Table A30 list of the file, **/home/progs/genopt/case/cylinder/howto.struct.** The purpose of this file is to provide the GENOPT user with some additional documentation about "**fleshing out**" the **skeletal struct.new** file created automatically by GENTEXT in cases in which BIGBOSOR4 software is being used to compute behaviors, such as stress, buckling, displacement. In this particular case (**/home/progs/genopt/case/cylinder**) not much is done to "**flesh out**" the **skeletal struct.new.** For cases in which much more is done to "flesh out" the skeletal struct.new, please see the files, /home/progs/genopt/case/wavycyl/struct.wavycyl [7] and /home/progs/genopt/case/torisph/struct.equivellipse (Table a16 here).
==========

10 March, 2008; modified September 26, 2008

********** NOTE **********
In the following the string, "/home/progs", frequently occurs. This is the PARENT directory of BOSOR4, BIGBOSOR4, BOSOR5, PANDA2, and GENOPT on the writer's computer. You must replace the string, "/home/progs", with whatever is the PARENT directory of BOSOR4, BIGBOSOR4, BOSOR5, PANDA2, and GENOPT at your facility.
********** END NOTE **********

Please read Section A.5.1 and Table a.5 of the appendix of the report:

Bushnell, D., Automated optimum design of shells of revolution with application to ring-stiffened cylindrical shells with wavy walls, Report LMMS P525674, November 1999. Section A.5.1 is on pp 46-47 of that report.

This text gives some guidance about how to modify ("flesh out") the skeletal file, struct.new, produced automatically by GENTEXT.

In this discussion there are three versions of "struct"

1. struct.tmpl (an initial skeletal "struct" file that is the same for all generic GENOPT cases. This file is stored in the directory, /home/progs/genopt/sources .)

2. struct.new (a second skeletal "struct" file which starts with struct.tmpl but includes statements that are added to struct.tmpl by the GENTEXT processor for the generic case set forth by the GENOPT user. This is the version of "struct" that the GENOPT user may need to augment with FORTRAN statements appropriate for his generic case. This skeletal "struct" file is stored in the directory, /home/progs/genoptcase, after the GENOPT

user's completion of the interactive GENTEXT session. It is called "struct.new".)

3. struct.new (the same as 2. except that now the GENOPT user has added FORTRAN statements pertaining to his generic case. This "final" version of "struct" is stored in the directory, /home/progs/genoptcase, and must be saved by the GENOPT user elsewhere, such as in this case as follows:
**cp /home/progs/genoptcase/struct.new**
**/home/progs/genopt/case/cylinder/struct.cylinder)**

The initial skeletal file for "struct", located in the directory, /home/progs/genopt/sources and called "**struct.tmpl**", is as follows:

**-----BEGINNING OF THE initial skeletal "struct" file, struct.tmpl ------**

```
C=DECK      STRUCT
      SUBROUTINE STRUCT(IMODX,CONSTX,OBJGEN,CONMAX,NCONSX,IPOINC,
     1 PCWORD,CPLOTX,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)
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
c     CHARACTER*4 ANSOUT,CHARAC,ANSWER
      CHARACTER*2 CIX
      character*2 CJX
      CHARACTER*13 CODNAM
c     DIMENSION ISUBX(100)
c     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 will
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
      ICARX   = ISTARX
      INUMTT = 0
      ICONSX = 0
      KCONX   = 0
      IF (IMODX.EQ.0) THEN
         CALL MOVERX(0.,0,CONSTX,1,99)
         CALL MOVERX(0, 0,IPOINC,1,1500)
      ENDIF
```

```
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
         IF (NINEQX.GT.0)
     1      CALL VARCON(WORDIQ,WORDMX,CINEQX,DPWREQ,IDINEQ,
     1      NINEQX,JINEQX,IEQTYP,INUMTT,IMODX,CONMAX,IPOINC,
     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 USRCON (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 USRCON IMMEDIATELY BELOW.
C
         CALL USRCON(INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1           WORDMX,PCWORD,CPLOTX,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
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  Start of part of TRANFR written by "GENTEXT"
C  INSERT ADDITIONAL COMMON BLOCKS HERE: (THESE ARE "GENTEXT" VARIABLES)
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
-------END OF THE initial skeletal "struct" file, struct.tmpl ----------
```

**Execution of GENTEXT automatically generates additional lines in the above, producing a skeletal file called "struct.new", as follows:**

**--- BEGINNING OF THE FILE "struct.new" after execution of GENTEXT ----**

```
C=DECK      STRUCT
      SUBROUTINE STRUCT(IMODX,CONSTX,OBJGEN,CONMAX,NCONSX,IPOINC,
     1 PCWORD,CPLOTX,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/FV07/NX(20)
      REAL NX
      COMMON/FV11/STRESS(20),STRSSA(20),STRSSF(20)
      REAL STRESS,STRSSA,STRSSF
      COMMON/FV14/BSYM(20),BSYMA(20),BSYMF(20)
      REAL BSYM,BSYMA,BSYMF
      COMMON/FV17/BANTI(20),BANTIA(20),BANTIF(20)
      REAL BANTI,BANTIA,BANTIF
```

```
      COMMON/FV20/FREQ(20),FREQA(20),FREQF(20)
      REAL FREQ,FREQA,FREQF
      COMMON/IV01/IBOUND
      INTEGER IBOUND
      COMMON/FV01/LENGTH,RADIUS,THICK,ESTIFF,NU,DENS,WEIGHT
      REAL LENGTH,RADIUS,THICK,ESTIFF,NU,DENS,WEIGHT
      COMMON/FV08/PRESS(20)
      REAL PRESS
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
c     CHARACTER*4 ANSOUT,CHARAC,ANSWER
      CHARACTER*2 CIX
      character*2 CJX
      CHARACTER*13 CODNAM
c     DIMENSION ISUBX(100)
c     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
      ICARX   = ISTARX
      INUMTT = 0
      ICONSX = 0
```

```
      KCONX   = 0
      IF (IMODX.EQ.0) THEN
         CALL MOVERX(0.,0,CONSTX,1,99)
         CALL MOVERX(0, 0,IPOINC,1,1500)
      ENDIF
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
         IF (NINEQX.GT.0)
     1        CALL VARCON(WORDIQ,WORDMX,CINEQX,DPWREQ,IDINEQ,
     1        NINEQX,JINEQX,IEQTYP,INUMTT,IMODX,CONMAX,IPOINC,
     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 USRCON (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 USRCON IMMEDIATELY BELOW.
C
         CALL USRCON(INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1            WORDMX,PCWORD,CPLOTX,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 STRESS:
C  STRESS = Maximum effective stress in wall of shell
C
      PHRASE =
     1 'Maximum effective stress in wall of shell'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(1  ).EQ.0) CALL BEHX1
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'Maximum effective stress in wall of shell')
      IF (STRESS(ILOADX  ).EQ.0.)  STRESS(ILOADX  ) = 1.E-10
      IF (STRSSA(ILOADX  ).EQ.0.)  STRSSA(ILOADX  ) = 1.0
      IF (STRSSF(ILOADX  ).EQ.0.)  STRSSF(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =STRESS(ILOADX  )
      WORDCX= '(STRESS('//CIX//')/STRSSA('//CIX//
     1  ')) X STRSSF('//CIX//')'
      CALL CONX(STRESS(ILOADX  ),STRSSA(ILOADX  ),STRSSF(ILOADX  )
     1,'Maximum effective stress in wall of shell',
     1 'Maximum allowable stress',
     1 'Factor of safety for stress',
     1 1,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  Maximum effective stress in wall of shell: '
         IENDP4 =45
         CODNAM ='STRESS('//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
   70 CONTINUE
   71 CONTINUE
C
C  Behavior and constraints generated next for BSYM:
C  BSYM = Symmetric buckling load factor
```

```
C
      PHRASE =
     1 'Symmetric buckling load factor'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(2  ).EQ.0) CALL BEHX2
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'Symmetric buckling load factor')
      IF (BSYM(ILOADX  ).EQ.0.)  BSYM(ILOADX  ) = 1.E+10
      IF (BSYMA(ILOADX  ).EQ.0.)  BSYMA(ILOADX  ) = 1.0
      IF (BSYMF(ILOADX  ).EQ.0.)  BSYMF(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =BSYM(ILOADX  )
      WORDCX= '(BSYM('//CIX//')/BSYMA('//CIX//
     1  ')) / BSYMF('//CIX//')'
      CALL CONX(BSYM(ILOADX  ),BSYMA(ILOADX  ),BSYMF(ILOADX  )
     1,'Symmetric buckling load factor',
     1 'Allowable for sym. buckling load factor',
     1 'Factor of safety for sym. buckling load',
     1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  Symmetric buckling load factor: '
         IENDP4 =34
         CODNAM ='BSYM('//CIX//')'
         MLET4 =4 + 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
   85 CONTINUE
   86 CONTINUE
C
C  Behavior and constraints generated next for BANTI:
C  BANTI = Antisymmetric buckling load factor
C
      PHRASE =
     1 'Antisymmetric buckling load factor'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(3  ).EQ.0) CALL BEHX3
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'Antisymmetric buckling load factor')
      IF (BANTI(ILOADX  ).EQ.0.)  BANTI(ILOADX  ) = 1.E+10
      IF (BANTIA(ILOADX  ).EQ.0.)  BANTIA(ILOADX  ) = 1.0
      IF (BANTIF(ILOADX  ).EQ.0.)  BANTIF(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =BANTI(ILOADX  )
      WORDCX= '(BANTI('//CIX//')/BANTIA('//CIX//
```

```
     1  ')) / BANTIF('//CIX//')'
      CALL CONX(BANTI(ILOADX  ),BANTIA(ILOADX  ),BANTIF(ILOADX  )
     1,'Antisymmetric buckling load factor',
     1 'Allowable for antisym. buckling load factor',
     1 'Factor of safety for antisym. buckling load',
     1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  Antisymmetric buckling load factor: '
         IENDP4 =38
         CODNAM ='BANTI('//CIX//')'
         MLET4 =5 + 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
  100 CONTINUE
  101 CONTINUE
C
C  Behavior and constraints generated next for FREQ:
C  FREQ = Fundamental modal frequency (hertz)
C
      PHRASE =
     1 'Fundamental modal frequency (hertz)'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(4  ).EQ.0) CALL BEHX4
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'Fundamental modal frequency (hertz)')
      IF (FREQ(ILOADX  ).EQ.0.)  FREQ(ILOADX  ) = 1.E+10
      IF (FREQA(ILOADX  ).EQ.0.)  FREQA(ILOADX  ) = 1.0
      IF (FREQF(ILOADX  ).EQ.0.)  FREQF(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =FREQ(ILOADX  )
      WORDCX= '(FREQ('//CIX//')/FREQA('//CIX//
     1  ')) / FREQF('//CIX//')'
      CALL CONX(FREQ(ILOADX  ),FREQA(ILOADX  ),FREQF(ILOADX  )
     1,'Fundamental modal frequency (hertz)',
     1 'Allowable for modal frequency',
     1 'Factor of safety for modal frequency',
     1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  Fundamental modal frequency (hertz): '
         IENDP4 =39
         CODNAM ='FREQ('//CIX//')'
         MLET4 =4 + 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
  115 CONTINUE
  116 CONTINUE
C
C  NEXT, EVALUATE THE OBJECTIVE, OBJGEN:
       IF (ILOADX.EQ.1) THEN
         PHRASE ='weight of half of cyl. shell'
         CALL BLANKX(PHRASE,IENDP4)
         CALL OBJECT(IFILE8,NPRINX,IMODX,OBJGEN,
     1    'weight of half of cyl. shell')
       ENDIF
       NCONSX = ICONSX
C
C
       RETURN
       END
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  Start of part of TRANFR written by "GENTEXT"
C  INSERT ADDITIONAL COMMON BLOCKS HERE: (THESE ARE "GENTEXT" VARIABLES)
       COMMON/FV07/NX(20)
       REAL NX
       COMMON/FV11/STRESS(20),STRSSA(20),STRSSF(20)
       REAL STRESS,STRSSA,STRSSF
```

```
      COMMON/FV14/BSYM(20),BSYMA(20),BSYMF(20)
      REAL BSYM,BSYMA,BSYMF
      COMMON/FV17/BANTI(20),BANTIA(20),BANTIF(20)
      REAL BANTI,BANTIA,BANTIF
      COMMON/FV20/FREQ(20),FREQA(20),FREQF(20)
      REAL FREQ,FREQA,FREQF
      COMMON/IV01/IBOUND
      INTEGER IBOUND
      COMMON/FV01/LENGTH,RADIUS,THICK,ESTIFF,NU,DENS,WEIGHT
      REAL LENGTH,RADIUS,THICK,ESTIFF,NU,DENS,WEIGHT
      COMMON/FV08/PRESS(20)
      REAL PRESS
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
C
C
C
```

**--- END OF THE SKELETAL FILE "struct.new" after execution of GENTEXT and before the GENOPT user has added any additional FORTRAN lines pertaining to his/her generic case ----**

**The difference between the struct.tmpl file (the initial skeletal "struct" file BEFORE execution of GENTEXT) and the struct.new file (the skeletal "struct" file AFTER execution of GENTEXT is obtained via the command:**

**/home/progs/genoptcase/struct.new /home/progs/genopt/sources/struct.tmpl >**

**struct.diff**

**The struct.diff file follows:**

**---------- BEGINNING OF struct.diff ----------**

```
35,50d34
<       COMMON/FV07/NX(20)
<       REAL NX
<       COMMON/FV11/STRESS(20),STRSSA(20),STRSSF(20)
<       REAL STRESS,STRSSA,STRSSF
<       COMMON/FV14/BSYM(20),BSYMA(20),BSYMF(20)
<       REAL BSYM,BSYMA,BSYMF
<       COMMON/FV17/BANTI(20),BANTIA(20),BANTIF(20)
<       REAL BANTI,BANTIA,BANTIF
<       COMMON/FV20/FREQ(20),FREQA(20),FREQF(20)
<       REAL FREQ,FREQA,FREQF
<       COMMON/IV01/IBOUND
<       INTEGER IBOUND
<       COMMON/FV01/LENGTH,RADIUS,THICK,ESTIFF,NU,DENS,WEIGHT
<       REAL LENGTH,RADIUS,THICK,ESTIFF,NU,DENS,WEIGHT
<       COMMON/FV08/PRESS(20)
<       REAL PRESS
198,353d181
< C  Behavior and constraints generated next for STRESS:
< C  STRESS = Maximum effective stress in wall of shell
< C
<       PHRASE =
<      1 'Maximum effective stress in wall of shell'
<       CALL BLANKX(PHRASE,IENDP4)
<       IF (IBEHV(1  ).EQ.0) CALL BEHX1
<      1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
<      1 'Maximum effective stress in wall of shell')
<       IF (STRESS(ILOADX  ).EQ.0.)  STRESS(ILOADX  ) = 1.E-10
<       IF (STRSSA(ILOADX  ).EQ.0.)  STRSSA(ILOADX  ) = 1.0
<       IF (STRSSF(ILOADX  ).EQ.0.)  STRSSF(ILOADX  ) = 1.0
<       KCONX = KCONX + 1
<       CARX(KCONX) =STRESS(ILOADX  )
<       WORDCX= '(STRESS('//CIX//')/STRSSA('//CIX//
<      1  ')) X STRSSF('//CIX//')'
<       CALL CONX(STRESS(ILOADX  ),STRSSA(ILOADX  ),STRSSF(ILOADX  )
<      1,'Maximum effective stress in wall of shell',
<      1 'Maximum allowable stress',
<      1 'Factor of safety for stress',
<      1 1,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
<      1 WORDMX,PCWORD,CPLOTX,ICARX)
<       IF (IMODX.EQ.0) THEN
<          CODPHR =
<      1 '  Maximum effective stress in wall of shell: '
<          IENDP4 =45
```

```
<           CODNAM ='STRESS('//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
<       70 CONTINUE
<       71 CONTINUE
< C
< C  Behavior and constraints generated next for BSYM:
< C  BSYM = Symmetric buckling load factor
< C
<          PHRASE =
<         1 'Symmetric buckling load factor'
<          CALL BLANKX(PHRASE,IENDP4)
<          IF (IBEHV(2  ).EQ.0) CALL BEHX2
<         1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
<         1 'Symmetric buckling load factor')
<          IF (BSYM(ILOADX  ).EQ.0.)  BSYM(ILOADX  ) = 1.E+10
<          IF (BSYMA(ILOADX  ).EQ.0.)  BSYMA(ILOADX  ) = 1.0
<          IF (BSYMF(ILOADX  ).EQ.0.)  BSYMF(ILOADX  ) = 1.0
<          KCONX = KCONX + 1
<          CARX(KCONX) =BSYM(ILOADX  )
<          WORDCX= '(BSYM('//CIX//')/BSYMA('//CIX//
<         1  ')) / BSYMF('//CIX//')'
<          CALL CONX(BSYM(ILOADX  ),BSYMA(ILOADX  ),BSYMF(ILOADX  )
<         1,'Symmetric buckling load factor',
<         1 'Allowable for sym. buckling load factor',
<         1 'Factor of safety for sym. buckling load',
<         1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
<         1 WORDMX,PCWORD,CPLOTX,ICARX)
<          IF (IMODX.EQ.0) THEN
<             CODPHR =
<         1 '  Symmetric buckling load factor: '
<             IENDP4 =34
<             CODNAM ='BSYM('//CIX//')'
<             MLET4 =4 + 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
<       85 CONTINUE
<       86 CONTINUE
< C
< C  Behavior and constraints generated next for BANTI:
< C  BANTI = Antisymmetric buckling load factor
< C
<          PHRASE =
```

```
<      1 'Antisymmetric buckling load factor'
<       CALL BLANKX(PHRASE,IENDP4)
<       IF (IBEHV(3  ).EQ.0) CALL BEHX3
<      1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
<      1 'Antisymmetric buckling load factor')
<       IF (BANTI(ILOADX  ).EQ.0.)  BANTI(ILOADX  ) = 1.E+10
<       IF (BANTIA(ILOADX  ).EQ.0.)  BANTIA(ILOADX  ) = 1.0
<       IF (BANTIF(ILOADX  ).EQ.0.)  BANTIF(ILOADX  ) = 1.0
<       KCONX = KCONX + 1
<       CARX(KCONX) =BANTI(ILOADX  )
<       WORDCX= '(BANTI('//CIX//')/BANTIA('//CIX//
<      1  ')) / BANTIF('//CIX//')'
<       CALL CONX(BANTI(ILOADX  ),BANTIA(ILOADX  ),BANTIF(ILOADX  )
<      1,'Antisymmetric buckling load factor',
<      1 'Allowable for antisym. buckling load factor',
<      1 'Factor of safety for antisym. buckling load',
<      1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
<      1 WORDMX,PCWORD,CPLOTX,ICARX)
<       IF (IMODX.EQ.0) THEN
<          CODPHR =
<      1 '  Antisymmetric buckling load factor: '
<          IENDP4 =38
<          CODNAM ='BANTI('//CIX//')'
<          MLET4 =5 + 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
<   100 CONTINUE
<   101 CONTINUE
< C
< C  Behavior and constraints generated next for FREQ:
< C  FREQ = Fundamental modal frequency (hertz)
< C
<       PHRASE =
<      1 'Fundamental modal frequency (hertz)'
<       CALL BLANKX(PHRASE,IENDP4)
<       IF (IBEHV(4  ).EQ.0) CALL BEHX4
<      1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
<      1 'Fundamental modal frequency (hertz)')
<       IF (FREQ(ILOADX  ).EQ.0.)  FREQ(ILOADX  ) = 1.E+10
<       IF (FREQA(ILOADX  ).EQ.0.)  FREQA(ILOADX  ) = 1.0
<       IF (FREQF(ILOADX  ).EQ.0.)  FREQF(ILOADX  ) = 1.0
<       KCONX = KCONX + 1
<       CARX(KCONX) =FREQ(ILOADX  )
<       WORDCX= '(FREQ('//CIX//')/FREQA('//CIX//
<      1  ')) / FREQF('//CIX//')'
<       CALL CONX(FREQ(ILOADX  ),FREQA(ILOADX  ),FREQF(ILOADX  )
```

```
<       1,'Fundamental modal frequency (hertz)',
<       1 'Allowable for modal frequency',
<       1 'Factor of safety for modal frequency',
<       1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
<       1 WORDMX,PCWORD,CPLOTX,ICARX)
<        IF (IMODX.EQ.0) THEN
<           CODPHR =
<       1 '  Fundamental modal frequency (hertz): '
<           IENDP4 =39
<           CODNAM ='FREQ('//CIX//')'
<           MLET4 =4 + 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
<    115 CONTINUE
<    116 CONTINUE
< C
< C  NEXT, EVALUATE THE OBJECTIVE, OBJGEN:
<        IF (ILOADX.EQ.1) THEN
<           PHRASE ='weight of half of cyl. shell'
<           CALL BLANKX(PHRASE,IENDP4)
<           CALL OBJECT(IFILE8,NPRINX,IMODX,OBJGEN,
<       1    'weight of half of cyl. shell')
<        ENDIF
<        NCONSX = ICONSX
< C
< C
<        RETURN
<        END
< C
< C
< C
< C
375,390d202
<        COMMON/FV07/NX(20)
<        REAL NX
<        COMMON/FV11/STRESS(20),STRSSA(20),STRSSF(20)
<        REAL STRESS,STRSSA,STRSSF
<        COMMON/FV14/BSYM(20),BSYMA(20),BSYMF(20)
<        REAL BSYM,BSYMA,BSYMF
<        COMMON/FV17/BANTI(20),BANTIA(20),BANTIF(20)
<        REAL BANTI,BANTIA,BANTIF
<        COMMON/FV20/FREQ(20),FREQA(20),FREQF(20)
<        REAL FREQ,FREQA,FREQF
<        COMMON/IV01/IBOUND
<        INTEGER IBOUND
<        COMMON/FV01/LENGTH,RADIUS,THICK,ESTIFF,NU,DENS,WEIGHT
```

```
<       REAL LENGTH,RADIUS,THICK,ESTIFF,NU,DENS,WEIGHT
<       COMMON/FV08/PRESS(20)
<       REAL PRESS
406,411d217
< C
<       RETURN
<       END
< C
< C
< C
```
**---------- end of the struct.diff file ----------**

This difference is produced automatically by GENTEXT: the GENOPT user doesn't have to do anything. The difference is due to the addition to struct.tmpl by GENTEXT of the labeled common blocks, that is, the *.COM file and the addition to struct.tmpl of the FORTRAN fragment in which the design constraints are computed, that is, the *.CON file.

**The “fleshed out” struct.new file,**
**(/home/progs/genopt/case/cylinder/struct.cylinder),**
**which is completed by the GENOPT user for the generic case called "CYLINDER" is as follows. The statements added by the GENOPT user are written in boldface.**

**------ BEGINNING OF THE completed (“fleshed out”) struct.new file -----**

```
C=DECK      STRUCT
      SUBROUTINE STRUCT(IMODX,CONSTX,OBJGEN,CONMAX,NCONSX,IPOINC,
     1 PCWORD,CPLOTX,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/FV07/NX(20)
      REAL NX
      COMMON/FV11/STRESS(20),STRSSA(20),STRSSF(20)
      REAL STRESS,STRSSA,STRSSF
      COMMON/FV14/BSYM(20),BSYMA(20),BSYMF(20)
      REAL BSYM,BSYMA,BSYMF
      COMMON/FV17/BANTI(20),BANTIA(20),BANTIF(20)
      REAL BANTI,BANTIA,BANTIF
      COMMON/FV20/FREQ(20),FREQA(20),FREQF(20)
      REAL FREQ,FREQA,FREQF
      COMMON/IV01/IBOUND
      INTEGER IBOUND
      COMMON/FV01/LENGTH,RADIUS,THICK,ESTIFF,NU,DENS,WEIGHT
      REAL LENGTH,RADIUS,THICK,ESTIFF,NU,DENS,WEIGHT
      COMMON/FV08/PRESS(20)
      REAL PRESS
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
c     CHARACTER*4 ANSOUT,CHARAC,ANSWER
      CHARACTER*2 CIX
      character*2 CJX
      CHARACTER*13 CODNAM
c     DIMENSION ISUBX(100)
c     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
      COMMON/TOTMAX/TOTMAS
C
C  THE FOLLOWING CODE WAS WRITTEN BY "GENTEXT":
C
C=================================================================
C  Start the second portion of STRUCT written by "GENTEXT":
C
      ICARX   = ISTARX
      INUMTT = 0
      ICONSX = 0
      KCONX   = 0
      IF (IMODX.EQ.0) THEN
         CALL MOVERX(0.,0,CONSTX,1,99)
         CALL MOVERX(0, 0,IPOINC,1,1500)
      ENDIF
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
         IF (NINEQX.GT.0)
     1        CALL VARCON(WORDIQ,WORDMX,CINEQX,DPWREQ,IDINEQ,
     1        NINEQX,JINEQX,IEQTYP,INUMTT,IMODX,CONMAX,IPOINC,
     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 USRCON (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 USRCON IMMEDIATELY BELOW.
C
         CALL USRCON(INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1            WORDMX,PCWORD,CPLOTX,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
      CALL OPNGEN
      CALL RWDGEN
C
C     initialize behaviors:
      STRESS(ILOADX) = 0.
      BSYM(ILOADX) = 0.
      BANTI(ILOADX) = 0.
      FREQ(ILOADX)   = 0.
C
C  Find mass of cyl. shell from boundary to mid-length symmetry plane.
C  The mass is stored in TOTMAS, which is one of the BOSOR4 labelled
C  common blocks.
      INDIC = 0
C BEG FEB 2008
```

```
C     CALL BOSDEC(5,ILOADX,INDIC)
      CALL BOSDEC(5,24,ILOADX,INDIC)
      CALL B4READ
      CALL GASP (DUM1,DUM2,-2,DUM3)
      GRAVTY = 386.4
      WEIGHT = GRAVTY*TOTMAS
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 STRESS:
C  STRESS = Maximum effective stress in wall of shell
C
      PHRASE =
     1 'Maximum effective stress in wall of shell'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(1  ).EQ.0) CALL BEHX1
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'Maximum effective stress in wall of shell')
      IF (STRESS(ILOADX  ).EQ.0.)  STRESS(ILOADX  ) = 1.E-10
      IF (STRSSA(ILOADX  ).EQ.0.)  STRSSA(ILOADX  ) = 1.0
      IF (STRSSF(ILOADX  ).EQ.0.)  STRSSF(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =STRESS(ILOADX  )
      WORDCX= '(STRESS('//CIX//')/STRSSA('//CIX//
     1  ')) X STRSSF('//CIX//')'
      CALL CONX(STRESS(ILOADX  ),STRSSA(ILOADX  ),STRSSF(ILOADX  )
     1,'Maximum effective stress in wall of shell',
     1 'Maximum allowable stress',
     1 'Factor of safety for stress',
     1 1,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  Maximum effective stress in wall of shell: '
         IENDP4 =45
         CODNAM ='STRESS('//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
   70 CONTINUE
   71 CONTINUE
C
C  Behavior and constraints generated next for BSYM:
```

```
C  BSYM = Symmetric buckling load factor
C
      PHRASE =
     1 'Symmetric buckling load factor'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(2  ).EQ.0) CALL BEHX2
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'Symmetric buckling load factor')
      IF (BSYM(ILOADX  ).EQ.0.)  BSYM(ILOADX  ) = 1.E+10
      IF (BSYMA(ILOADX  ).EQ.0.)  BSYMA(ILOADX  ) = 1.0
      IF (BSYMF(ILOADX  ).EQ.0.)  BSYMF(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =BSYM(ILOADX  )
      WORDCX= '(BSYM('//CIX//')/BSYMA('//CIX//
     1  ')) / BSYMF('//CIX//')'
      CALL CONX(BSYM(ILOADX  ),BSYMA(ILOADX  ),BSYMF(ILOADX  )
     1,'Symmetric buckling load factor',
     1 'Allowable for sym. buckling load factor',
     1 'Factor of safety for sym. buckling load',
     1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  Symmetric buckling load factor: '
         IENDP4 =34
         CODNAM ='BSYM('//CIX//')'
         MLET4 =4 + 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
   85 CONTINUE
   86 CONTINUE
C
C  Behavior and constraints generated next for BANTI:
C  BANTI = Antisymmetric buckling load factor
C
      PHRASE =
     1 'Antisymmetric buckling load factor'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(3  ).EQ.0) CALL BEHX3
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'Antisymmetric buckling load factor')
      IF (BANTI(ILOADX  ).EQ.0.)  BANTI(ILOADX  ) = 1.E+10
      IF (BANTIA(ILOADX  ).EQ.0.)  BANTIA(ILOADX  ) = 1.0
      IF (BANTIF(ILOADX  ).EQ.0.)  BANTIF(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =BANTI(ILOADX  )
```

```
       WORDCX= '(BANTI('//CIX//')/BANTIA('//CIX//
      1  ')) / BANTIF('//CIX//')'
       CALL CONX(BANTI(ILOADX  ),BANTIA(ILOADX  ),BANTIF(ILOADX  )
      1,'Antisymmetric buckling load factor',
      1 'Allowable for antisym. buckling load factor',
      1 'Factor of safety for antisym. buckling load',
      1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
      1 WORDMX,PCWORD,CPLOTX,ICARX)
       IF (IMODX.EQ.0) THEN
          CODPHR =
      1 '  Antisymmetric buckling load factor: '
          IENDP4 =38
          CODNAM ='BANTI('//CIX//')'
          MLET4 =5 + 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
  100 CONTINUE
  101 CONTINUE
C
C  Behavior and constraints generated next for FREQ:
C  FREQ = Fundamental modal frequency (hertz)
C
       PHRASE =
      1 'Fundamental modal frequency (hertz)'
       CALL BLANKX(PHRASE,IENDP4)
       IF (IBEHV(4  ).EQ.0) CALL BEHX4
      1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
      1 'Fundamental modal frequency (hertz)')
       IF (FREQ(ILOADX  ).EQ.0.)  FREQ(ILOADX  ) = 1.E+10
       IF (FREQA(ILOADX  ).EQ.0.)  FREQA(ILOADX  ) = 1.0
       IF (FREQF(ILOADX  ).EQ.0.)  FREQF(ILOADX  ) = 1.0
       KCONX = KCONX + 1
       CARX(KCONX) =FREQ(ILOADX  )
       WORDCX= '(FREQ('//CIX//')/FREQA('//CIX//
      1  ')) / FREQF('//CIX//')'
       CALL CONX(FREQ(ILOADX  ),FREQA(ILOADX  ),FREQF(ILOADX  )
      1,'Fundamental modal frequency (hertz)',
      1 'Allowable for modal frequency',
      1 'Factor of safety for modal frequency',
      1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
      1 WORDMX,PCWORD,CPLOTX,ICARX)
       IF (IMODX.EQ.0) THEN
          CODPHR =
      1 '  Fundamental modal frequency (hertz): '
          IENDP4 =39
          CODNAM ='FREQ('//CIX//')'
```

```
         MLET4 =4 + 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
  115 CONTINUE
  116 CONTINUE
C
C  NEXT, EVALUATE THE OBJECTIVE, OBJGEN:
      IF (ILOADX.EQ.1) THEN
         PHRASE ='weight of half of cyl. shell'
         CALL BLANKX(PHRASE,IENDP4)
         CALL OBJECT(IFILE8,NPRINX,IMODX,OBJGEN,
     1    'weight of half of cyl. shell')
      ENDIF
      NCONSX = ICONSX
C
      CALL CLSGEN
C
      RETURN
      END
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  Start of part of TRANFR written by "GENTEXT"
C  INSERT ADDITIONAL COMMON BLOCKS HERE: (THESE ARE "GENTEXT" VARIABLES)
      COMMON/FV07/NX(20)
      REAL NX
```

```
      COMMON/FV11/STRESS(20),STRSSA(20),STRSSF(20)
      REAL STRESS,STRSSA,STRSSF
      COMMON/FV14/BSYM(20),BSYMA(20),BSYMF(20)
      REAL BSYM,BSYMA,BSYMF
      COMMON/FV17/BANTI(20),BANTIA(20),BANTIF(20)
      REAL BANTI,BANTIA,BANTIF
      COMMON/FV20/FREQ(20),FREQA(20),FREQF(20)
      REAL FREQ,FREQA,FREQF
      COMMON/IV01/IBOUND
      INTEGER IBOUND
      COMMON/FV01/LENGTH,RADIUS,THICK,ESTIFF,NU,DENS,WEIGHT
      REAL LENGTH,RADIUS,THICK,ESTIFF,NU,DENS,WEIGHT
      COMMON/FV08/PRESS(20)
      REAL PRESS
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
C
C
C
-----END OF THE GENOPT-user-completed struct.new file, struct.new ------
```

Note that in this particular case there is not much difference between struct.new after GENTEXT but before GENOPT-user-added statements and after completion by the user (struct.cylinder). Typing the command,

**diff /home/progs/genoptcase/struct.new**
**/home/progs/genopt/case/cylinder/struct.cylinder > struct.diff**

**produces the following file:**

```
------------- BEGINNING OF THE struct.diff FILE ---------------
130a131,132
>       COMMON/TOTMAX/TOTMAS
> C
192a195,216
>       CALL OPNGEN
>       CALL RWDGEN
> C
> C     initialize behaviors:
>       STRESS(ILOADX) = 0.
>       BSYM(ILOADX) = 0.
>       BANTI(ILOADX) = 0.
>       FREQ(ILOADX)   = 0.
> C
> C  Find mass of cyl. shell from boundary to mid-length symmetry plane.
> C  The mass is stored in TOTMAS, which is one of the BOSOR4 labelled
> C  common blocks.
>       INDIC = 0
> C BEG FEB 2008
> C     CALL BOSDEC(5,ILOADX,INDIC)
>       CALL BOSDEC(5,24,ILOADX,INDIC)
> C END FEB 2008
>       CALL B4READ
>       CALL GASP (DUM1,DUM2,-2,DUM3)
>       GRAVTY = 386.4
>       WEIGHT = GRAVTY*TOTMAS
> C
346a371
>       CALL CLSGEN
------------------ END OF THE struct.diff FILE ----------------------
```

The following comments apply to any GENOPT application in which BIGBOSOR4 is being used to perform most of the computations:

1. The statements,

```
      CALL OPNGEN
      CALL RWDGEN
```

must appear at the beginning of SUBROUTINE STRUCT.

2. The statement,

```
      CALL CLSGEN
```

must appear at the end of SUBROUTINE STRUCT.

**These “OPN”, “RWD”, and “CLS” statements call subroutines that OPEN, REWIND, and CLOSE files used by BIGBOSOR4 (or BOSOR4).**

In this particular case (cylinder) there are few user-supplied statements added by the GENOPT user to the version of struct.new generatied automatically by GENOPT during the execution of GENTEXT, as seen from the short struct.diff file reproduced above.

The only purpose of the statements,

```
      CALL BOSDEC(5,24,ILOADX,INDIC)
      CALL B4READ
      CALL GASP (DUM1,DUM2,-2,DUM3)
      GRAVTY = 386.4
```

is to compute TOTMAS, the total mass of the structure, from which the weight, WEIGHT, is determined in the next line:

```
      WEIGHT = GRAVTY*TOTMAS
```

The total mass of the structure, TOTMAS, is computed in the BIGBOSOR4 subroutine B4READ.

The line:

```
      CALL GASP (DUM1,DUM2,-2,DUM3)
```

is needed after completion of every BIGBOSOR4 analysis in order to clear the random access storage so that this storage will not continue needlessly to increase in size with each subsequent BIGBOSOR4 execution.

Please see the file, howto.behavior (Table a31), for more on GENOPT-user-supplied FORTRAN coding to the GENOPT-generated "skeletal" version of behavior.new for the case, "cylinder".

NOTE: Usually the GENOPT-user-written FORTRAN code to "flesh out" the GENTEXT-generated struct.new file is much more involved than that shown in the "cylinder" example. See, for example, the files,
/home/progs/genopt/case/torisph/struct.equivellipse (Table a16)
/home/progs/genopt/case/torisph/struct.ellipse
/home/progs/genopt/case/torisph/struct.tori