Table A31 list of the file,
/home/progs/genopt/case/cylinder/howto.behavior.
The purpose of this file is to provide the GENOPT user with some additional documentation about "**fleshing out**" the **skeletal behavior.new** file created automatically by GENTEXT in cases in which BIGBOSOR4 software is being used to compute behaviors, such as stress, buckling, displacement, etc. NOTE: In the generic case that is the subject of this report, that is, the GENOPT-user-selected case, "equivellipse", there is no modification of the behavior.new library by the GENOPT user. This table is included, however, because the reader may want to set up a different generic case in which he/she decides to "flesh out" the skeletal behavior.new library in addition to modifying the skeletal struct.new library. **The example provided here is the generic case called "cylinder", which is stored in the directory, /home/progs/genopt/case/cylinder**

==

/home/progs/genopt/case/cylinder/howto.behavior

10 March, 2008, modified 26 September, 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.5.2 and Table a.6 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.5.2 is on pp 47-49 of that report.

A "skeletal" version of behavior.new is created by GENOPT.
For the case "**cylinder**" that "skeletal" version is as follows:

----BEGINNING OF SKELETAL VERSION OF behavior.new CREATED BY GENOPT-----

```
C=DECK      BEHAVIOR.NEW
C  This library contains the skeletons of
C  subroutines called SUBROUTINE BEHXn, n = 1,
C  2, 3, . . . that will yield predictions
C  of behavioral responses of various systems
C  to environments (loads).
C
C  You may complete the subroutines by writing
```

```
C  algorithms that yield the responses,
C  each of which plays a part in constraining
C  the design to a feasible region. Examples
C  of responses are: stress, buckling, drag,
C  vibration, deformation, clearances, etc.
C
C  A skeleton routine called SUBROUTINE OBJECT
C  is also provided for any objective function
C  (e.g. weight, deformation, conductivity)
C  you may wish to create.
C
C  A skeleton routine called SUBROUTINE USRCON
C  is also provided for any user-written
C  constraint condition you may wish to write:
C  This is an INEQUALITY condition that
C  involves any program variables.  However,
C  note that this kind of thing is done
C  automatically in the program DECIDE, so
C  try DECIDE first to see if your particular
C  constraint conditions can be accommodated
C  more easily there.
C
C  Please note that you do not have to modify
C  BEHAVIOR.NEW in any way, but may instead
C  prefer to insert your subroutines into the
C  skeletal libraries ADDCODEn.NEW, n=1,2,...
C  and appropriate common blocks, dimension
C  and type statements and calls to these
C  subroutines in the library STRUCT.NEW.
C  This strategy is best if your FORTRAN
C  input to GENOPT contains quite a bit
C  of software previously written by
C  yourself or others, and/or the generation
C  of behavioral constraints is more easily
C  accomplished via another architecture
C  than that provided for in the
C  BEHAVIOR.NEW library. (See instructions
C  in the libraries ADDCODEn.NEW and
C  STRUCT.NEW for this procedure.)
C
C  (many lines skipped in order to save space. See Table a13.)
C
C==============================================================
C   TABLE 2   GLOSSARY OF VARIABLES USED IN
C/home/progs/genopt/case/cylinder
C==============================================================
C  ARRAY  NUMBER OF       PROMPT
C    ?   (ROWS,COLS) ROLE NUMBER  NAME      DEFINITION OF VARIABLE
```

```
C                      (cylinder.PRO)
C======================================================================
C    n   ( 0,   0)    2     10   LENGTH= Length of the cylindrical shell
C    n   ( 0,   0)    2     15   RADIUS= Radius of the cylindrical shell
C    n   ( 0,   0)    1     20   THICK = Thickness of the cylindrical shel
C    n   ( 0,   0)    2     25   ESTIFF= Youngs modulus of the shell wall
C    n   ( 0,   0)    2     30   NU    = Poisson ratio of the shell wall
C    n   ( 0,   0)    2     35   DENS  = mass density (e.g. lb-sec^2/in^4)
C    n   ( 0,   0)    2     40   IBOUND= IBOUND = 1 = simple support; 2 =
C    n   ( 0,   0)    2     45   NCASES= Number of load cases (number of e
C    y   (20,   0)    3     50   NX    = Axial resultant (e.g. lb/in)
C    y   (20,   0)    3     55   PRESS = Pressure, positive for internal
C    y   (20,   0)    4     60   STRESS= Maximum effective stress in wall
C    y   (20,   0)    5     65   STRSSA= Maximum allowable stress
C    y   (20,   0)    6     70   STRSSF= Factor of safety for stress
C    y   (20,   0)    4     75   BSYM  = Symmetric buckling load factor
C    y   (20,   0)    5     80   BSYMA = Allowable for sym. buckling load
C    y   (20,   0)    6     85   BSYMF = Factor of safety for sym. bucklin
C    y   (20,   0)    4     90   BANTI = Antisymmetric buckling load facto
C    y   (20,   0)    5     95   BANTIA= Allowable for antisym. buckling l
C    y   (20,   0)    6    100   BANTIF= Factor of safety for antisym. buc
C    y   (20,   0)    4    105   FREQ  = Fundamental modal frequency (hert
C    y   (20,   0)    5    110   FREQA = Allowable for modal frequency
C    y   (20,   0)    6    115   FREQF = Factor of safety for modal freque
C    n   ( 0,   0)    7    120   WEIGHT= weight of half of cyl. shell
C
C
C
C=DECK      BEHX1
      SUBROUTINE BEHX1
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN Maximum effective stress in wall of shell
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     STRESS(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
```

```
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = Maximum effective stress in wall of shell
C
C   OUTPUT:
C
C     STRESS(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      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  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
C
C
C
C
```

```
C=DECK      BEHX2
      SUBROUTINE BEHX2
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN Symmetric buckling load factor
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     BSYM(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = Symmetric buckling load factor
C
C   OUTPUT:
C
C     BSYM(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      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  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX3
      SUBROUTINE BEHX3
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN Antisymmetric buckling load factor
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     BANTI(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
```

```
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = Antisymmetric buckling load factor
C
C   OUTPUT:
C
C     BANTI(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      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  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX4
      SUBROUTINE BEHX4
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN Fundamental modal frequency (hertz)
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
```

```
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     FREQ(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = Fundamental modal frequency (hertz)
C
C   OUTPUT:
C
C     FREQ(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      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  INSERT SUBROUTINE STATEMENTS HERE.
C
C
C
C
      RETURN
      END
C
C
C
C
C=DECK      USRCON
      SUBROUTINE USRCON(INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,
     1  WORDCX,WORDMX,PCWORD,CPLOTX,ICARX,IFILEX)
C   PURPOSE: GENERATE USER-WRITTEN
C   INEQUALITY CONSTRAINT CONDITION
C   USING ANY COMBINATION OF PROGRAM
C   VARIABLES.
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS A CONSTRAINT CONDITION,
C   CALLED "CONX" IN THIS ROUTINE.
      DIMENSION WORDCX(*),WORDMX(*),IPOINC(*),CONSTX(*)
      DIMENSION PCWORD(*),CPLOTX(*)
      CHARACTER*80 WORDCX,WORDMX,PCWORD
C  INSERT ADDITIONAL COMMON BLOCKS:
      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
      CONX = 0.0
C
C  INSERT USER-WRITTEN STATEMENTS
```

```
C  HERE. THE CONSTRAINT CONDITION
C  THAT YOU CALCULATE IS CALLED "CONX"
C
      IF (CONX.EQ.0.0) RETURN
      IF (CONX.LT.0.0) THEN
         WRITE(IFILEX,*)' CONX MUST BE GREATER THAN ZERO.'
         CALL EXIT
      ENDIF
C
C  DO NOT CHANGE THE FOLLOWING STATEMENTS, EXCEPT WORDC
C
      ICARX = ICARX + 1
      INUMTT = INUMTT + 1
      WORDCX(ICARX) = ' USER: PROVIDE THIS.'
      CPLOTX(ICARX) = CONX - 1.
      CALL BLANKX(WORDCX(ICARX),IENDP)
      PCWORD(ICARX)  = WORDCX(ICARX)(1:IENDP)//' -1'
      IF (IMODX.EQ.0.AND.CONX.GT.CONMAX) GO TO 200
      IF (IMODX.EQ.1.AND.IPOINC(INUMTT).EQ.0) GO TO 200
      ICONSX = ICONSX + 1
      IF (IMODX.EQ.0) IPOINC(INUMTT) = 1
      CONSTX(ICONSX) = CONX
      WORDMX(ICONSX)  = WORDCX(ICARX)(1:IENDP)//' -1'
  200 CONTINUE
C  END OF USRCON
C
C
      RETURN
      END
C
C
C
C=DECK      USRLNK
      SUBROUTINE USRLNK(VARI,I,VARIAB)
C Purpose: generate user-written
C linking conditions using any
C combination of decision variables.
C You must write conde that, using
C the variables in the subroutine
C argument VARIAB as input, ultimately
C yield a value for the linked variable
C VARI.
C
C VARI is the Ith entry of the array
C VARIAB.  You have decided that this
C is to be a linked variable with user
C defined linking.  It is linked to
C the decision variables in the array
```

```
C VARIAB.
C An example will provide the simplest
C explanation of this:
C Let"s say that the 5th decision
C variable candidate (I=5) is linked
C to the decision variable candidates
C 2 and 7. (You used DECIDE to select
C these as decision variables.
C In this case VARI is equal to
C VARIAB(I).  You then write your
C linking equation in the form
C VARI=f(VARIAB(2),VARIAB(7)).
C Use the index I in an IF statement if
C you have more than one user-defined
C linked variable.
C
C
      REAL VARI,VARIAB(50)
      INTEGER I
C
C  INSERT USER-WRITTEN DECLARATION
C  STATEMENTS HERE.
C
C  INSERT USER-WRITTEN
C  STATEMENTS HERE.
C
C
C  END OF USRLNK
      RETURN
      END
C=DECK      OBJECT
      SUBROUTINE OBJECT(IFILE,NPRINX,IMODX,OBJGEN,PHRASE)
C   PURPOSE:weight of half of cyl. shell
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE OBJECTIVE FUNCTION
C         WEIGHT
C   AS OUTPUT. MAKE SURE TO INCLUDE AT
C   THE END OF THE SUBROUTINE, THE
C   STATEMENT: OBJGEN = WEIGHT
C
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
```

```
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C   DEFINITION OF PHRASE:
C     PHRASE = weight of half of cyl. shell
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      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  INSERT SUBROUTINE STATEMENTS HERE.
C
C
      OBJGEN =WEIGHT
C
C
      RETURN
      END
C
C
C
```

------END OF SKELETAL VERSION OF behavior.new CREATED BY GENOPT-----

Wherever the line:

```
C  INSERT SUBROUTINE STATEMENTS HERE.
```

occurs, the user can write whatever FORTRAN code he/she wants that accomplishes the job. In this case we are using BIGBOSOR4 software to perform most of the computations. Therefore, the user-added FORTRAN code must include calls to BIGBOSOR4 subroutines.

The following list gives the expanded version of behavior.new as "fleshed out" by the GENOPT user for the GENOPT sample case, **/home/progs/genopt/case/cylinder.**
The GENOPT-user-added FORTRAN statements are listed in boldface:

----BEGINNING OF "FLESHED OUT" (GENOPT-USER-COMPLETED) VERSION OF behavior.new for the case, /home/progs/genopt/case/cylinder-----

```
C=DECK      BEHAVIOR.NEW
C  This library contains the skeletons of
C
C  (many lines skipped to save space)
C
C
C=DECK      BEHX1
      SUBROUTINE BEHX1
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN Maximum effective stress in wall of shell
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     STRESS(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
```

```
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = Maximum effective stress in wall of shell
C
C   OUTPUT:
C
C     STRESS(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      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  INSERT SUBROUTINE STATEMENTS HERE.
C
      COMMON/INSTAB/INDIC
      COMMON/ENDUVX/ENDUV,STRMAX,ARCLEN
C BEG FEB 2008
      COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
      common/caseblock/CASE
      CHARACTER*28 CASE
      CHARACTER*35 CASA
C END FEB 2008
C
      INDIC = 0
      CALL BOSDEC(1,24,ILOADX,INDIC)
C BEG FEB 2008
      IF (ITYPEX.EQ.2) THEN
C      Get CASE.BEHX1 file for input for BIGBOSOR4...
C      CASE.BEHX1 is an input file for BIGBOSOR4 for behavior no. 1:
```

```
C      Maximum effective stress in wall of shell
         I=INDEX(CASE,' ')
         IF(I.NE.0) THEN
            CASA=CASE(:I-1)//'.BEHX1'
         ELSE
            CASA=CASE//'.BEHX1'
         ENDIF
         OPEN(UNIT=61,FILE=CASA,STATUS='UNKNOWN')
         CALL BOSDEC(1,61,ILOADX,INDIC)
         CLOSE(UNIT=61)
         WRITE(IFILE,'(/,/,A,A,/,A)')
     1 ' BIGBOSOR4 input file for:',
     1 ' Maximum effective stress in wall of shell',
     1   CASA
      ENDIF
C END FEB 2008
      CALL B4READ
      CALL B4MAIN
      CALL B4POST
      CALL GASP(DUM1, DUM2, -2, DUM3)
      STRESS(ILOADX) = STRMAX
      WRITE(IFILE,'(A,1P,E12.4)')
     1' Maximum effective stress from BEHX1: STRESS=',STRMAX
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX2
      SUBROUTINE BEHX2
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN Symmetric buckling load factor
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     BSYM(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
```

```
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = Symmetric buckling load factor
C
C   OUTPUT:
C
C     BSYM(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      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  INSERT SUBROUTINE STATEMENTS HERE.
C
      COMMON/INSTAB/INDIC
      COMMON/EIGB4M/EIGCOM(200),EIGNEG(200),EIGCRN
      COMMON/WVEB4M/NWVCOM(200),NWVNEG(200),IWAVEB,NWVCRN
      COMMON/EIGBUK/EIGCRT
      COMMON/NWVBUK/NWVCRT
      COMMON/BUCKN/N0BX,NMINBX,NMAXBX,INCRBX
      COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
      COMMON/EIGALL/EIG2,EIG3,EIG4
      COMMON/WAVALL/NWAV2,NWAV3,NWAV4
C BEG FEB 2008
      COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
      common/caseblock/CASE
      CHARACTER*28 CASE
      CHARACTER*35 CASA
C END FEB 2008
C
      INDIC = 1
      NOB = 2
      NMAXB = 10
      CALL BOSDEC(2,24,ILOADX,INDIC)
C BEG FEB 2008
      IF (ITYPEX.EQ.2) THEN
C      Get CASE.BEHX2 file for input for BIGBOSOR4...
C      CASE.BEHX2 is an input file for BIGBOSOR4 for behavior no. 2:
C      Symmetric buckling load factor
         I=INDEX(CASE,' ')
         IF(I.NE.0) THEN
            CASA=CASE(:I-1)//'.BEHX2'
         ELSE
            CASA=CASE//'.BEHX2'
         ENDIF
         OPEN(UNIT=61,FILE=CASA,STATUS='UNKNOWN')
         CALL BOSDEC(2,61,ILOADX,INDIC)
         CLOSE(UNIT=61)
         WRITE(IFILE,'(/,/,A,A,/,A)')
     1 ' BIGBOSOR4 input file for:',
     1 ' Symmetric buckling load factor',
     1   CASA
      ENDIF
C END FEB 2008
      CALL B4READ
      IF (IMODX.EQ.0) THEN
         NOBX = NOB
         NMINBX = NOB
         NMAXBX = NMAXB
         INCRBX = 1
      ELSE
         NOBX = NWAV2
         NMINBX = NWAV2
         NMAXBX = NWAV2
         INCRBX = 1
      ENDIF
      REWIND IFILE9
      CALL STOCM1(IFILE9)
      CALL STOCM2(IFILE9)
      CALL B4MAIN
      CALL GASP(DUM1,DUM2,-2,DUM3)
      IF (IMODX.EQ.0) THEN
         EIG2 = EIGCRT
         NWAV2= NWVCRT
      ENDIF
      WRITE(IFILE,'(/,A)')
     1 ' SYMMETRIC BUCKLING LOAD FACTORS AND MODES (BEHX2)'
      DO 10 I = 1,IWAVEB
         WRITE(IFILE,'(A,1P,E12.4,A,I4,A)')
     1  '      ',EIGCOM(I),'(',NWVCOM(I),')'
   10 CONTINUE
      WRITE(IFILE,'(A,1P,E12.4)')
     1' Critical buckling load factor, BSYM=',EIGCRT
      WRITE(IFILE,'(A,I5)')
     1' Critical number of circumferential waves, NWVCRT=',NWVCRT
      BSYM(ILOADX) = EIGCRT
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX3
      SUBROUTINE BEHX3
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN Antisymmetric buckling load factor
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     BANTI(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
```

```
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = Antisymmetric buckling load factor
C
C   OUTPUT:
C
C     BANTI(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      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  INSERT SUBROUTINE STATEMENTS HERE.
C
      COMMON/INSTAB/INDIC
      COMMON/EIGB4M/EIGCOM(200),EIGNEG(200),EIGCRN
      COMMON/WVEB4M/NWVCOM(200),NWVNEG(200),IWAVEB,NWVCRN
      COMMON/EIGBUK/EIGCRT
      COMMON/NWVBUK/NWVCRT
      COMMON/BUCKN/N0BX,NMINBX,NMAXBX,INCRBX
      COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
      COMMON/EIGALL/EIG2,EIG3,EIG4
      COMMON/WAVALL/NWAV2,NWAV3,NWAV4
C BEG FEB 2008
      COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
      common/caseblock/CASE
      CHARACTER*28 CASE
      CHARACTER*35 CASA
C END FEB 2008
C
      INDIC = 1
      NOB = 2
      NMAXB = 15
      CALL BOSDEC(3,24,ILOADX,INDIC)
C BEG FEB 2008
      IF (ITYPEX.EQ.2) THEN
C      Get CASE.BEHX3 file for input for BIGBOSOR4...
C      CASE.BEHX3 is an input file for BIGBOSOR4 for behavior no. 3:
C      Antisymmetric buckling load factor
         I=INDEX(CASE,' ')
         IF(I.NE.0) THEN
            CASA=CASE(:I-1)//'.BEHX3'
         ELSE
            CASA=CASE//'.BEHX3'
         ENDIF
         OPEN(UNIT=61,FILE=CASA,STATUS='UNKNOWN')
         CALL BOSDEC(3,61,ILOADX,INDIC)
         CLOSE(UNIT=61)
         WRITE(IFILE,'(/,/,A,A,/,A)')
     1 ' BIGBOSOR4 input file for:',
     1 ' Antisymmetric buckling load factor',
     1   CASA
      ENDIF
C END FEB 2008
      CALL B4READ
      IF (IMODX.EQ.0) THEN
         NOBX = NOB
         NMINBX = NOB
         NMAXBX = NMAXB
         INCRBX = 1
      ELSE
         NOBX = NWAV3
         NMINBX = NWAV3
         NMAXBX = NWAV3
         INCRBX = 1
      ENDIF
      REWIND IFILE9
      CALL STOCM1(IFILE9)
      CALL STOCM2(IFILE9)
      CALL B4MAIN
      CALL GASP(DUM1,DUM2,-2,DUM3)
      IF (IMODX.EQ.0) THEN
         EIG3 = EIGCRT
         NWAV3= NWVCRT
      ENDIF
      WRITE(IFILE,'(/,A)')
     1 ' ANTISYMMETRIC BUCKLING LOAD FACTORS AND MODES (BEHX3)'
      DO 10 I = 1,IWAVEB
         WRITE(IFILE,'(A,1P,E12.4,A,I4,A)')
     1  '       ',EIGCOM(I),'(',NWVCOM(I),')'
   10 CONTINUE
      WRITE(IFILE,'(A,1P,E12.4)')
     1' Critical buckling load factor, BANTI=',EIGCRT
      WRITE(IFILE,'(A,I5)')
     1' Critical number of circumferential waves, NWVCRT=',NWVCRT
      BANTI(ILOADX) = EIGCRT
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX4
      SUBROUTINE BEHX4
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN Fundamental modal frequency (hertz)
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE RESPONSE VARIABLE FOR
C   THE ith LOAD CASE, ILOADX:
C
C     FREQ(ILOADX)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
```

```
C     ILOADX = ith LOADING COMBINATION
C     PHRASE = Fundamental modal frequency (hertz)
C
C   OUTPUT:
C
C     FREQ(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      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  INSERT SUBROUTINE STATEMENTS HERE.
C
      COMMON/INSTAB/INDIC
      COMMON/EIGB4M/EIGCOM(200),EIGNEG(200),EIGCRN
      COMMON/WVEB4M/NWVCOM(200),NWVNEG(200),IWAVEB,NWVCRN
      COMMON/EIGBUK/EIGCRT
      COMMON/NWVBUK/NWVCRT
      COMMON/BUCKN/N0BX,NMINBX,NMAXBX,INCRBX
      COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
      COMMON/EIGALL/EIG2,EIG3,EIG4
      COMMON/WAVALL/NWAV2,NWAV3,NWAV4
C BEG FEB 2008
      COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
      common/caseblock/CASE
      CHARACTER*28 CASE
      CHARACTER*35 CASA
C END FEB 2008
C
      INDIC = 2
      N0B = 2
      NMAXB = 10
      CALL BOSDEC(4,24,ILOADX,INDIC)
C BEG FEB 2008
      IF (ITYPEX.EQ.2) THEN
C      Get CASE.BEHX4 file for input for BIGBOSOR4...
C      CASE.BEHX4 is an input file for BIGBOSOR4 for behavior no. 4:
C      Fundamental modal frequency (hertz).
         I=INDEX(CASE,' ')
         IF(I.NE.0) THEN
            CASA=CASE(:I-1)//'.BEHX4'
         ELSE
            CASA=CASE//'.BEHX4'
         ENDIF
         OPEN(UNIT=61,FILE=CASA,STATUS='UNKNOWN')
         CALL BOSDEC(4,61,ILOADX,INDIC)
         CLOSE(UNIT=61)
         WRITE(IFILE,'(/,/,A,A,/,A)')
     1 ' BIGBOSOR4 input file for:',
     1 ' Fundamental modal frequency (hertz)',
     1   CASA
      ENDIF
C END FEB 2008
      CALL B4READ
      IF (IMODX.EQ.0) THEN
         NOBX = NOB
         NMINBX = NOB
         NMAXBX = NMAXB
         INCRBX = 1
      ELSE
         NOBX = NWAV4
         NMINBX = NWAV4
         NMAXBX = NWAV4
         INCRBX = 1
      ENDIF
      REWIND IFILE9
      CALL STOCM1(IFILE9)
      CALL STOCM2(IFILE9)
      CALL B4MAIN
      CALL GASP(DUM1,DUM2,-2,DUM3)
      IF (IMODX.EQ.0) THEN
         EIG4 = EIGCRT
         NWAV4= NWVCRT
      ENDIF
      WRITE(IFILE,'(/,A)')
     1 ' NATURAL FREQUENCIES AND MODES (BEHX4)'
      DO 10 I = 1,IWAVEB
         WRITE(IFILE,'(A,1P,E12.4,A,I4,A)')
     1  '      ',EIGCOM(I),'(',NWVCOM(I),')'
   10 CONTINUE
```

```
      WRITE(IFILE,'(A,1P,E12.4)')
     1' Critical buckling load factor, FREQ=',EIGCRT
      WRITE(IFILE,'(A,I5)')
     1' Critical number of circumferential waves, NWVCRT=',NWVCRT
      FREQ(ILOADX) = EIGCRT
C
      RETURN
      END
C
C
C
C
C=DECK      USRCON
      SUBROUTINE USRCON(INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,
     1  WORDCX,WORDMX,PCWORD,CPLOTX,ICARX,IFILEX)
C   PURPOSE: GENERATE USER-WRITTEN
C   INEQUALITY CONSTRAINT CONDITION
C   USING ANY COMBINATION OF PROGRAM
C   VARIABLES.
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS A CONSTRAINT CONDITION,
C   CALLED "CONX" IN THIS ROUTINE.
      DIMENSION WORDCX(*),WORDMX(*),IPOINC(*),CONSTX(*)
      DIMENSION PCWORD(*),CPLOTX(*)
      CHARACTER*80 WORDCX,WORDMX,PCWORD
C  INSERT ADDITIONAL COMMON BLOCKS:
      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
      CONX = 0.0
C
C  INSERT USER-WRITTEN STATEMENTS
```

```
C  HERE. THE CONSTRAINT CONDITION
C  THAT YOU CALCULATE IS CALLED "CONX"
C
      IF (CONX.EQ.0.0) RETURN
      IF (CONX.LT.0.0) THEN
         WRITE(IFILEX,*)' CONX MUST BE GREATER THAN ZERO.'
         CALL EXIT
      ENDIF
C
C  DO NOT CHANGE THE FOLLOWING STATEMENTS, EXCEPT WORDC
C
      ICARX = ICARX + 1
      INUMTT = INUMTT + 1
      WORDCX(ICARX) = ' USER: PROVIDE THIS.'
      CPLOTX(ICARX) = CONX - 1.
      CALL BLANKX(WORDCX(ICARX),IENDP)
      PCWORD(ICARX)  = WORDCX(ICARX)(1:IENDP)//' -1'
      IF (IMODX.EQ.0.AND.CONX.GT.CONMAX) GO TO 200
      IF (IMODX.EQ.1.AND.IPOINC(INUMTT).EQ.0) GO TO 200
      ICONSX = ICONSX + 1
      IF (IMODX.EQ.0) IPOINC(INUMTT) = 1
      CONSTX(ICONSX) = CONX
      WORDMX(ICONSX)  = WORDCX(ICARX)(1:IENDP)//' -1'
  200 CONTINUE
C  END OF USRCON
C
C
      RETURN
      END
C
C
C
C=DECK      USRLNK
      SUBROUTINE USRLNK(VARI,I,VARIAB)
C Purpose: generate user-written
C linking conditions using any
C combination of decision variables.
C You must write conde that, using
C the variables in the subroutine
C argument VARIAB as input, ultimately
C yield a value for the linked variable
C VARI.
C
C VARI is the Ith entry of the array
C VARIAB.  You have decided that this
C is to be a linked variable with user
C defined linking.  It is linked to
C the decision variables in the array
```

```
C VARIAB.
C An example will provide the simplest
C explanation of this:
C Let"s say that the 5th decision
C variable candidate (I=5) is linked
C to the decision variable candidates
C 2 and 7. (You used DECIDE to select
C these as decision variables.
C In this case VARI is equal to
C VARIAB(I).  You then write your
C linking equation in the form
C VARI=f(VARIAB(2),VARIAB(7)).
C Use the index I in an IF statement if
C you have more than one user-defined
C linked variable.
C
C
      REAL VARI,VARIAB(50)
      INTEGER I
C
C  INSERT USER-WRITTEN DECLARATION
C  STATEMENTS HERE.
C
C  INSERT USER-WRITTEN
C  STATEMENTS HERE.
C
C
C  END OF USRLNK
      RETURN
      END
C=DECK      OBJECT
      SUBROUTINE OBJECT(IFILE,NPRINX,IMODX,OBJGEN,PHRASE)
C   PURPOSE:weight of half of cyl. shell
C
C   YOU MUST WRITE CODE THAT, USING
C   THE VARIABLES IN THE LABELLED
C   COMMON BLOCKS AS INPUT, ULTIMATELY
C   YIELDS THE OBJECTIVE FUNCTION
C         WEIGHT
C   AS OUTPUT. MAKE SURE TO INCLUDE AT
C   THE END OF THE SUBROUTINE, THE
C   STATEMENT: OBJGEN = WEIGHT
C
C
C   DEFINITIONS OF INPUT DATA:
C    IMODX  = DESIGN CONTROL INTEGER:
C     IMODX = 0 MEANS BASELINE DESIGN
C     IMODX = 1 MEANS PERTURBED DESIGN
```

```
C     IFAST = 0 MEANS FEW  SHORTCUTS FOR PERTURBED DESIGNS
C     IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS
C    IFILE = FILE FOR OUTPUT LIST:
C    NPRINX= OUTPUT CONTROL INTEGER:
C     NPRINX=0 MEANS SMALLEST AMOUNT
C     NPRINX=1 MEANS MEDIUM AMOUNT
C     NPRINX=2 MEANS LOTS OF OUTPUT
C
C   DEFINITION OF PHRASE:
C     PHRASE = weight of half of cyl. shell
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      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  INSERT SUBROUTINE STATEMENTS HERE.
C
C
      OBJGEN =WEIGHT
C
C
      RETURN
      END
C
C
C
```

----END OF "FLESHED OUT" (USER-COMPLETED) VERSION OF behavior.new -----

This "fleshed out" version of behavior.new should be saved by the GENOPT user via a statement such as the following:

```
cp /home/progs/genoptcase/behavior.new
/home/progs/genopt/case/cylinder/behavior.cylinder
```

Please see Table a.6 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.

for explanations of the user-added FORTRAN code.

Note that the part of the user-added FORTRAN code devoted to computations in SUBROUTINE BEHX1 (for example) has the form:

```
--------------------------------------------------------------
      INDIC = 0
      CALL BOSDEC(1,24,ILOADX,INDIC)
C BEG FEB 2008
      IF (ITYPEX.EQ.2) THEN
C      Get CASE.BEHX1 file for input for BIGBOSOR4...
C      CASE.BEHX1 is an input file for BIGBOSOR4 for behavior no. 1:
C      Maximum effective stress in wall of shell
         I=INDEX(CASE,' ')
         IF(I.NE.0) THEN
            CASA=CASE(:I-1)//'.BEHX1'
         ELSE
            CASA=CASE//'.BEHX1'
         ENDIF
         OPEN(UNIT=61,FILE=CASA,STATUS='UNKNOWN')
         CALL BOSDEC(1,61,ILOADX,INDIC)
         CLOSE(UNIT=61)
         WRITE(IFILE,'(/,/,A,A,/,A)')
     1 ' BIGBOSOR4 input file for:',
     1 ' Maximum effective stress in wall of shell',
     1   CASA
      ENDIF
C END FEB 2008
      CALL B4READ
      CALL B4MAIN
      CALL B4POST
      CALL GASP(DUM1, DUM2, -2, DUM3)
      STRESS(ILOADX) = STRMAX
      WRITE(IFILE,'(A,1P,E12.4)')
     1' Maximum effective stress from BEHX1: STRESS=',STRMAX
----------------------------------------------------------------
```

This is typical when BIGBOSOR4 is being used to perform most of the work. In the example just listed:

1. The BIGBOSOR4 analysis type indicator, INDIC, is set to a value; in this case INDIC = 0 because we want to perform a nonlinear axisymmetric stress analysis in SUBROUTINE BEHX1 .

2. SUBROUTINE BOSDEC [see howto.bosdec (Table a29)] is executed in order to generate a valid input file for BIGBOSOR4 in this instance with INDIC = 0

3. The statements bracketed by C BEG FEB 2008 and C END FEB 2008 create a file called <casename>.BEHX1 (in which <casename> = cyl, the specific (rather than generic) user-assigned name of the case. The cyl.BEHX1 file can later be used as input to BIGBOSOR4 (or BOSOR4) in a BOSOR run independent of GENOPT. The user may want to do this in order to obtain plots of the shell configuration and/or shell response to the applied loading for this particular behavior, BEHX1. Analogous files, cyl.BEHXi, i = 2, 3, 4, are created corresponding to each of the “behavior” subroutines, BEHXi.

4. The three statements,

```
      CALL B4READ
      CALL B4MAIN
      CALL B4POST
```

represent execution of BIGBOSOR4. These three statements perform the same computations as the command, BIGBOSORALL (for BIGBOSOR4) or BOSORALL (for BOSOR4).

5. The statement,

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

re-initializes random access storage (the file called cyl.RAN). It is necessary to include this statement after completion of any BIGBOSOR4 computation in order to prevent the file cyl.RAN from becoming huge.

6. The statement,

```
      STRESS(ILOADX) = STRMAX
```

inserts the BIGBOSOR4 variable, STRMAX (maximum stress) into the user-named GENOPT variable, STRESS(ILOADX), in which ILOADX is the load set number.

The margin relating to behavior BEHX1 is computed in SUBROUTINE STRUCT [see howto.struct (Table a30)] with use of the quantity, STRESS(ILOADX). The relevant GENTOPT-created statements in SUBROUTINE STRUCT are as follows (see howto.struct):

```
--------------------------------------------------------------
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
---------------------------------------------------------------------
```

NOTE: It is important for the reader to grasp that the just-listed FORTRAN code fragment from SUBROUTINE STRUCT is created entirely by GENOPT during the GENTEXT execution that leads to the "skeletal" version of struct.new.

The command,

diff /home/progs/genoptcase/behavior.new
/home/progs/genopt/case/cylinder/behavior.cylinder > behavior.diff

yields the behavior.diff file, which is the difference between the skeletal behavior.new file created automatically by GENTEXT and the "fleshed-out" behavior.cylinder file augmented by the GENOPT user:

--------BEGINNING OF behavior.diff FILE ------------------

```
428,429c428,465
< C
< C        (User-added changes to SUBROUTINE BEHX1)
---
>       COMMON/INSTAB/INDIC
>       COMMON/ENDUVX/ENDUV,STRMAX,ARCLEN
> C BEG FEB 2008
>       COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
>       common/caseblock/CASE
>       CHARACTER*28 CASE
>       CHARACTER*35 CASA
> C END FEB 2008
> C
>       INDIC = 0
>       CALL BOSDEC(1,24,ILOADX,INDIC)
> C BEG FEB 2008
>       IF (ITYPEX.EQ.2) THEN
> C      Get CASE.BEHX1 file for input for BIGBOSOR4...
> C      CASE.BEHX1 is an input file for BIGBOSOR4 for behavior no. 1:
> C      Maximum effective stress in wall of shell
>         I=INDEX(CASE,' ')
>         IF(I.NE.0) THEN
>            CASA=CASE(:I-1)//'.BEHX1'
>         ELSE
>            CASA=CASE//'.BEHX1'
>         ENDIF
>         OPEN(UNIT=61,FILE=CASA,STATUS='UNKNOWN')
>         CALL BOSDEC(1,61,ILOADX,INDIC)
>         CLOSE(UNIT=61)
>         WRITE(IFILE,'(/,/,A,A,/,A)')
>      1 ' BIGBOSOR4 input file for:',
>      1 ' Maximum effective stress in wall of shell',
>      1   CASA
>       ENDIF
> C END FEB 2008
>       CALL B4READ
>       CALL B4MAIN
>       CALL B4POST
>       CALL GASP(DUM1, DUM2, -2, DUM3)
>       STRESS(ILOADX) = STRMAX
>       WRITE(IFILE,'(A,1P,E12.4)')
>      1' Maximum effective stress from BEHX1: STRESS=',STRMAX
496,497c532,603
```

```
< C
< C        (User-added changes to SUBROUTINE BEHX2)
---
>       COMMON/INSTAB/INDIC
>       COMMON/EIGB4M/EIGCOM(200),EIGNEG(200),EIGCRN
>       COMMON/WVEB4M/NWVCOM(200),NWVNEG(200),IWAVEB,NWVCRN
>       COMMON/EIGBUK/EIGCRT
>       COMMON/NWVBUK/NWVCRT
>       COMMON/BUCKN/N0BX,NMINBX,NMAXBX,INCRBX
>       COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
>       COMMON/EIGALL/EIG2,EIG3,EIG4
>       COMMON/WAVALL/NWAV2,NWAV3,NWAV4
> C BEG FEB 2008
>       COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
>       common/caseblock/CASE
>       CHARACTER*28 CASE
>       CHARACTER*35 CASA
> C END FEB 2008
> C
>       INDIC = 1
>       N0B = 2
>       NMAXB = 10
>       CALL BOSDEC(2,24,ILOADX,INDIC)
> C BEG FEB 2008
>       IF (ITYPEX.EQ.2) THEN
> C      Get CASE.BEHX2 file for input for BIGBOSOR4...
> C      CASE.BEHX2 is an input file for BIGBOSOR4 for behavior no. 2:
> C      Symmetric buckling load factor
>          I=INDEX(CASE,' ')
>          IF(I.NE.0) THEN
>             CASA=CASE(:I-1)//'.BEHX2'
>          ELSE
>             CASA=CASE//'.BEHX2'
>          ENDIF
>          OPEN(UNIT=61,FILE=CASA,STATUS='UNKNOWN')
>          CALL BOSDEC(2,61,ILOADX,INDIC)
>          CLOSE(UNIT=61)
>          WRITE(IFILE,'(/,/,A,A,/,A)')
>      1 ' BIGBOSOR4 input file for:',
>      1 ' Symmetric buckling load factor',
>      1   CASA
>       ENDIF
> C END FEB 2008
>       CALL B4READ
>       IF (IMODX.EQ.0) THEN
>          N0BX = N0B
>          NMINBX = N0B
>          NMAXBX = NMAXB
>            INCRBX = 1
>         ELSE
>            NOBX = NWAV2
>            NMINBX = NWAV2
>            NMAXBX = NWAV2
>            INCRBX = 1
>         ENDIF
>         REWIND IFILE9
>         CALL STOCM1(IFILE9)
>         CALL STOCM2(IFILE9)
>         CALL B4MAIN
>         CALL GASP(DUM1,DUM2,-2,DUM3)
>         IF (IMODX.EQ.0) THEN
>            EIG2 = EIGCRT
>            NWAV2= NWVCRT
>         ENDIF
>         WRITE(IFILE,'(/,A)')
>        1 ' SYMMETRIC BUCKLING LOAD FACTORS AND MODES (BEHX2)'
>         DO 10 I = 1,IWAVEB
>            WRITE(IFILE,'(A,1P,E12.4,A,I4,A)')
>        1  '      ',EIGCOM(I),'(',NWVCOM(I),')'
>    10 CONTINUE
>         WRITE(IFILE,'(A,1P,E12.4)')
>        1' Critical buckling load factor, BSYM=',EIGCRT
>         WRITE(IFILE,'(A,I5)')
>        1' Critical number of circumferential waves, NWVCRT=',NWVCRT
>         BSYM(ILOADX) = EIGCRT
564,565c670,741
< C
< C        (User-added changes to SUBROUTINE BEHX3)
---
>         COMMON/INSTAB/INDIC
>         COMMON/EIGB4M/EIGCOM(200),EIGNEG(200),EIGCRN
>         COMMON/WVEB4M/NWVCOM(200),NWVNEG(200),IWAVEB,NWVCRN
>         COMMON/EIGBUK/EIGCRT
>         COMMON/NWVBUK/NWVCRT
>         COMMON/BUCKN/N0BX,NMINBX,NMAXBX,INCRBX
>         COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
>         COMMON/EIGALL/EIG2,EIG3,EIG4
>         COMMON/WAVALL/NWAV2,NWAV3,NWAV4
> C BEG FEB 2008
>         COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
>         common/caseblock/CASE
>         CHARACTER*28 CASE
>         CHARACTER*35 CASA
> C END FEB 2008
> C
>         INDIC = 1
>       N0B = 2
>       NMAXB = 15
>       CALL BOSDEC(3,24,ILOADX,INDIC)
> C BEG FEB 2008
>       IF (ITYPEX.EQ.2) THEN
> C      Get CASE.BEHX3 file for input for BIGBOSOR4...
> C      CASE.BEHX3 is an input file for BIGBOSOR4 for behavior no. 3:
> C      Antisymmetric buckling load factor
>          I=INDEX(CASE,' ')
>          IF(I.NE.0) THEN
>             CASA=CASE(:I-1)//'.BEHX3'
>          ELSE
>             CASA=CASE//'.BEHX3'
>          ENDIF
>          OPEN(UNIT=61,FILE=CASA,STATUS='UNKNOWN')
>          CALL BOSDEC(3,61,ILOADX,INDIC)
>          CLOSE(UNIT=61)
>          WRITE(IFILE,'(/,/,A,A,/,A)')
>      1 ' BIGBOSOR4 input file for:',
>      1 ' Antisymmetric buckling load factor',
>      1   CASA
>       ENDIF
> C END FEB 2008
>       CALL B4READ
>       IF (IMODX.EQ.0) THEN
>          N0BX = N0B
>          NMINBX = N0B
>          NMAXBX = NMAXB
>          INCRBX = 1
>       ELSE
>          NOBX = NWAV3
>          NMINBX = NWAV3
>          NMAXBX = NWAV3
>          INCRBX = 1
>       ENDIF
>       REWIND IFILE9
>       CALL STOCM1(IFILE9)
>       CALL STOCM2(IFILE9)
>       CALL B4MAIN
>       CALL GASP(DUM1,DUM2,-2,DUM3)
>       IF (IMODX.EQ.0) THEN
>          EIG3 = EIGCRT
>          NWAV3= NWVCRT
>       ENDIF
>       WRITE(IFILE,'(/,A)')
>      1 ' ANTISYMMETRIC BUCKLING LOAD FACTORS AND MODES (BEHX3)'
>       DO 10 I = 1,IWAVEB
>          WRITE(IFILE,'(A,1P,E12.4,A,I4,A)')
>      1  '      ',EIGCOM(I),'(',NWVCOM(I),')'
>    10 CONTINUE
>       WRITE(IFILE,'(A,1P,E12.4)')
>      1' Critical buckling load factor, BANTI=',EIGCRT
>       WRITE(IFILE,'(A,I5)')
>      1' Critical number of circumferential waves, NWVCRT=',NWVCRT
>       BANTI(ILOADX) = EIGCRT
632,633c808,879
< C
< C        (User-added changes to SUBROUTINE BEHX4)
---
>       COMMON/INSTAB/INDIC
>       COMMON/EIGB4M/EIGCOM(200),EIGNEG(200),EIGCRN
>       COMMON/WVEB4M/NWVCOM(200),NWVNEG(200),IWAVEB,NWVCRN
>       COMMON/EIGBUK/EIGCRT
>       COMMON/NWVBUK/NWVCRT
>       COMMON/BUCKN/N0BX,NMINBX,NMAXBX,INCRBX
>       COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
>       COMMON/EIGALL/EIG2,EIG3,EIG4
>       COMMON/WAVALL/NWAV2,NWAV3,NWAV4
> C BEG FEB 2008
>       COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
>       common/caseblock/CASE
>       CHARACTER*28 CASE
>       CHARACTER*35 CASA
> C END FEB 2008
> C
>       INDIC = 2
>       N0B = 2
>       NMAXB = 10
>       CALL BOSDEC(4,24,ILOADX,INDIC)
> C BEG FEB 2008
>       IF (ITYPEX.EQ.2) THEN
> C      Get CASE.BEHX4 file for input for BIGBOSOR4...
> C      CASE.BEHX4 is an input file for BIGBOSOR4 for behavior no. 4:
> C      Fundamental modal frequency (hertz).
>         I=INDEX(CASE,' ')
>         IF(I.NE.0) THEN
>            CASA=CASE(:I-1)//'.BEHX4'
>         ELSE
>            CASA=CASE//'.BEHX4'
>         ENDIF
>         OPEN(UNIT=61,FILE=CASA,STATUS='UNKNOWN')
>         CALL BOSDEC(4,61,ILOADX,INDIC)
>         CLOSE(UNIT=61)
>         WRITE(IFILE,'(/,/,A,A,/,A)')
>      1 ' BIGBOSOR4 input file for:',
>      1 ' Fundamental modal frequency (hertz)',
>      1   CASA
>       ENDIF
> C END FEB 2008
>       CALL B4READ
>       IF (IMODX.EQ.0) THEN
>          N0BX = N0B
>          NMINBX = N0B
>          NMAXBX = NMAXB
>          INCRBX = 1
>       ELSE
>          NOBX = NWAV4
>          NMINBX = NWAV4
>          NMAXBX = NWAV4
>          INCRBX = 1
>       ENDIF
>       REWIND IFILE9
>       CALL STOCM1(IFILE9)
>       CALL STOCM2(IFILE9)
>       CALL B4MAIN
>       CALL GASP(DUM1,DUM2,-2,DUM3)
>       IF (IMODX.EQ.0) THEN
>          EIG4 = EIGCRT
>          NWAV4= NWVCRT
>       ENDIF
>       WRITE(IFILE,'(/,A)')
>      1 ' NATURAL FREQUENCIES AND MODES (BEHX4)'
>       DO 10 I = 1,IWAVEB
>          WRITE(IFILE,'(A,1P,E12.4,A,I4,A)')
>      1  '      ',EIGCOM(I),'(',NWVCOM(I),')'
>    10 CONTINUE
>       WRITE(IFILE,'(A,1P,E12.4)')
>      1' Critical buckling load factor, FREQ=',EIGCRT
>       WRITE(IFILE,'(A,I5)')
>      1' Critical number of circumferential waves, NWVCRT=',NWVCRT
>       FREQ(ILOADX) = EIGCRT
```

--------END OF behavior.diff FILE ------------------

Please see above and see Table a.6 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.

for explanations of the user-added FORTRAN code.