RED * DISP = P                                  UTM00420
C       DO 340 J=L1,N                                                    UTM00430
C         IC1=J-(MD(J+1)-MD(J)) +1                                       UTM00440
C         IC2=MAX(IC1,L1)                                                UTM00450
C         DO 340 I=J,IC2,-1                                              UTM00460
C           IJ=IJI(I,J)                                                  UTM00470
C 340     P(I)=P(I)+A(IJ)*X(J)                                           UTM00480
C                                                                        UTM00490
      IF (PRINT) CALL WMATRX(P,N ,1,'RESULT = A*X ')                     UTM00500
C                                                                        UTM00510
  500 FORMAT (/2X,A/,2X,' IR1=',I5,' IR2=',I5,                           UTM00520
     &                   ' JC1=',I5,' JC2=',I5,' N=',I5)                 UTM00530
      IF (PRINT) THEN                                                    UTM00540
        DO 515 I=IR1,IR2                                                 UTM00550
  515     WRITE(6,520) I,X(I),P(I)                                       UTM00560
          WRITE(6,*)  ' '                                                UTM00570
  520 FORMAT (2X,'DOF=',I6,'  X(DOF)',1P,G14.5,'  A*X=P(DOF)=',G14.5)    UTM00580
      ENDIF                                                              UTM00590
      RETURN                                                             UTM00600
      END                                                                UTM00610
C=======================================================================VCO00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE                                  VCO00020
C         END DEBUG                                                      VCO00030
      REAL FUNCTION VCOS(V1,V2)                                          VCO00040
      DIMENSION V1(3),V2(3)                                              VCO00050
      VALUE (A1,A2,A3)=SQRT(A1**2+A2**2+A3**2)                           VCO00060
      DOT   (A1,B1,C1,A2,B2,C2)=A1*A2+B1*B2+C1*C2                        VCO00070
C                                                                        VCO00080
      AV1=VALUE(V1(1),V1(2),V1(3))                                       VCO00090
      AV2=VALUE(V2(1),V2(2),V2(3))                                       VCO00100
      VDOT=DOT(V1(1),V1(2),V1(3) ,V2(1),V2(2),V2(3))                     VCO00110
      VCOS=VDOT/(AV1*AV2)                                                VCO00120
      RETURN                                                             VCO00130
      END                                                                VCO00140
C=======================================================================VCR00010
C     V3 = V1 X V2,  VECTOR CROSS PRODUCT ...                            VCR00020
C=======================================================================VCR00030
      SUBROUTINE  VCROSS(V3,V1,V2)                                       VCR00040
      DIMENSION V1(3),V2(3),V3(3)                                        VCR00050
C                                                                        VCR00060
      V3(1) = V1(2)*V2(3) - V1(3)*V2(2)                                  VCR00070
      V3(2) = -(V1(1)*V2(3) - V1(3)*V2(1) )                              VCR00080
      V3(3) = V1(1)*V2(2) - V1(2)*V2(1)                                  VCR00090
C                                                                        VCR00100
      RETURN                                                             VCR00110
      END                                                                VCR00120
C=======================================================================WIL00010
C         DEBUG UNIT(6),SUBCHK,SUBTRACE                                  WIL00020
C         END DEBUG                                                      WIL00030
      SUBROUTINE WILSON(IOPT,BUG,DT,TO,TF,T,THETA,NEWKDY,                WIL00040
     &    L1,L2,L3,L4,L5,L6,IMDMAS,IMD,IMDS,IMDKG,NDOF,NFREE,            WIL00050
     &    KGLOAD,KGFORM,                                                 WIL00060
     &    A     ,V     ,D     ,P     ,WORK  ,                           WIL00070
     &    DA    ,DV    ,DD    ,DP    ,                                  WIL00080
     &    MASS  ,MDMASS,DAMP  ,STIFF ,MD    ,                           WIL00090
     &    KG    ,MDKG  ,ACC   ,GRAV  ,                                   WIL00100
     &    S     ,MDS   ,Q     ,R     ,PBAR  ,                           WIL00110
     &    FDAMP,FMASS,C1,C2,C3,C4,C5,C6,C7,ALPHA,BETA,C10,C11,ABORT)     WIL00120
C                                                                        WIL00130
      COMMON /IDSTEP/ISTEP                                               WIL00140
      INTEGER FDAMP,FMASS                                                WIL00150
      REAL MASS,KG                                                       WIL00160
      LOGICAL BUG,NEWKDT,BTEST,ABORT                                     WIL00170
      DIMENSION A(NDOF),V(NDOF),D(NDOF),DA(NDOF),DV(NDOF),DD(NDOF)       WIL00180
      DIMENSION P(NDOF),DP(NDOF),DUMMY(1),IDUMMY(1),WORK(2*NDOF)         WIL00190
      DIMENSION PBAR(L3:L4),Q(L3:L4),R(L3:L4)                            WIL00200
      DIMENSION MDMASS(NDOF+1),MASS(IMDMAS)                              WIL00210
      DIMENSION MDKG(NDOF+1)   ,KG(IMDKG)                                WIL00220
      DIMENSION MD(NDOF+1)     ,STIFF(IMD)                               WIL00230
      DIMENSION MDS(NDOF+1)    ,S(IMDS)     ,DAMP(3)                     WIL00240
C                                                                        WIL00250
      GO TO (100,200,300) IOPT                                           WIL00260
C                                                                        WIL00270
C----------------------                                                  WIL00280
C- INITIALIZE VALUES ---                                                 WIL00290
C----------------------                                                  WIL00300
  100 CONTINUE                                                           WIL00310
      C1=6./(THETA*DT)**2                                                WIL00320
      C2=3./(THETA*DT)                                                   WIL00330
      C3=6./(THETA*DT)                                                   WIL00340
      C4=THETA*DT/2.                                                     WIL00350
      C5=DT/2.                                                           WIL00360
      C6=DT**2  /2.                                                      WIL00370
      C7=DT**2  /6.                                                      WIL00380
      ALPHA=DAMP(1)                                                      WIL00390
      BETA =DAMP(2)                                                      WIL00400
      C10=1+BETA*C2                                                      WIL00410
      C11=C1+ALPHA*C2                                                    WIL00420
C                                                                        WIL00430
C------- DEVELOP SKYLINE OF DYNAMIC STIFFNESS S                          WIL00440
C....... LSTIFF = # OF STIFFNESS TERMS IN COLUMN J                       WIL00450
C....... LMASS  = # OF MASS TERMS IN COLUMN J                            WIL00460
C....... LKG    = # OF KG TERMS IN COLUMN J                              WIL00470
C                                                                        WIL00480
C     MDS(1)=1                                                           WIL00490
C     LKG=0                                                              WIL00500
C     DO 110 J1=2,NFREE+1                                                WIL00510
C       J=J1+L3-1                                                        WIL00520
C       LSTIFF=MD(J)-MD(J-1)                                             WIL00530
C       LMASS =MDMASS(J)-MDMASS(J-1)                                     WIL00540
C       IF (KGFORM.EQ.2)  LKG = MDKG(J)-MDKG(J-1)                        WIL00550
C       L=MIN(J1-1,MAX(LSTIFF,LMASS,LKG))                                WIL00560
C 110   MDS(J1)=MDS(J1-1)+L                                             WIL00570
C                                                                        WIL00580
      RETURN                                                             WIL00590
C                                                                        WIL00600
C------------------------------------------------------------------------WIL00610
C- SOLVE FOR INCRIMENTAL DISPLACEMENTS, VELOCITIES AND ACCELERATIONS --  WIL00620
C------------------------------------------------------------------------WIL00630
  200 CONTINUE                                                           WIL00640
      IF (BUG ) THEN                                                     WIL00650
        DO 205 I=L3,L4                                                   WIL00660
  205   WRITE (6,206) I,D(I),V(I),A(I),P(I),DP(I)                        WIL00670
  206   FORMAT (' DOF=',I6,'  DISP=',1P,G12.4,'  VEL=',G12.4,            WIL00680
     &          '  ACC=',G12.4,'  LOAD=',   G12.4,'   DP=',G12.4)       WIL00690
      ENDIF                                                              WIL00700
C                                                                        WIL00710
C------- FORM DYNAMIC STIFFNESS                                          WIL00720
      IF (FDAMP.EQ.1) THEN                                               WIL00730
C                                                                        WIL00740
C....... SOME LOCAL VARIABLES                                            WIL00750
C J     = COLUMN NUMBER FOR MASS AND STIFF                               WIL00760
C I     = ROW NUMBER FOR MASS AND STIFF                                  WIL00770
C J1    = COLUMN NUMBER FOR S                                            WIL00780
C I1    = ROW NUMBER FOR S                                               WIL00790
C IJSTIF = ADDRESS OF ELEMENT I,J IN MATRIX STIFF                        WIL00800
C IJMASS = ADDRESS OF ELEMENT I,J IN MATRIX MASS                         WIL00810
C IJKG   = ADDRESS OF ELEMENT I,J IN GEOMETRIC STIFFNESS MATRIX          WIL00820
C IJS    = ADDRESS OF ELEMENT I1,J1 IN MATRIX S                          WIL00830
C LSTIFF = ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF STIFF  WIL00840
C LMASS  = ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF MASS   WIL00850
C LKG    = ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF KG     WIL00860
C LS     = ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF S       WIL00870
C                                                                        WIL00880
C     DETERMINE THE FACTOR FOR KG                                        WIL00890
      IF (KGLOAD.EQ.1) THEN                                              WIL00900
            C10KG=C10*(1+ACC)                                            WIL00910
      ELSE IF (KGLOAD.EQ.2) THEN                                         WIL00920
            C10KG=C10                                                    WIL00930
      ENDIF                                                              WIL00940
C                                                                        WIL00950
      DO 225 J=L3,L4                                                     WIL00960
        J1=J-L3+1                                                        WIL00970
        LSTIFF=J-(MD(J+1)-MD(J)-1)                                       WIL00980
        LS    =J1-(MDS(J1+1)-MDS(J1)-1)+L2                               WIL00990
        L=MAX(LS,LSTIFF)                                                 WIL01000
        DO 210 I=J,L,-1                                                  WIL01010
           I1=I-L3+1                                                     WIL01020
           IJS=(MDS(J1)+J1)-I1                                           WIL01030
           IJSTIF=(MD(J)+J)-I                                            WIL01040
  210      S(IJS)=STIFF(IJSTIF)*C10                                      WIL01050
C                                                                        WIL01060
        IF (LS.LT.LSTIFF) THEN                                           WIL01070
           DO 215 I=L-1,LS,-1                                            WIL01080
              IJS=(MDS(J1)+J1)-I1                                        WIL01090
  215         S(IJS)=0                                                   WIL01100
        ENDIF                                                            WIL01110
C                                                                        WIL01120
        LMASS=J-(MDMASS(J+1)-MDMASS(J)-1)                                WIL01130
        L=MAX(LS,LMASS)                                                  WIL01140
        DO 220 I=J,L,-1                                                  WIL01150
           I1=I-L3+1                                                     WIL01160
           IJS=(MDS(J1)+J1)-I1                                           WIL01170
           IJMASS=(MDMASS(J)+J)-I                                        WIL01180
  220      S(IJS)=MASS(IJMASS)*C11 + S(IJS)                              WIL01190
C                                                                        WIL01200
        IF (KGFORM.EQ.2) THEN                                            WIL01210
C                                                                        WIL01220
C.        NOTE: THE GEOMETRIC STIFFNESS WAS DERIVED WITH COMPRESSION     WIL01230
C.        BEING POSITIVE, KG HAS POSITIVE TERMS ON THE MAIN DIAGIONAL    WIL01240
C.        REPRESENTING COMPRESSION, THUS KG IS SUBTRACTED FROM S         WIL01250
C.        CKG IS  AN INCREASE FOR VERT ACCEL                             WIL01260
      CKG=1.00                                                           WIL01270
C     ?????????                                                          WIL01280
C                                                                        WIL01290
C---- K(BAR) = K(BAR) - CKG*GEOMETRIC STIFFNESS                          WIL01300
           LKG=J-(MDKG(J+1)-MDKG(J)-1)                                   WIL01310
           L=MAX(LS,LKG)                                                 WIL01320
           DO 222 I=J,L,-1                                               WIL01330
             I1=I-L3+1                                                   WIL01340
             IJS=(MDS(J1)+J1)-I1                                         WIL01350
             IJKG=(MDKG(J)+J)-I                                          WIL01360
  222        S(IJS)=S(IJS) - KG(IJKG)*CKG                                WIL01370
        ENDIF                                                            WIL01380
C                                                                        WIL01390
  225   CONTINUE                                                         WIL01400
      ENDIF                                                              WIL01410
C                                                                        WIL01420
C------- CALCULATE Q AND R VECTORS                                       WIL01430
C....... QI IS TEMPROARLY STORED IN DA                                   WIL01440
C        Q(I) AND R(I) ARE MULTIPLIED BUT THE PROPER COIEFICIENTS FOR    WIL01450
C        PROPORTIONAL DAMPING                                            WIL01460
      DO 230 I=L3,L4                                                     WIL01470
        QI= C3*V(I) +  3*A(I)                                            WIL01480
        RI=  3*V(I) + C4*A(I)                                            WIL01490
        DA(I)= QI                                                        WIL01500
        Q(I)= QI + ALPHA*RI                                              WIL01510
  230   R(I)=      BETA *RI                                              WIL01520
C                                                                        WIL01530
C------- FORM DYNAMIC LOAD                                               WIL01540
C                                                                        WIL01550
C....... INITILIZE PBAR=DP+THETA                                         WIL01560
      DO 240 I=L3,L4                                                     WIL01570
  240   PBAR(I)=  DP(I)*THETA                                            WIL01580
C                                                                        WIL01590
      IF (FDAMP.EQ.1) THEN                                               WIL01600
C                                                                        WIL01610
C....... PBAR=PBAR +MASS*Q                                               WIL01620
        DO 250 J=L3,L4                                                   WIL01630
          MDJ=MDMASS(J)+J                                                WIL01640
          M=J-(MDMASS(J+1)-MDMASS(J))+1                                  WIL01650
          M=MAX(M,L3)                                                    WIL01660
          DO 250 I=M,J                                                   WIL01670
            PBAR(I)=PBAR(I)+ MASS(MDJ-I)*Q(J)                            WIL01680
  250     CONTINUE                                                       WIL01690
        DO 255 I=L3+1,L4                                                 WIL01700
          MDI=MDMASS(I)+I                                                WIL01710
          M=I-(MDMASS(I+1)-MDMASS(I))+1                                  WIL01720
          M=MAX(M,L3)                                                    WIL01730
          DO 255 J=M,I-1                                                 WIL01740
            PBAR(I)=PBAR(I)+ MASS(MDI-J)*Q(J)                            WIL01750
  255     CONTINUE                                                       WIL01760
C                                                                        WIL01770
C....... PBAR=BBAR +STIFF*R                                              WIL01780
        DO 260 J=L3,L4                                                   WIL01790
          MDJ=MD(J)+J                                                    WIL01800
          M=J-(MD(J+1)-MD(J))+1                                          WIL01810
          M=MAX(M,L3)                                                    WIL01820
          DO 260 I=M,J                                                   WIL01830
            PBAR(I)=PBAR(I)+ STIFF(MDJ-I)*R(J)                           WIL01840
  260     CONTINUE                                                       WIL01850
        DO 265 I=L3+1,L4                                                 WIL01860
          MDI=MD(I)+I                                                    WIL01870
          M=I-(MD(I+1)-MD(I))+1                                          WIL01880
          M=MAX(M,L3)                                                    WIL01890
          DO 265 J=M,I-1                                                 WIL01900
            PBAR(I)=PBAR(I)+ STIFF(MDI-J)*R(J)                           WIL01910
  265     CONTINUE                                                       WIL01920
      ENDIF                                                              WIL01930
C                                                                        WIL01940
C------- SOLVE FOR DISPLACEMENT AT TIME THETA*DT                         WIL01950
C....... NOTE: - WORK IS USED FOR TEMPORARY WORK SPACE                   WIL01960
C              - PBAR CONTAINS THE INCRIMENTAL DISPLACMENT BETWEEN       WIL01970
C                TIMES T AND T+THETA*DT AFTER THE SECOND CALL TO MSOLL   WIL01980
C                                                                        WIL01990
      CALL MSOLL(1,BUG,NFREE,IMDS,NFREE,1,1,NFREE,MDS,S,PBAR,WORK,       WIL02000
     &           .FALSE.,DUMMY,IDUMMY,1,                                 WIL02010
     &           .FALSE.,DUMMY,IDUMMY,1,ABORT)                           WIL02020
      IF (ABORT) RETURN                                                  WIL02030
      CALL MSOLL(3,BUG,NFREE,IMDS,NFREE,1,1,NFREE,MDS,S,PBAR,WORK,       WIL02040
     &           .FALSE.,DUMMY,IDUMMY,1,                                 WIL02050
     &           .FALSE.,DUMMY,IDUMMY,1,ABORT)                           WIL02060
      IF (ABORT) RETURN                                                  WIL02070
C                                                                        WIL02080
C                                                                        WIL02090
C------- DETERMINE ACCELERATION, VELOCITY AND DISPLACEMENT AT TIME DT    WIL02100
C....... RECALL THAT QI WAS TEMPORARLY STORED IN DA(I)                   WIL02110
C        DA = INCRIMENTAL ACCELERATION BETWEEN TIMES T AND T+DT          WIL02120
C        DV = INCRIMENTAL VELOCTIY BETWEEN TIMES T AND T+DT              WIL02130
C        DD = INCRIMENTAL DISPLACMENT BETWEEN TIMES T AND T+DT           WIL02140
C                                                                        WIL02150
      DO 270 I=L3,L4                                                     WIL02160
        DA(I)=( C1*PBAR(I) -  DA(I) )/THETA                              WIL02170
        DV(I)=A(I)*DT + C5*DA(I)                                         WIL02180
  270   DD(I)=V(I)*DT + C6* A(I) + C7*DA(I)                              WIL02190
C                                                                        WIL02200
      RETURN                                                             WIL02210
C                                                                        WIL02220
C------------------------------------------------------                  WIL02230
C- CALC TOTAL DISPLACEMENTS, VELOCITIES AND ACCELERATIONS --             WIL02240
C------------------------------------------------------                  WIL02250
  300 CONTINUE                                                           WIL02260
      DO 310 I=1,NDOF                                                    WIL02270
        A(I)=DA(I) + A(I)                                                WIL02280
        V(I)=DV(I) + V(I)                                                WIL02290
        D(I)=DD(I) + D(I)                                                WIL02300
        P(I)=DP(I) + P(I)                                                WIL02310
  310   DP(I)=0.00                                                       WIL02320
C                                                                        WIL02330
      RETURN                                                             WIL02340
      END                                                                WIL02350
C=======================================================================WMA00010
```

```
C          DEBUG UNIT(6),SUBCHK,SUBTRACE                                WMA00020
C          END DEBUG                                                    WMA00030
      SUBROUTINE WMATRX(A,IR,IC,NAME)                                   WMA00040
      DIMENSION A(IR,IC)                                                WMA00050
      CHARACTER*(*) NAME                                                WMA00060
      LOGICAL DMP,BTEST                                                 WMA00070
C                                                                       WMA00080
      CALL GETINT(NAME,'DUMP=',IDUMP,0,.TRUE.,.FALSE.)                  WMA00090
      IF (IDUMP.NE.0)  THEN                                             WMA00100
           DMP=.TRUE.                                                   WMA00110
      ELSE                                                              WMA00120
           DMP=.FALSE.                                                  WMA00130
      ENDIF                                                             WMA00140
C                                                                       WMA00150
      WRITE (6,*) ' '                                                   WMA00160
      WRITE (6,*) 'PRINTING MATRIX: ',NAME                              WMA00170
      DO 50 ICG=1,IC,6                                                  WMA00180
         IF (ICG.GT.1) WRITE (6,*) ' '                                  WMA00190
         K=MIN((ICG+5),IC)                                              WMA00200
         WRITE (6,10) (J,J=ICG,K)                                       WMA00210
         DO 20 I=1,IR                                                   WMA00220
           IF (DMP) THEN                                                WMA00230
             DO 15 J=ICG,K                                              WMA00240
  15         IF (A(I,J).NE.0) WRITE (IDUMP,*) I,J,A(I,J),NAME           WMA00250
             ENDIF                                                      WMA00260
  20         WRITE (6,30)  I,(A(I,J),J=ICG,K)                           WMA00270
  50 CONTINUE                                                           WMA00280
      RETURN                                                            WMA00290
  10 FORMAT(' COLUMN',T17,I5,5(15X,I5))                                 WMA00300
  30 FORMAT(' ROW',I4,6G20.6)                                           WMA00310
      END                                                               WMA00320
C=======================================================================WMT00010
C          DEBUG UNIT(6),SUBCHK,SUBTRACE                                WMT00020
C          END DEBUG                                                    WMT00030
      SUBROUTINE WMTRX2(A,IR,IC,NAME,IROW)                              WMT00040
      DIMENSION A(IROW,IC)                                              WMT00050
      CHARACTER*(*) NAME                                                WMT00060
      LOGICAL DMP,BTEST                                                 WMT00070
C                                                                       WMT00080
      CALL GETINT(NAME,'DUMP=',IDUMP,0,.TRUE.,.FALSE.)                  WMT00090
      IF (IDUMP.NE.0)  THEN                                             WMT00100
           DMP=.TRUE.                                                   WMT00110
      ELSE                                                              WMT00120
           DMP=.FALSE.                                                  WMT00130
      ENDIF                                                             WMT00140
C                                                                       WMT00150
      WRITE (6,*) ' '                                                   WMT00160
      WRITE (6,*) 'PRINTING MATRIX: ',NAME                              WMT00170
      DO 50 ICG=1,IC,6                                                  WMT00180
         IF (ICG.GT.1) WRITE (6,*) ' '                                  WMT00190
         K=MIN((ICG+5),IC)                                              WMT00200
         WRITE (6,10) (J,J=ICG,K)                                       WMT00210
         DO 20 I=1,IR                                                   WMT00220
           IF (DMP) THEN                                                WMT00230
             DO 15 J=ICG,K                                              WMT00240
  15         IF (A(I,J).NE.0) WRITE (IDUMP,*) I,J,A(I,J),NAME           WMT00250
             ENDIF                                                      WMT00260
  20         WRITE (6,30)  I,(A(I,J),J=ICG,K)                           WMT00270
  50 CONTINUE                                                           WMT00280
      RETURN                                                            WMT00290
  10 FORMAT(' COLUMN',T17,I5,5(15X,I5))                                 WMT00300
  30 FORMAT(' ROW',I4,6F20.6)                                           WMT00310
      END                                                               WMT00320
.*
.pf  ;.ls normal
.fo on
.sp 1 A
.us C, PROGRAM SEE
.fo off trunc
.sp 1 A
.us FILE: SEE FORTRAN
.sp 1 A
.bf p4 ;.ls normal p5
C=======================================================================SEE00010
C          DEBUG UNIT(6),TRACE,SUBCHK,INIT,SUBTRACE                     SEE00020
C          END DEBUG                                                    SEE00030
      PARAMETER (MMAX=1500)                                             SEE00040
      LOGICAL PAXIS                                                     SEE00050
      CHARACTER*20 LABEL(3)                                             SEE00060
      CHARACTER*80 TITLE                                                SEE00070
      CHARACTER*42 TEMP                                                 SEE00080
      DIMENSION VX(3),VY(3),VZ(3),X(4),Y(4)                             SEE00090
      DIMENSION S(3,MMAX),SR(3,MMAX),KEY(MMAX),RMAX(3),RMIN(3)          SEE00100
      DIMENSION E(3,MMAX),ER(3,MMAX)                                    SEE00110
      CROSSI(A1,B1,C1,A2,B2,C2)=B1*C2-B2*C1                             SEE00120
      CROSSJ(A1,B1,C1,A2,B2,C2)=C1*A2-C2*A1                             SEE00130
      CROSSK(A1,B1,C1,A2,B2,C2)=A1*B2-A2*B1                             SEE00140
      VALUE (A1,A2,A3)=SQRT(A1**2+A2**2+A3**2)                          SEE00150
      DOT   (A1,B1,C1,A2,B2,C2)=A1*A2+B1*B2+C1*C2                       SEE00160
      LABEL(1)='ROTATED X AXIS'                                         SEE00170
      LABEL(2)='ROTATED Y AXIS'                                         SEE00180
      LABEL(3)='ROTATED Z AXIS'                                         SEE00190
C                                                                       SEE00200
C------ INITIALISE THE PLOTTING DEVICE ----                            SEE00210
      CALL PLOTS(0,0,7)                                                 SEE00220
C                                                                       SEE00230
C------ INPUT & PRINT POINTS --------                                   SEE00240
      DO 10 I=1,MMAX                                                    SEE00250
      READ  (10,*,END=15) KEY(I),(S(J,I),J=1,3),(E(J,I),J=1,3)          SEE00260
   10 WRITE (6,20) KEY(I),(S(J,I),J=1,3),(E(J,I),J=1,3),'INPUT POINT'   SEE00270
   15 NUMP=I                                                            SEE00280
      IF (I.LE.MMAX) NUMP=I-1                                           SEE00290
   20 FORMAT (1X,I5,1P,6G12.4,1X,A)                                     SEE00300
C                                                                       SEE00310
C------ LOOP FOR EACH VIEW ----------                                   SEE00320
      READ(5,*) NVIEW                                                   SEE00330
      DO 2000 IVIEW=1,NVIEW                                             SEE00340
C                                                                       SEE00350
C------ INPUT VIEWING DATA ----------                                   SEE00360
      READ (5,*)  TITLE,XLEN,YLEN,PAXIS,VX,VY                          SEE00370
C                                                                       SEE00380
C------ CALCULATE DIRECTION COSINES -                                   SEE00390
      VZ(1)= VX(2)*VY(3)-VX(3)*VY(2)                                    SEE00400
      VZ(2)=-VX(1)*VY(3)+VX(3)*VY(1)                                    SEE00410
      VZ(3)= VX(1)*VY(2)-VX(2)*VY(1)                                    SEE00420
      VY(1)= VZ(2)*VX(3)-VZ(3)*VX(2)                                    SEE00430
      VY(2)=-VZ(1)*VX(3)+VZ(3)*VX(1)                                    SEE00440
      VY(3)= VZ(1)*VX(2)-VZ(2)*VX(1)                                    SEE00450
      DX=VALUE(VX(1),VX(2),VX(3))                                       SEE00460
      DY=VALUE(VY(1),VY(2),VY(3))                                       SEE00470
      DZ=VALUE(VZ(1),VZ(2),VZ(3))                                       SEE00480
      DO 30 I=1,3                                                       SEE00490
        VX(I)=VX(I)/DX                                                  SEE00500
        VY(I)=VY(I)/DY                                                  SEE00510
   30   VZ(I)=VZ(I)/DZ                                                  SEE00520
C                                                                       SEE00530
C------ PRINT DIRECTION COSINES -----                                   SEE00540
      WRITE (6,*) ' VIEW NUMBER.....:',IVIEW                            SEE00550
      WRITE (6,*) ' TITLE.....:',TITLE                                  SEE00560
      WRITE (6,*) ' X-LENGTH.....:',XLEN                                SEE00570
      WRITE (6,*) ' Y-LENGTH.....:',YLEN                                SEE00580
      WRITE (6,*) ' VX.....:',VX                                        SEE00590
      WRITE (6,*) ' VY.....:',VY                                        SEE00600
      WRITE (6,*) ' VZ.....:',VZ                                        SEE00610
C                                                                       SEE00620
C------ CALCULATE & PRINT ROTATED COORDINATES ----                      SEE00630
      DO 100 J=1,NUMB                                                   SEE00640
        DO 40 I=1,3                                                     SEE00650
          SR(I,J)=0                                                     SEE00660
   40     SR(I,J)=0                                                     SEE00670
        DO 50 I=1,3                                                     SEE00680
          SR(1,J)= SR(1,J) + VX(I)*S(I,J)                               SEE00690
          SR(2,J)= SR(2,J) + VY(I)*S(I,J)                               SEE00700
          SR(3,J)= SR(3,J) + VZ(I)*S(I,J)                               SEE00710
          ER(1,J)= ER(1,J) + VX(I)*E(I,J)                               SEE00720
          ER(2,J)= ER(2,J) + VY(I)*E(I,J)                               SEE00730
   50     ER(3,J)= ER(3,J) + VZ(I)*E(I,J)                               SEE00740
        IF (J.EQ.1) THEN                                                SEE00750
          DO 60 I=1,3                                                   SEE00760
            RMAX(I)=MAX(SR(I,J),ER(I,J))                                SEE00770
   60       RMIN(I)=MIN(SR(I,J),ER(I,J))                                SEE00780
        ELSE                                                            SEE00790
          DO 70 I=1,3                                                   SEE00800
            RMAX(I)=MAX(SR(I,J),ER(I,J),RMAX(I))                        SEE00810
   70       RMIN(I)=MIN(SR(I,J),ER(I,J),RMIN(I))                        SEE00820
        ENDIF                                                           SEE00830
  100   WRITE (6,20) KEY(J),(SR(I,J),I=1,3),(ER(I,J),I=1,3),           SEE00840
     &          'ROTATED POINT'                                         SEE00850
C                                                                       SEE00860
C=======================================================================SEE00870
C------ PLOT POINTS, USING CALCOMP 1051 PLOTTER COMMANDS...             SEE00880
C=======================================================================SEE00890
C                                                                       SEE00900
C                                                                       SEE00910
C------ LOOP FOR DIFFENINET VIEWS ---                                   SEE00920
      DO 500 IJ=1,3                                                     SEE00930
        IF (IJ.EQ.1) THEN                                               SEE00940
            IX=1                                                        SEE00950
            IY=2                                                        SEE00960
        ELSE IF (IJ.EQ.2) THEN                                          SEE00970
            IX=2                                                        SEE00980
            IY=3                                                        SEE00990
        ELSE                                                            SEE01000
            IX=3                                                        SEE01010
            IY=1                                                        SEE01020
        ENDIF                                                           SEE01030
C                                                                       SEE01040
C------ DRAW TRIM LINE --------------                                   SEE01050
      CALL PLOT(0.,11.,2)                                               SEE01060
      CALL PLOT(0., 0.,2)                                               SEE01070
C                                                                       SEE01080
C------ MOVE ORIGIN TO 0.5,0.5 ------                                   SEE01090
      CALL PLOT(.75,.5,-3)                                              SEE01100
C                                                                       SEE01110
C------ GET SCALE FACTORS FOR X-Y PLOT ---                              SEE01120
      X(1)=RMIN(IX)                                                     SEE01130
      X(2)=RMAX(IX)                                                     SEE01140
      Y(1)=RMIN(IY)                                                     SEE01150
      Y(2)=RMAX(IY)                                                     SEE01160
      CALL SCALE(X,XLEN,2,1)                                            SEE01170
      CALL SCALE(Y,YLEN,2,1)                                            SEE01180
      X3=X(3)                                                           SEE01190
      X4=X(4)                                                           SEE01200
      Y3=Y(3)                                                           SEE01210
      Y4=Y(4)                                                           SEE01220
C                                                                       SEE01230
C------ PLOT AXIS -------------------------                             SEE01240
      IF (PAXIS) THEN                                                   SEE01250
           CALL AXIS(0.00,-.05,LABEL(IX),-20,XLEN, 0.,X(3),X(4))        SEE01260
           CALL AXIS(-.05,0.00,LABEL(IY), 20,YLEN,90.,Y(3),Y(4))        SEE01270
      ENDIF                                                             SEE01280
C                                                                       SEE01290
C------ PLOT MEMBERS ----------------------                             SEE01300
      DO 200 J=1,NUMB                                                   SEE01310
        X(1)=SR(IX,J)                                                   SEE01320
        X(2)=ER(IX,J)                                                   SEE01330
        Y(1)=SR(IY,J)                                                   SEE01340
        Y(2)=ER(IY,J)                                                   SEE01350
  200   CALL LINE(X,Y,2,1,0,0)                                          SEE01360
C                                                                       SEE01370
C------ PLOT TITLE ----------------------                               SEE01380
      CALL SYMBOL(0.1,(YLEN+.15),.14,TITLE ,0.,80)                      SEE01390
      TEMP=LABEL(IX)//'--'//LABEL(IY)                                   SEE01400
      CALL SYMBOL(0.1,(YLEN+.01),.05,TEMP  ,0.,42)                      SEE01410
C                                                                       SEE01420
C------ MOVE ORIGIN BACK TO 0,0 AND ADVANCE PAGE -----                  SEE01430
      CALL PLOT((XLEN+.25),-.5,-3)                                      SEE01440
C                                                                       SEE01450
  500 CONTINUE                                                          SEE01460
C                                                                       SEE01470
 1000 CONTINUE                                                          SEE01480
 2000 CONTINUE                                                          SEE01490
C                                                                       SEE01500
C------ DRAW TRIM LINE --------------                                   SEE01510
      CALL PLOT(0.,11.,2)                                               SEE01520
      CALL PLOT(0., 0.,2)                                               SEE01530
C                                                                       SEE01540
C------ CLOSE THE PLOTTING DEVICE ----                                  SEE01550
      CALL PLOT(0.,0.,999)                                              SEE01560
C                                                                       SEE01570
      STOP                                                              SEE01580
      END                                                               SEE01590
```