

Table A14 List of the file, **skeletal struct.new**.

This is the completed file after the GENOPT user's completion of the "GENTEXT" interactive session. The FORTRAN statements in this file become the **skeletal struct.new** library, in particular, part of skeletal SUBROUTINE STRUCT. GENOPT does this automatically. See Table a30 for a list of the "howto.struct" file.

```
=====
C=DECK      STRUCT
      SUBROUTINE STRUCT(IMODX,CONSTX,OBJGEN,CONMAX,NCONSX,IPOINC,
1 PCWORD,CPLTX,ILOADX,ISTARX,NUSERC,IBEHV,IDV,IFAST,JJJ1)
C
C PURPOSE IS TO PERFORM THE ANALYSIS FOR A GIVEN DESIGN AND LOADING.
C CONSTRAINT CONDITIONS ARE ALSO GENERATED.
C
C Common blocks already present in the struct.tmpl file, that is,
C in the "skeletal" file possibly to be augmented by the user:
      COMMON/PRMFIL/IFILEX,IFILE2,IOUT,IPRM(5)
      COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
      COMMON/INDAT/INFILE
      COMMON/LWRUPR/VLBX(50),VUBX(50),CLINKX(50,5),VLINKX(50),VBVX(99)
      COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
      COMMON/PARAMS/PARX(99),VARX(50),ALLOWX(99),CONSXX(99),DECX(50),
1          ESCX(50)
      COMMON/WORDS1/WORDPX(99),WORDVX(50),WORDAX(99),WORDCC(99),
1          WORDDX(50)
      COMMON/WORDS2/WORDLX(50),WORDEX(50),WORDIQ(20)
      COMMON/OPTVAR/IDVX(50),ILVX(50),IDLINK(50,5),IEVX(50),JTERMS(20)
      COMMON/NUMPR2/ILARX,ICARX,IOARX,IFLATX,NCASES,NPRINX
      COMMON/PARAM2/FLARX(50),CARX(99),OARX(50),FSAFEX(99),CPWRX(50,5)
      COMMON/PARAM3/CINEQX(15,20),DPWREQ(15,20)
      COMMON/PARAM4/IDINEQ(15,20),NINEQX,JINEQX(20),IEQTYP(20)
      COMMON/WORDS3/WORDFX(50),WORDBX(99),WORDOB(50),WORDSX(99)
      COMMON/WORDS4/WORDMX(99)
      COMMON/PWORD/PHRASE
      COMMON/PWORD2/IBLANK
      COMMON/ISKIPX/ISKIP(30)
      DIMENSION IBEHV(99)
C
C=====
=
C Start of first part of STRUCT written by "GENTEXT"
C INSERT ADDITIONAL COMMON BLOCKS HERE: (THESE ARE "GENTEXT" VARIABLES)
      COMMON/FV01/xinput(21),Ixinput
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
```

```

COMMON/FV16/PRESS(20)
REAL PRESS
COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
REAL CLAPS1,CLAPS1A,CLAPS1F
COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
REAL GENBK1,GENBK1A,GENBK1F
COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
REAL SKNBK1,SKNBK1A,SKNBK1F
COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
REAL STFBK1,STFBK1A,STFBK1F
COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
REAL SKNST1,SKNST1A,SKNST1F
COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
REAL STFST1,STFST1A,STFST1F
COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
REAL WAPEX1,WAPEX1A,WAPEX1F
COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
REAL CLAPS2,CLAPS2A,CLAPS2F
COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
REAL GENBK2,GENBK2A,GENBK2F
COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
REAL SKNBK2,SKNBK2A,SKNBK2F
COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
REAL STFBK2,STFBK2A,STFBK2F
COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
REAL SKNST2,SKNST2A,SKNST2F
COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
REAL STFST2,STFST2A,STFST2F
COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
REAL WAPEX2,WAPEX2A,WAPEX2F
COMMON/IV01/npoint,nodes,IMODE
INTEGER npoint,nodes,IMODE
COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT

```

C  
C

```

CHARACTER*80 PHRASE,CODPHR,PCWORD
CHARACTER*80 WORDPX,WORDVX,WORDAX,WORDCX,WORDDX,WORDLX,WORDEX
CHARACTER*80 WORDFX,WORDBX,WORDOB,WORDSX,WORDMX,WORDCC,WORDIQ
CHARACTER*4 ANSOUT,CHARAC,ANSWER
CHARACTER*2 CIX
character*2 CJX
CHARACTER*13 CODNAM
DIMENSION ISUBX(100)
LOGICAL ANSL1

```

C

```

DIMENSION CONSTX(*),IPOINC(*),PCWORD(*),CPLOTX(*)

```

C End of first part of STRUCT written by "GENTEXT"

```

C=====
C
C  INSERT ADDITIONAL DIMENSION AND/OR LABELLED COMMON BLOCKS HERE,
C  IF NECESSARY. THESE WOULD BE STATEMENTS THAT ARE CONSISTENT WITH
C  SUBROUTINES THAT YOU OR OTHERS MAY HAVE WRITTEN THAT ARE REQUIRED
C  FOR WHATEVER ANALYSIS YOU ARE PERSUING.  MAKE SURE THAT YOU DO NOT
C  INTRODUCE NAME CONFLICTS WITH THE "GENTEXT" LABELLED COMMON BLOCKS
C  LISTED ABOVE.
C
C  Please note that you do not have to modify STRUCT.NEW if you would
C  rather provide all of your algorithms via the BEHAVIOR.NEW library.
C  (See instructions in BEHAVIOR.NEW).
C
C  If you are using a lot of software previously written either by
C  yourself or others, or if there are a lot of behavioral constraints
C  that are best generated by looping over array indices (such as
C  occurs, for example, with stress constraints in laminates of
C  composite materials), then it may be best to insert your common
C  blocks and dimension statements here, your subroutine calls
C  below (where indicated), and your subroutines in any of the libraries
C  called ADDCODEn.NEW, n = 1,2,...,5.  Please note that you
C  may also have to add statements to SUBROUTINE TRANFR, the
C  purpose of which is described below (in TRANFR).
C
C  The several test cases provided with GENOPT demonstrate different
C  methods:
C
C  PLATE   : leave STRUCT.NEW unchanged and modify BEHAVIOR.NEW
C  SPHERE  : leave STRUCT.NEW unchanged and modify BEHAVIOR.NEW
C  TORISPH: leave BEHAVIOR.NEW unchanged except possibly for the objective
C            function (SUBROUTINE OBJECT), modify STRUCT.NEW,
C            possibly add a subroutine library called ADDCODE1.NEW, and
C            possibly augment the usermake.linux file to collect object
C            libraries from other directories.  In the "TORISPH" case
C            BEHAVIOR.NEW remains unchanged, no ADDCODE1.NEW library is
C            added, and usermake.linux is not changed.  Instead, the
C            BIGBOSOR4 code is added and SUBROUTINE BOSDEC is written
C            by the genopt user.  The BIGBOSOR4 code and SUBROUTINE
C            BOSDEC must be stored in /home/progs/bosdec/sources, as
C            follows:
C  BIGBOSOR4 code:
C      -rw-r--r--  1 bush bush 579671 Feb 29 07:19 addbosor4.src
C      -rw-r--r--  1 bush bush 83175 Feb 22 09:13 b4plot.src
C      -rw-r--r--  1 bush bush 89671 Feb 28 16:20 b4util.src
C      -rw-r--r--  1 bush bush 22723 Feb 10 14:27 bio.c
C      -rw-r--r--  1 bush bush 31175 Feb 10 14:27 bio_linux.c
C      -rw-r--r--  1 bush bush 37152 Feb 10 14:27 bio_linux.o

```

```

C      -rw-r--r--  1 bush bush  15650 Feb 10 14:26 gasp.F
C      -rw-r--r--  1 bush bush  18364 Feb 10 14:26 gasp_linux.o
C      -rw-r--r--  1 bush bush   6310 Feb 13 10:12 opngen.src
C      -rw-r--r--  1 bush bush  22440 Feb 10 14:25 prompter.src
C      -rw-r--r--  1 bush bush  13426 Feb 22 09:14 resetup.src
C      BOSDEC.src code:
C      -rw-r--r--  1 bush bush  33851 Mar  1 08:34 bosdec.src
C
C      WAVYCYL: both BEHAVIOR.NEW and STRUCT.NEW are both changed. Otherwise
C                the activity is the same as that described for TORISPH,
C                except, of course, that struct.new is different from
C                that used in connection with TORISPH.
C
C      CYLINDER:same as the description for WAVYCYL.
C
C
C      INSERT YOUR ADDITIONAL COMMON BLOCKS FOR THIS GENERIC CASE HERE:
C
C
C      THE FOLLOWING CODE WAS WRITTEN BY "GENTEXT":
C
C=====
C      Start the second portion of STRUCT written by "GENTEXT":
C
C      ICARX    = ISTARX
C      INUMTT = 0
C      ICONSX  = 0
C      KCONX   = 0
C      IF (IMODX.EQ.0) THEN
C          CALL MOVERX(0.,0,CONSTX,1,99)
C          CALL MOVERX(0, 0,IPOINC,1,1500)
C      ENDIF
C
C      IF (ILOADX.EQ.1) THEN
C
C      ESTABLISH FIRST ANY CONSTRAINTS THAT ARE INEQUALITY RELATIONSHIPS
C      AMONG THE VARIABLES IN THE ARRAY VARX(*) (THAT IS, VARIABLES THAT
C      ARE EITHER DECISION VARIABLES, LINKED VARIABLES, ESCAPE VARIABLES,
C      OR CANDIDATES FOR ANY OF THESE TYPES OF VARIABLES.
C
C          IF (NINEQX.GT.0)
C      1          CALL VARCON(WORDIQ,WORDMX,CINEQX,DPWREQ,IDINEQ,
C      1          NINEQX,JINEQX,IEQTYP,INUMTT,IMODX,CONMAX,IPOINC,
C      1          ICONSX,CONSTX,VARX,PCWORD,CPLOTX,ICARX)
C
C      NEXT, ESTABLISH USER-WRITTEN CONSTRAINTS. AT PRESENT, THE PROGRAM
C      ALLOWS ONLY ONE USER-WRITTEN CONSTRAINT. HOWEVER, THE USER CAN
C      EASILY EXPAND THIS CAPABILITY SIMPLY BY ADDING SUBROUTINES THAT

```

C ARE ANALOGOUS TO USRCN (WITH NAMES SUCH AS USRCN2, USRCN3, ETC.  
 C TO THE BEHAVIOR.NEW LIBRARY, AND ADD CALLS TO THESE ADDITIONAL  
 C SUBROUTINES FOLLOWING THE CALL TO USRCN IMMEDIATELY BELOW.

```
C
      CALL USRCN(INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1          WORDMX,PCWORD,CPLTX,ICARX,IFILE8)
```

```
C
      NUSERC = ICARX - NINEQX
      ENDIF
```

```
C
      IF (NPRINX.GT.0) THEN
        WRITE(IFILE8,'(1X,A,I2,A)')
1      ' BEHAVIOR FOR ',ILOADX,' ENVIRONMENT (LOAD SET)'
        WRITE(IFILE8,'(A)')' '
        WRITE(IFILE8,'(A)')
1      ' CONSTRAINT BEHAVIOR          DEFINITION'
        WRITE(IFILE8,'(A)')
1      '   NUMBER      VALUE'
      ENDIF
```

```
C
      CALL CONVR2(ILOADX,CIX)
      IF (NPRINX.GT.0) THEN
        WRITE(IFILE8,'(1X,A)')' '
        WRITE(IFILE8,'(1X,A,I2)')
1      ' BEHAVIOR FOR LOAD SET NUMBER, ILOADX=',ILOADX
      ENDIF
```

C  
 C End of the second portion of STRUCT written by "GENTEXT"

C=====

C  
 C USER: YOU MAY WANT TO INSERT SUBROUTINE CALLS FROM SOFTWARE DEVELOPED  
 C ELSEWHERE FOR ANY CALCULATIONS PERTAINING TO THIS LOAD SET.

C  
 C=====

C Start of the final portion of STRUCT written by "GENTEXT"

C  
 C INSERT THE PROGRAM FILE HERE:

C  
 C Behavior and constraints generated next for CLAPS1:  
 C CLAPS1 = collapse pressure with imperfection mode 1

```
C
      PHRASE =
1      'collapse pressure with imperfection mode 1'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(1 ).EQ.0) CALL BEHX1
1      (IFILE8,NPRINX,IMODX,IFAST,ILOADX ,
1      'collapse pressure with imperfection mode 1')
      IF (CLAPS1(ILOADX ).EQ.0.) CLAPS1(ILOADX ) = 1.E+10
```

```

IF (CLAPS1A(ILOADX ).EQ.0.) CLAPS1A(ILOADX ) = 1.0
IF (CLAPS1F(ILOADX ).EQ.0.) CLAPS1F(ILOADX ) = 1.0
KCONX = KCONX + 1
CARX(KCONX) =CLAPS1(ILOADX )
WORDCX= '(CLAPS1('//CIX//')/CLAPS1A('//CIX//
1 ')) / CLAPS1F('//CIX//')'
CALL CONX(CLAPS1(ILOADX ),CLAPS1A(ILOADX ),CLAPS1F(ILOADX )
1,'collapse pressure with imperfection mode 1',
1 'allowable pressure for axisymmetric collapse',
1 'factor of safety for axisymmetric collapse',
1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLOTX,ICARX)
IF (IMODX.EQ.0) THEN
CODPHR =
1 ' collapse pressure with imperfection mode 1: '
IENDP4 =46
CODNAM ='CLAPS1('//CIX//')'
MLET4 =6 + 4
WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1 KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
ENDIF
125 CONTINUE
126 CONTINUE

```

```

C
C Behavior and constraints generated next for GENBK1:
C GENBK1 = general buckling load factor, mode 1
C

```

```

PHRASE =
1 'general buckling load factor, mode 1'
CALL BLANKX(PHRASE,IENDP4)
IF (IBEHV(2 ).EQ.0) CALL BEHX2
1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX ,
1 'general buckling load factor, mode 1')
IF (GENBK1(ILOADX ).EQ.0.) GENBK1(ILOADX ) = 1.E+10
IF (GENBK1A(ILOADX ).EQ.0.) GENBK1A(ILOADX ) = 1.0
IF (GENBK1F(ILOADX ).EQ.0.) GENBK1F(ILOADX ) = 1.0
KCONX = KCONX + 1
CARX(KCONX) =GENBK1(ILOADX )
WORDCX= '(GENBK1('//CIX//')/GENBK1A('//CIX//
1 ')) / GENBK1F('//CIX//')'
CALL CONX(GENBK1(ILOADX ),GENBK1A(ILOADX ),GENBK1F(ILOADX )
1,'general buckling load factor, mode 1',
1 'allowable general buckling load factor (use 1.0)',
1 'factor of safety for general buckling',
1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLOTX,ICARX)
IF (IMODX.EQ.0) THEN

```

```

        CODPHR =
1  '  general buckling load factor, mode 1: '
        IENDP4 =40
        CODNAM ='GENBK1('//CIX//')'
        MLET4 =6 + 4
        WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
        IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1      KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
        ENDIF
140 CONTINUE
141 CONTINUE
C
C  Behavior and constraints generated next for SKNBK1:
C  SKNBK1 = local skin buckling load factor, mode 1
C
        IF (JSKNBK1.EQ.0) GO TO 161
        IF (NPRINX.GT.0) THEN
            IF (JSKNBK1.GT.1) THEN
                WRITE(IFILE8,'(1X,A)') ' '
                WRITE(IFILE8,'(1X,A,$)') ' BEHAVIOR OVER J = '
                WRITE(IFILE8,'(1X,A)')
1          'number of regions for computing behavior'
            ENDIF
        ENDIF
        DO 160 J=1,JSKNBK1
            CALL CONVR2(J,CJX)
            PHRASE =
1  'local skin buckling load factor, mode 1'
            CALL BLANKX(PHRASE,IENDP4)
            IF (IBEHV(3 ).EQ.0) CALL BEHX3
1  (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
1  'local skin buckling load factor, mode 1')
            IF (SKNBK1(ILOADX,J).EQ.0.) SKNBK1(ILOADX,J) = 1.E+10
            IF (SKNBK1A(ILOADX,J).EQ.0.) SKNBK1A(ILOADX,J) = 1.0
            IF (SKNBK1F(ILOADX,J).EQ.0.) SKNBK1F(ILOADX,J) = 1.0
            KCONX = KCONX + 1
            CARX(KCONX) =SKNBK1(ILOADX,J)
            WORDCX= '(SKNBK1('//CIX//','//CJX//')/SKNBK1A('//CIX//','//CJX//
1  ')) / SKNBK1F('//CIX//','//CJX//')'
            CALL CONX(SKNBK1(ILOADX,J),SKNBK1A(ILOADX,J),SKNBK1F(ILOADX,J)
1,'local skin buckling load factor, mode 1',
1 'allowable buckling load factor',
1 'factor of safety for skin buckling',
1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLOTX,ICARX)
            IF (IMODX.EQ.0) THEN
                CODPHR =
1  '  local skin buckling load factor, mode 1: '

```

```

        IENDP4 =43
        CODNAM ='SKNBK1('//CIX//','//CJX//')'
        MLET4 =6 + 7
        WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
        IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1      KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
        ENDIF
160 CONTINUE
161 CONTINUE
C
C Behavior and constraints generated next for STFBK1:
C STFBK1 = buckling load factor, isogrid member, mode 1
C
        IF (JSKNBK1.EQ.0) GO TO 176
        IF (NPRINX.GT.0) THEN
            IF (JSKNBK1.GT.1) THEN
                WRITE(IFILE8,'(1X,A)') ' '
                WRITE(IFILE8,'(1X,A,$)') ' BEHAVIOR OVER J = '
                WRITE(IFILE8,'(1X,A)')
1            'number of regions for computing behavior'
            ENDIF
        ENDIF
        DO 175 J=1,JSKNBK1
            CALL CONVR2(J,CJX)
            PHRASE =
1 'buckling load factor, isogrid member, mode 1'
            CALL BLANKX(PHRASE,IENDP4)
            IF (IBEHV(4 ).EQ.0) CALL BEHX4
1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
1 'buckling load factor, isogrid member, mode 1')
            IF (STFBK1(ILOADX,J).EQ.0.) STFBK1(ILOADX,J) = 1.E+10
            IF (STFBK1A(ILOADX,J).EQ.0.) STFBK1A(ILOADX,J) = 1.0
            IF (STFBK1F(ILOADX,J).EQ.0.) STFBK1F(ILOADX,J) = 1.0
            KCONX = KCONX + 1
            CARX(KCONX) =STFBK1(ILOADX,J)
            WORDCX= '(STFBK1('//CIX//','//CJX//')/STFBK1A('//CIX//','//CJX//
1 ' ')) / STFBK1F('//CIX//','//CJX//')'
            CALL CONX(STFBK1(ILOADX,J),STFBK1A(ILOADX,J),STFBK1F(ILOADX,J)
1,'buckling load factor, isogrid member, mode 1',
1 'allowable for isogrid stiffener buckling (Use 1.)',
1 'factor of safety for isogrid stiffener buckling',
1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLTX,ICARX)
            IF (IMODX.EQ.0) THEN
                CODPHR =
1 ' buckling load factor, isogrid member, mode 1: '
                IENDP4 =48
                CODNAM ='STFBK1('//CIX//','//CJX//')'

```



```

        MLET4 =6 + 7
        WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
        IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1       KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
        ENDIF
175 CONTINUE
176 CONTINUE
C
C Behavior and constraints generated next for SKNST1:
C SKNST1 = maximum stress in the shell skin, mode 1
C
        IF (JSKNBK1.EQ.0) GO TO 191
        IF (NPRINX.GT.0) THEN
            IF (JSKNBK1.GT.1) THEN
                WRITE(IFILE8,'(1X,A)') ' '
                WRITE(IFILE8,'(1X,A,$)') ' BEHAVIOR OVER J = '
                WRITE(IFILE8,'(1X,A)')
1            'number of regions for computing behavior'
            ENDIF
        ENDIF
        DO 190 J=1,JSKNBK1
            CALL CONVR2(J,CJX)
            PHRASE =
1 'maximum stress in the shell skin, mode 1'
            CALL BLANKX(PHRASE,IENDP4)
            IF (IBEHV(5 ).EQ.0) CALL BEHX5
1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
1 'maximum stress in the shell skin, mode 1')
            IF (SKNST1(ILOADX,J).EQ.0.) SKNST1(ILOADX,J) = 1.E-10
            IF (SKNST1A(ILOADX,J).EQ.0.) SKNST1A(ILOADX,J) = 1.0
            IF (SKNST1F(ILOADX,J).EQ.0.) SKNST1F(ILOADX,J) = 1.0
            KCONX = KCONX + 1
            CARX(KCONX) =SKNST1(ILOADX,J)
            WORDCX= '(SKNST1A('//CIX//','//CJX//')/SKNST1('//CIX//','//CJX//
1 ' ')) / SKNST1F('//CIX//','//CJX//')'
            CALL CONX(SKNST1(ILOADX,J),SKNST1A(ILOADX,J),SKNST1F(ILOADX,J)
1,'maximum stress in the shell skin, mode 1',
1 'allowable stress for the shell skin',
1 'factor of safety for skin stress',
1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLOTX,ICARX)
            IF (IMODX.EQ.0) THEN
                CODPHR =
1 ' maximum stress in the shell skin, mode 1: '
                IENDP4 =44
                CODNAM ='SKNST1('//CIX//','//CJX//')'
                MLET4 =6 + 7
                WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)

```

```

        IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1      KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
190 CONTINUE
191 CONTINUE
C
C Behavior and constraints generated next for STFST1:
C STFST1 = maximum stress in isogrid stiffener, mode 1
C
      IF (JSKNBK1.EQ.0) GO TO 206
      IF (NPRINX.GT.0) THEN
        IF (JSKNBK1.GT.1) THEN
          WRITE(IFILE8,'(1X,A)')' '
          WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
          WRITE(IFILE8,'(1X,A)')
1        'number of regions for computing behavior'
        ENDIF
      ENDIF
      DO 205 J=1,JSKNBK1
      CALL CONVR2(J,CJX)
      PHRASE =
1 'maximum stress in isogrid stiffener, mode 1'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(6 ).EQ.0) CALL BEHX6
1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
1 'maximum stress in isogrid stiffener, mode 1')
      IF (STFST1(ILOADX,J).EQ.0.) STFST1(ILOADX,J) = 1.E-10
      IF (STFST1A(ILOADX,J).EQ.0.) STFST1A(ILOADX,J) = 1.0
      IF (STFST1F(ILOADX,J).EQ.0.) STFST1F(ILOADX,J) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =STFST1(ILOADX,J)
      WORDCX= '(STFST1A('///CIX///','///CJX///')/STFST1('///CIX///','///CJX///
1 ')) / STFST1F('///CIX///','///CJX///')'
      CALL CONX(STFST1(ILOADX,J),STFST1A(ILOADX,J),STFST1F(ILOADX,J)
1,'maximum stress in isogrid stiffener, mode 1',
1 'allowable stress in isogrid stiffeners',
1 'factor of safety for stress in isogrid member',
1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
        CODPHR =
1 ' maximum stress in isogrid stiffener, mode 1: '
        IENDP4 =47
        CODNAM ='STFST1('///CIX///','///CJX///')'
        MLET4 =6 + 7
        WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
        IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1      KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)

```

```

        ENDIF
205 CONTINUE
206 CONTINUE
C
C Behavior and constraints generated next for WAPEx1:
C WAPEx1 = normal (axial) displacement at apex, mode 1
C
        PHRASE =
1 'normal (axial) displacement at apex, mode 1'
        CALL BLANKX(PHRASE,IENDP4)
        IF (IBEHV(7 ).EQ.0) CALL BEHX7
1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX ,
1 'normal (axial) displacement at apex, mode 1')
        IF (WAPEx1(ILOADX ).EQ.0.) WAPEx1(ILOADX ) = 1.E-10
        IF (WAPEx1A(ILOADX ).EQ.0.) WAPEx1A(ILOADX ) = 1.0
        IF (WAPEx1F(ILOADX ).EQ.0.) WAPEx1F(ILOADX ) = 1.0
        KCONX = KCONX + 1
        CARX(KCONX) =WAPEx1(ILOADX )
        WORDCX= '(WAPEx1A('//CIX//')/WAPEx1('//CIX//
1 ' ')) / WAPEx1F('//CIX//')'
        CALL CONX(WAPEx1(ILOADX ),WAPEx1A(ILOADX ),WAPEx1F(ILOADX )
1,'normal (axial) displacement at apex, mode 1',
1 'allowable normal (axial) displacement at apex',
1 'factor of safety for WAPEx',
1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLOTX,ICARX)
        IF (IMODX.EQ.0) THEN
            CODPHR =
1 ' normal (axial) displacement at apex, mode 1: '
            IENDP4 =47
            CODNAM ='WAPEx1('//CIX//')'
            MLET4 =6 + 4
            WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
            IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1 KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
        ENDIF
220 CONTINUE
221 CONTINUE
C
C Behavior and constraints generated next for CLAPS2:
C CLAPS2 = collapse pressure with imperfection mode 2
C
        PHRASE =
1 'collapse pressure with imperfection mode 2'
        CALL BLANKX(PHRASE,IENDP4)
        IF (IBEHV(8 ).EQ.0) CALL BEHX8
1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX ,
1 'collapse pressure with imperfection mode 2')

```

```

IF (CLAPS2(ILOADX ).EQ.0.) CLAPS2(ILOADX ) = 1.E+10
IF (CLAPS2A(ILOADX ).EQ.0.) CLAPS2A(ILOADX ) = 1.0
IF (CLAPS2F(ILOADX ).EQ.0.) CLAPS2F(ILOADX ) = 1.0
KCONX = KCONX + 1
CARX(KCONX) =CLAPS2(ILOADX )
WORDCX= '(CLAPS2('//CIX//')/CLAPS2A('//CIX//
1 ')) / CLAPS2F('//CIX//')'
CALL CONX(CLAPS2(ILOADX ),CLAPS2A(ILOADX ),CLAPS2F(ILOADX )
1,'collapse pressure with imperfection mode 2',
1 'allowable pressure for axisymmetric collapse',
1 'factor of safety for axisymmetric collapse',
1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLOTX,ICARX)
IF (IMODX.EQ.0) THEN
CODPHR =
1 ' collapse pressure with imperfection mode 2: '
IENDP4 =46
CODNAM ='CLAPS2('//CIX//')'
MLET4 =6 + 4
WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1 KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
ENDIF

```

235 CONTINUE

236 CONTINUE

C

C Behavior and constraints generated next for GENBK2:

C GENBK2 = general buckling load factor, mode 2

C

```

PHRASE =
1 'general buckling load factor, mode 2'
CALL BLANKX(PHRASE,IENDP4)
IF (IBEHV(9 ).EQ.0) CALL BEHX9
1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX ,
1 'general buckling load factor, mode 2')
IF (GENBK2(ILOADX ).EQ.0.) GENBK2(ILOADX ) = 1.E+10
IF (GENBK2A(ILOADX ).EQ.0.) GENBK2A(ILOADX ) = 1.0
IF (GENBK2F(ILOADX ).EQ.0.) GENBK2F(ILOADX ) = 1.0
KCONX = KCONX + 1
CARX(KCONX) =GENBK2(ILOADX )
WORDCX= '(GENBK2('//CIX//')/GENBK2A('//CIX//
1 ')) / GENBK2F('//CIX//')'
CALL CONX(GENBK2(ILOADX ),GENBK2A(ILOADX ),GENBK2F(ILOADX )
1,'general buckling load factor, mode 2',
1 'allowable general buckling load factor (use 1.0)',
1 'factor of safety for general buckling',
1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLOTX,ICARX)

```

```

      IF (IMODX.EQ.0) THEN
        CODPHR =
1 '   general buckling load factor, mode 2: '
        IENDP4 =40
        CODNAM ='GENBK2('//CIX//')'
        MLET4 =6 + 4
        WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
        IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1      KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
250 CONTINUE
251 CONTINUE
C
C Behavior and constraints generated next for SKNBK2:
C SKNBK2 = local skin buckling load factor, mode 2
C
      IF (JSKNBK2.EQ.0) GO TO 271
      IF (NPRINX.GT.0) THEN
        IF (JSKNBK2.GT.1) THEN
          WRITE(IFILE8,'(1X,A)') ' '
          WRITE(IFILE8,'(1X,A,$)') ' BEHAVIOR OVER J = '
          WRITE(IFILE8,'(1X,A)')
1      'number of regions for computing behavior'
        ENDIF
      ENDIF
      DO 270 J=1,JSKNBK2
        CALL CONVR2(J,CJX)
        PHRASE =
1 'local skin buckling load factor, mode 2'
        CALL BLANKX(PHRASE,IENDP4)
        IF (IBEHV(10).EQ.0) CALL BEHX10
1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
1 'local skin buckling load factor, mode 2')
        IF (SKNBK2(ILOADX,J).EQ.0.) SKNBK2(ILOADX,J) = 1.E+10
        IF (SKNBK2A(ILOADX,J).EQ.0.) SKNBK2A(ILOADX,J) = 1.0
        IF (SKNBK2F(ILOADX,J).EQ.0.) SKNBK2F(ILOADX,J) = 1.0
        KCONX = KCONX + 1
        CARX(KCONX) =SKNBK2(ILOADX,J)
        WORDCX= '(SKNBK2('//CIX//','//CJX//')/SKNBK2A('//CIX//','//CJX//
1 ')) / SKNBK2F('//CIX//','//CJX//')'
        CALL CONX(SKNBK2(ILOADX,J),SKNBK2A(ILOADX,J),SKNBK2F(ILOADX,J)
1,'local skin buckling load factor, mode 2',
1 'allowable skin buckling load factor (use 1.0)',
1 'factor of safety for local skin buckling',
1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
        CODPHR =

```

```

1 ' local skin buckling load factor, mode 2: '
  IENDP4 =43
  CODNAM ='SKNBK2('//CIX//','//CJX//')'
  MLET4 =6 + 7
  WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
  IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1   KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
  ENDIF
270 CONTINUE
271 CONTINUE
C
C Behavior and constraints generated next for STFBK2:
C STFBK2 = buckling load factor for isogrid member, mode 2
C
  IF (JSKNBK2.EQ.0) GO TO 286
  IF (NPRINX.GT.0) THEN
    IF (JSKNBK2.GT.1) THEN
      WRITE(IFILE8,'(1X,A)') ' '
      WRITE(IFILE8,'(1X,A,$)') ' BEHAVIOR OVER J = '
      WRITE(IFILE8,'(1X,A)')
1      'number of regions for computing behavior'
    ENDIF
  ENDIF
  DO 285 J=1,JSKNBK2
    CALL CONVR2(J,CJX)
    PHRASE =
1 'buckling load factor for isogrid member, mode 2'
    CALL BLANKX(PHRASE,IENDP4)
    IF (IBEHV(11).EQ.0) CALL BEHX11
1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
1 'buckling load factor for isogrid member, mode 2')
    IF (STFBK2(ILOADX,J).EQ.0.) STFBK2(ILOADX,J) = 1.E+10
    IF (STFBK2A(ILOADX,J).EQ.0.) STFBK2A(ILOADX,J) = 1.0
    IF (STFBK2F(ILOADX,J).EQ.0.) STFBK2F(ILOADX,J) = 1.0
    KCONX = KCONX + 1
    CARX(KCONX) =STFBK2(ILOADX,J)
    WORDCX= '(STFBK2('//CIX//','//CJX//')/STFBK2A('//CIX//','//CJX//
1 ')) / STFBK2F('//CIX//','//CJX//')'
    CALL CONX(STFBK2(ILOADX,J),STFBK2A(ILOADX,J),STFBK2F(ILOADX,J)
1,'buckling load factor for isogrid member, mode 2',
1 'allowable for isogrid stiffener buckling (Use 1.)',
1 'factor of safety for isogrid stiffener buckling',
1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLOTX,ICARX)
    IF (IMODX.EQ.0) THEN
      CODPHR =
1 ' buckling load factor for isogrid member, mode 2: '
      IENDP4 =51

```

```

        CODNAM = 'STFBK2('//CIX//','//CJX//')'
        MLET4 = 6 + 7
        WORDBX(KCONX) = CODPHR(1:IENDP4)//CODNAM(1:MLET4)
        IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1       KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
        ENDIF
285 CONTINUE
286 CONTINUE
C
C Behavior and constraints generated next for SKNST2:
C SKNST2 = maximum stress in the shell skin, mode 2
C
        IF (JSKNBK2.EQ.0) GO TO 301
        IF (NPRINX.GT.0) THEN
            IF (JSKNBK2.GT.1) THEN
                WRITE(IFILE8,'(1X,A)') ' '
                WRITE(IFILE8,'(1X,A,$)') ' BEHAVIOR OVER J = '
                WRITE(IFILE8,'(1X,A)')
1         'number of regions for computing behavior'
            ENDIF
        ENDIF
        DO 300 J=1,JSKNBK2
            CALL CONVR2(J,CJX)
            PHRASE =
1         'maximum stress in the shell skin, mode 2'
            CALL BLANKX(PHRASE,IENDP4)
            IF (IBEHV(12).EQ.0) CALL BEHX12
1         (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
1         'maximum stress in the shell skin, mode 2')
            IF (SKNST2(ILOADX,J).EQ.0.) SKNST2(ILOADX,J) = 1.E-10
            IF (SKNST2A(ILOADX,J).EQ.0.) SKNST2A(ILOADX,J) = 1.0
            IF (SKNST2F(ILOADX,J).EQ.0.) SKNST2F(ILOADX,J) = 1.0
            KCONX = KCONX + 1
            CARX(KCONX) = SKNST2(ILOADX,J)
            WORDCX = ' (SKNST2A('//CIX//','//CJX//')/SKNST2('//CIX//','//CJX//
1         ')) / SKNST2F('//CIX//','//CJX//')'
            CALL CONX(SKNST2(ILOADX,J),SKNST2A(ILOADX,J),SKNST2F(ILOADX,J)
1         'maximum stress in the shell skin, mode 2',
1         'allowable stress for the shell skin',
1         'factor of safety for skin stress',
1         3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1         WORDMX,PCWORD,CPLOTX,ICARX)
            IF (IMODX.EQ.0) THEN
                CODPHR =
1         ' maximum stress in the shell skin, mode 2: '
                IENDP4 = 44
                CODNAM = 'SKNST2('//CIX//','//CJX//')'
                MLET4 = 6 + 7

```

```

        WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
        IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1      KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
300 CONTINUE
301 CONTINUE
C
C Behavior and constraints generated next for STFST2:
C STFST2 = maximum stress in isogrid stiffener, mode 2
C
      IF (JSKNBK2.EQ.0) GO TO 316
      IF (NPRINX.GT.0) THEN
        IF (JSKNBK2.GT.1) THEN
          WRITE(IFILE8,'(1X,A)')' '
          WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
          WRITE(IFILE8,'(1X,A)')
1        'number of regions for computing behavior'
        ENDIF
      ENDIF
      DO 315 J=1,JSKNBK2
      CALL CONVR2(J,CJX)
      PHRASE =
1 'maximum stress in isogrid stiffener, mode 2'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(13 ).EQ.0) CALL BEHX13
1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
1 'maximum stress in isogrid stiffener, mode 2')
      IF (STFST2(ILOADX,J).EQ.0.) STFST2(ILOADX,J) = 1.E-10
      IF (STFST2A(ILOADX,J).EQ.0.) STFST2A(ILOADX,J) = 1.0
      IF (STFST2F(ILOADX,J).EQ.0.) STFST2F(ILOADX,J) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =STFST2(ILOADX,J)
      WORDCX= '(STFST2A('//CIX//','//CJX//')/STFST2('//CIX//','//CJX//
1 ')) / STFST2F('//CIX//','//CJX//')'
      CALL CONX(STFST2(ILOADX,J),STFST2A(ILOADX,J),STFST2F(ILOADX,J)
1,'maximum stress in isogrid stiffener, mode 2',
1 'allowable stress in isogrid stiffeners',
1 'factor of safety for stress in isogrid member',
1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLTX,ICARX)
      IF (IMODX.EQ.0) THEN
        CODPHR =
1 ' maximum stress in isogrid stiffener, mode 2: '
        IENDP4 =47
        CODNAM ='STFST2('//CIX//','//CJX//')'
        MLET4 =6 + 7
        WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
        IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')

```



```

1      KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
315 CONTINUE
316 CONTINUE
C
C Behavior and constraints generated next for WAPEx2:
C WAPEx2 = normal (axial) displacement at apex, mode 2
C
      PHRASE =
1 'normal (axial) displacement at apex, mode 2'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(14).EQ.0) CALL BEHX14
1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX ,
1 'normal (axial) displacement at apex, mode 2')
      IF (WAPEx2(ILOADX).EQ.0.) WAPEx2(ILOADX) = 1.E-10
      IF (WAPEx2A(ILOADX).EQ.0.) WAPEx2A(ILOADX) = 1.0
      IF (WAPEx2F(ILOADX).EQ.0.) WAPEx2F(ILOADX) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) = WAPEx2(ILOADX)
      WORDCX= ' (WAPEx2A('//CIX//')/WAPEx2('//CIX//
1 ' ')) / WAPEx2F('//CIX//')'
      CALL CONX(WAPEx2(ILOADX),WAPEx2A(ILOADX),WAPEx2F(ILOADX) )
1,'normal (axial) displacement at apex, mode 2',
1 'allowable normal (axial) displacement at apex',
1 'factor of safety for WAPEx',
1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
          CODPHR =
1 ' normal (axial) displacement at apex, mode 2: '
          IENDP4 =47
          CODNAM ='WAPEx2('//CIX//')'
          MLET4 =6 + 4
          WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
          IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
1      KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
330 CONTINUE
331 CONTINUE
C
C NEXT, EVALUATE THE OBJECTIVE, OBJGEN:
      IF (ILOADX.EQ.1) THEN
          PHRASE ='weight of the equivalent ellipsoidal head'
          CALL BLANKX(PHRASE,IENDP4)
          CALL OBJECT(IFILE8,NPRINX,IMODX,OBJGEN,
1 'weight of the equivalent ellipsoidal head')
          ENDIF
          NCONSX = ICONSX

```

```

C
C
      RETURN
      END
C
C
C
C
C
C
C  End of the final portion of STRUCT written by "GENTEXT"
C=====
C
C=DECK      TRANFR
      SUBROUTINE TRANFR(ARG1,ARG2,ARG3,ARG4,ARG5)
C
C  USER:  DO NOT FORGET TO MODIFY THE ARGUMENT LIST OF TRANFR AS
C          APPROPRIATE FOR YOUR CASE!
C
C  PURPOSE IS TO TRANSFER DATA FROM THE LABELLED COMMON BLOCKS
C  SET UP BY THE GENOPT CODE TO LABELLED COMMON OR ARGUMENTS IN
C  THE SUBROUTINE ARGUMENT LIST THAT MATCH PREVIOUSLY WRITTEN CODE
C  BY YOURSELF OR OTHER PROGRAM DEVELOPERS.  THE USER SHOULD ESTABLISH
C  THE ARGUMENT LIST AND/OR LABELLED COMMON BLOCKS THAT MATCH VARIABLES
C  IN THE PREVIOUSLY WRITTEN CODE.  FOR AN EXAMPLE, SEE THE DISCUSSION
C  OF THE CASE CALLED "PANEL".
C
C=====
C
C  Start of part of TRANFR written by "GENTEXT"
C  INSERT ADDITIONAL COMMON BLOCKS HERE: (THESE ARE "GENTEXT" VARIABLES)
      COMMON/FV01/xinput(21),Ixinput
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F

```

```

COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
REAL STFST1,STFST1A,STFST1F
COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
REAL WAPEX1,WAPEX1A,WAPEX1F
COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
REAL CLAPS2,CLAPS2A,CLAPS2F
COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
REAL GENBK2,GENBK2A,GENBK2F
COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
REAL SKNBK2,SKNBK2A,SKNBK2F
COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
REAL STFBK2,STFBK2A,STFBK2F
COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
REAL SKNST2,SKNST2A,SKNST2F
COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
REAL STFST2,STFST2A,STFST2F
COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
REAL WAPEX2,WAPEX2A,WAPEX2F
COMMON/IV01/npoin, nodes, IMODE
INTEGER npoin, nodes, IMODE
COMMON/FV11/LENCYL, WIMP, EMATL, NUMATL, DNMATL, WEIGHT
REAL LENCYL, WIMP, EMATL, NUMATL, DNMATL, WEIGHT

```

C

C

C End of part of TRANFR written by "GENTEXT"

C=====

=

C INSERT ADDITIONAL DIMENSION AND/OR LABELLED COMMON BLOCKS HERE,  
C IF NECESSARY. THESE WOULD BE STATEMENTS THAT ARE CONSISTENT WITH  
C SUBROUTINES THAT YOU OR OTHERS MAY HAVE WRITTEN THAT ARE REQUIRED  
C FOR WHATEVER ANALYSIS YOU ARE NOW PERSUING. MAKE SURE THERE ARE  
C NO NAME CONFLICTS WITH THE "GENTEXT" LABELLED COMMON BLOCKS.

C

C

C INSERT APPROPRIATE FORTRAN STATEMENTS HERE (DON'T FORGET TO CORRECT  
C THE ARGUMENT LIST OF SUBROUTINE TRANFR!)  
C PROGRAM FILE:

C

C

RETURN

END

=====