Table A29 How to create **SUBROUTINE BOSDEC**, which generates a valid input file for BIGBOSOR4 (or BOSOR4). The GENOPT user must generate a version of SUBROUTINE BOSDEC (meaning “BOSorDECk”) if he/she plans to use BIGBOSOR4 (or BOSOR4) as part of the software that he/she adds to the skeletal libraries, struct.new and/or behavior.new, that are generated automatically by GENOPT. This table reproduces a file called **/home/progs/genopt/case/cylinder/howto.bosdec** that is part of the documentation provided with GENOPT. The example used here (“cylinder”) is much simpler than the generic case, “equivellipse”, but the guidelines given here are valid for any case in which BIGBOSOR4 (or BOSOR4) is used.
==========

10 March, 2008, revised 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 **********

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

HOW TO CREATE A VALID bosdec.src FILE

This text gives a brief guide to the GENOPT user on how to create a bosdec.src source file. The purpose of bosdec.src is to generate a valid input file for BIGBOSOR4 (or BOSOR4).

First, please read Section A.4 on p. 46 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.4 is entitled, "The best way to create SUBROUTINE BOSDEC".

Basically, the best way to create a valid bosdec.src source file is to first execute BIGBOSOR4 (or BOSOR4) for a typical example that is a member of the class of structures you are using GENOPT to set up an optimization scheme for. You execute the INPUT processor of BIGBOSOR4 (or BOSOR4) and interactively generate a *.ALL (or *.DOC) file that you next can refer to as a guide. Make certain when you do this that the file you create is a VALID one. Execute the entire BIGBOSOR4 (or BOSOR4) commands to determine that your file is valid and that BIGBOSOR4 yields reasonable results. To

do this, you type the following commands:

Command	meaning of the command
bigbosor4log	(you activate bigbosor4 command set)
input	(you provide input data interactively, thus creating a *.ALL file, where "*" represents your name for the case)
bigbosorall	(you execute BIGBOSOR4)
(inspect the *.OUT file to view the results and make certain that they are reasonable)	
bosorplot	(you obtain plots)
cleanup	(you "clean up" the <casename> files)

Note: the processor, "cleanup", not only "cleans up" files, but also creates properly annotated files, *.ALL and *.DOC . Make certain that before you execute "cleanup" your case runs to completion in your execution of "bigbosorall"! Otherwise, you will lose all or part of your input data that you provided interactively. After "cleanup" the *.ALL and *.DOC files are identical.

For example, suppose you interactively create, via the "input" command listed above, a file very similar to the cyl.BEHX2 file identified in /home/progs/genopt/case/cylinder/readme.cylinder as follows:

cyl.BEHX2 = linear buckling, symmetry at the symmetry plane.

You then execute bigbosorall followed by cleanup. You will then have the following cyl.ALL file (assuming that your choice of case name = "cyl" and that your input data are valid for BIGBOSOR4 or BOSOR4). The cyl.ALL file is analogous to the following (you will generally have different values of the input data, but the sequence, type, and meaning of the data will be the same for the generic case "cylinder"):

---------- BEGINNING OF cyl.ALL file ----------

```
 Bifurcation buckling analysis (INDIC=1)
      1   $ INDIC = analysis type indicator
      1   $ NPRT = output options (1=minimum, 2=medium, 3=maximum)
      0   $ ISTRES= output control (0=resultants,1=sigma,2=epsilon)
1   $ NSEG  = number of shell segments (less than 195)
   H      $
   H      $ SEGMENT NUMBER    1    1    1    1    1    1    1    1
   H      $ NODAL POINT DISTRIBUTION FOLLOWS...
     97   $ NMESH = number of node points (5 = min.; 98 = max.)( 1)
      3   $ NTYPEH= control integer (1 or 3) for nodal point spacing
   H      $ REFERENCE SURFACE GEOMETRY FOLLOWS...
      1   $ NSHAPE= indicator (1,2 or 4) for geometry of meridian
100.0000  $ R1     = radius at beginning of segment (see p. 66)
0.000000  $ Z1     = global axial coordinate at beginning of segment
```

```
100.0000  $ R2     = radius at end of segment
100.0000  $ Z2     = global axial coordinate at end of segment
   H      $ IMPERFECTION SHAPE FOLLOWS...
      0   $ IMP   = indicator for imperfection (0=none, 1=some)
   H      $ REFERENCE SURFACE LOCATION RELATIVE TO WALL
      3   $ NTYPEZ= control (1 or 3) for reference surface location
0.3408068 $ ZVAL  = distance from leftmost surf. to reference surf.
   N      $ Do you want to print r(s), r'(s), etc. for this segment?
   H      $ DISCRETE RING INPUT FOLLOWS...
      1   $ NRINGS= number (max=20) of discrete rings in this segment
      2   $ NTYPE = control for identification of ring location (2=z, 3=r)
      0   $ Z(I)  = axial coordinate of Ith ring, z( 1)
      0   $ NTYPER= type (0 or 1 or 2 or 4 or 5) of discrete ring no.( 1)
      0   $ K=elastic foundation modulus (e.g. lb/in**3)in this seg.
   H      $ LINE LOAD INPUT FOLLOWS...
      1   $ LINTYP= indicator (0, 1, 2 or 3) for type of line loads
      1   $ NLOAD(1)=indicator for axial load or disp.(0=none,1=some)
      0   $ NLOAD(2)=indicator for shear load or disp.(0=none,1=some)
      0   $ NLOAD(3)=indicator for radial load or disp.(0 or 1)
      0   $ NLOAD(4)=indicator for line moment or rotation (0 or 1)
0.000000  $ V(i)=fixed or initial axial load or displacement, V( 1)
      1   $ NLOAD(1)=indicator for axial load or disp. increment(0 or 1)
      0   $ NLOAD(2)= should be zero
      0   $ NLOAD(3)=indicator for radial load or disp. increment(0 or 1)
      0   $ NLOAD(4)=indicator for moment or rot. increment (0 or 1)-
1.000000  $ DV(i)=axial load or displacement increment, DV( 1)
   H      $ DISTRIBUTED LOAD INPUT FOLLOWS...
      1   $ IDISAB= indicator (0, 1, 2 or 3) for load set A and B
   H      $ SURFACE LOAD INPUT FOR LOAD SET "A" FOLLOWS
      1   $ NLTYPE=control (0,1,2,3) for type of surface loading
      2   $ NPSTAT= number of meridional callouts for surface loading
      0   $ NLOAD(1)=indicator for meridional traction (0=none, 1=some)
      0   $ NLOAD(2)=indicator for circumferential traction
      1   $ NLOAD(3)=indicator for normal pressure     (0=none, 1=some)
1.000000  $ PN(i)   = normal pressure (p.74) at ith callout, PN( 1)
1.000000  $ PN(i)   = normal pressure (p.74) at ith callout, PN( 2)
      2   $ NTYPE = control for meaning of loading callout (2=z, 3=r)
0.000000  $ Z(I)  = axial coordinate of Ith loading callout, z( 1)
100.0000  $ Z(I)  = axial coordinate of Ith loading callout, z( 2)
   H      $ SHELL WALL CONSTRUCTION FOLLOWS...
      2   $ NWALL=index (1, 2, 4, 5, 6, 7, 8, 9, 10) for wall construction
0.100E+08 $ E     = Young's modulus for skin
0.300     $ U     = Poisson's ratio for skin
0.250E-03 $ SM =mass density of skin (e.g. alum.=.00025 lb-sec**2/in**4)
0.000000  $ ALPHA = coefficient of thermal expansion
      0   $ NRS  = control (0 or 1) for addition of smeared stiffeners
      0   $ NSUR   = control for thickness input (0 or 1 or -1)
   N      $ Do you want to print out ref. surf. location and thickness?
```

```
   N      $ Do you want to print out the C(i,j) at meridional stations?
   N      $ Do you want to print out distributed loads along meridian?
   H      $
   H      $ GLOBAL DATA BEGINS...
      0   $ NLAST = plot options  (-1=none, 0=geometry, 1=u,v,w)
   N      $ Are there any regions for which you want expanded plots?
      2   $ N0B   = starting number of circ. waves (buckling analysis)
      2   $ NMINB = minimum number of circ. waves  (buckling analysis)
     10   $ NMAXB = maximum number of circ. waves  (buckling analysis)
      1   $ INCRB = increment in number of circ. waves (buckling)
      1   $ NVEC  = number of eigenvalues for each wave number
0.000000  $ P      = pressure or surface traction multiplier
-0.20E-01 $ DP     = pressure or surface traction multiplier increment
0.000000  $ TEMP   = temperature rise multiplier
0.000000  $ DTEMP  = temperature rise multiplier increment
0.000000  $ OMEGA  = angular vel. about axis of revolution (rad/sec)
0.000000  $ DOMEGA = angular velocity increment (rad/sec)
   H      $ CONSTRAINT CONDITIONS FOLLOW....
      1   $ How many segments in the structure?
   H      $
   H      $ CONSTRAINT CONDITIONS FOR SEGMENT NO.    1    1    1    1
   H      $ POLES INPUT FOLLOWS...
      0   $ Number of poles (places where r=0) in SEGMENT( 1)
   H      $ INPUT FOR CONSTRAINTS TO GROUND FOLLOWS...
      2   $ At how many stations is this segment constrained to ground?
      1   $ INODE = nodal point number of constraint to ground, INODE( 1)
      0   $ IUSTAR=axial displacement constraint (0 or 1 or 2)
      1   $ IVSTAR=circumferential displacement(0=free,1=0,2=imposed)
      1   $ IWSTAR=radial displacement(0=free,1=constrained,2=imposed)
      1   $ ICHI=meridional rotation (0=free,1=constrained,2=imposed)
0.000000  $ D1    = radial component of offset of ground support
0.000000  $ D2    = axial component of offset of ground support
   N      $ Is this constraint the same for both prebuckling and buckling?
      1   $ IUSTARB= axial displacement for buckling or vibration phase
      1   $ IVSTARB= circ. displacement for buckling or vibration phase
      1   $ IWSTARB= radial displacement for buckling or vibration
      1   $ ICHIB  = meridional rotation for buckling or vibration
     97   $ INODE = nodal point number of constraint to ground, INODE( 2)
      1   $ IUSTAR=axial displacement constraint (0 or 1 or 2)
      0   $ IVSTAR=circumferential displacement(0=free,1=0,2=imposed)
      0   $ IWSTAR=radial displacement(0=free,1=constrained,2=imposed)
      1   $ ICHI=meridional rotation (0=free,1=constrained,2=imposed)
0.000000  $ D1    = radial component of offset of ground support
0.000000  $ D2    = axial component of offset of ground support
   Y      $ Is this constraint the same for both prebuckling and buckling?
   H      $ JUNCTION CONDITION INPUT FOLLOWS...
   N      $ Is this segment joined to any lower-numbered segments?
   H      $ RIGID BODY CONSTRAINT INPUT FOLLOWS...
```

```
   N      $ Given existing constraints, are rigid body modes possible?
   H      $ "GLOBAL3" QUESTIONS (AT END OF CASE)...
   Y      $ Do you want to list output for segment( 1)
   Y      $ Do you want to list forces in the discrete rings, if any?
```
---------- END OF cyl.ALL file ----------

Now you want to create a bosdec.src file that generates the same sequence of input data as in the above file. **NOTE: YOU DO NOT HAVE TO WRITE CODE THAT REPRODUCES ALL THE TEXT ON EACH LINE THAT FOLLOWS THE DOLLAR SIGN.** However, it is best to have your bosdec.src code write the dollar signs followed by a space and some short comment about each input datum as has been done in the version of bosdec.src that produces the BIGBOSOR4 input data file listed next.

For example, the cyl.ALL file produced by my version of bosdec.src (called bosdec.cylinder in the directory /home/progs/genopt/case/cylinder) appears as follows (corresponding to cyl.BEHX2 = linear buckling with symmetry at the symmetry plane, that is, “behavior” no. 2):

------BEGINNING OF MY cyl.ALL (cyl.BEHX2) file produced by /home/prog/genopt/case/cylinder/bosdec.cylinder-------

```
 Bifurcation buckling analysis (INDIC=1)
  1 $ INDIC
  1      $ NPRT
  0 $ ISTRES
  1      $ NSEG
 97      $ NMESH
  3      $ NTYPEH
  1      $ NSHAPE
  1.000000E+02 $ R1
 0.      $ Z1
  1.000000E+02 $ R2
  1.000000E+02 $ Z2
  0      $ IMP
  3      $ NTYPEZ
  3.408068E-01 $ THICK
  N      $ print r(s)...?
  1      $ NRINGS
  2      $ NTYPE
  0      $ Z(I)
  0      $ NTYPER
  0      $ K
  1      $ LINTYP
  1      $ NLOAD(1)
  0      $ NLOAD(2)
  0      $ NLOAD(3)
  0      $ NLOAD(4)
  0.     $ V(1)
```

```
 1      $ NLOAD(1)
 0      $ NLOAD(2)
 0      $ NLOAD(3)
 0      $ NLOAD(4)
-1.000000E+00 $ DV(1)
 1      $ IDISAB
 1      $ NLTYPE
 2      $ NPSTAT
 0      $ NLOAD(1)
 0      $ NLOAD(2)
 1      $ NLOAD(3)
 1.     $ PN(1)
 1.     $ PN(2)
 2      $ NTYPE
 0.     $ Z(1)
 1.000000E+02 $ Z(2)
 2      $ NWALL
 1.000000E+07 $ E
 3.000000E-01 $ U
 2.500000E-04 $ SM
 0.     $ ALPHA
 0      $ NRS
 0      $ NSUR
 N      $ print refsurf...?
 N      $ print Cij?
 N      $ print loads?
 0      $ NLAST
 N      $ expanded plots?
 2      $ N0B
 2      $ NMINB
10      $ NMAXB
 1      $ INCRB
 1      $ NVEC
 0.   $ P
-2.000000E-02 $ DP
 0.     $ TEMP
 0.     $ DTEMP
 0.     $ OMEGA
 0.     $ DOMEGA
 1      $ how many segs?
 0      $ number of poles
 2      $ how many stations?
 1      $ INODE
 0      $ IUSTAR
 1      $ IVSTAR
 1      $ IWSTAR
 1 $ ICHI
 0.     $ D1
```

```
  0.      $ D2
  N       $ bc same for pre,bif?
  1 $ IUSTARB
  1       $ IVSTARB
  1       $ IWSTARB
  1 $ ICHIB
 97       $ INODE
  1       $ IUSTAR
  0       $ IVSTAR
  0       $ IWSTAR
  1       $ ICHI
  0.      $ D1
  0.      $ D2
  Y       $ bc same for pre,bif?
  N       $ joined to lower segs?
  N       $ rigid body possible?
  Y       $ output for seg. 1?
  Y       $ output for rings?
```

----- END OF cyl.ALL (cyl.BEHX2) file produced by bosdec.cylinder -----

This file works with BIGBOSOR4. When you execute "bigbosorall" with this file as input followed by execution of "cleanup", you obtain a cyl.ALL file that is identical to the properly annotated file listed just before that just given.

My version of /home/progs/genopt/case/cylinder/bosdec.src (called bosdec.cylinder here) is as follows:

--------- BEGINNING OF bosdec.cylinder (or bosdec.src) ---------------

```
C=DECK      BOSDEC
C
C  PURPOSE IS TO SET UP BOSOR4 INPUT FILE FOR "cylinder" (cyl)
C
      SUBROUTINE BOSDEC(INDX,IFIL14,ILOADX,INDIC)
C  Insert the labelled common blocks that are contained in the file:
C/home/progs/genopt/case/cylinder/cylinder.COM
      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  end of cylinder.COM
      COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
C
      REWIND IFIL14
C
      WRITE(IFILE4,3)
    3 FORMAT(//'  ****************  BOSDEC  *******************'/
     1'  The purpose of BOSDEC is to set up an input file, NAME.ALL,'/
     1'  for a cylindrical shell. NAME is your name for'/
     1'  the case. The file NAME.ALL is a BOSOR4 input "deck" used'/
     1'  by SUBROUTINE B4READ.'/
     1'  ********************************************'/)
C
      IF (INDIC.EQ.0) WRITE(IFIL14,'(A)')
     1' Nonlinear axisymmetric stress analysis (INDIC=0)'
      IF (INDIC.EQ.1) WRITE(IFIL14,'(A)')
     1' Bifurcation buckling analysis (INDIC=1)'
      IF (INDIC.EQ.2) WRITE(IFIL14,'(A)')
     1' Modal vibration of prestressed shell'
      WRITE(IFIL14,'(I3,A)') INDIC,  ' $ INDIC'
      WRITE(IFIL14,'(A)')'  1      $ NPRT'
      ISTRES = 0
      IF (INDIC.EQ.0) ISTRES = 1
      WRITE(IFIL14,'(I3,A)') ISTRES,  ' $ ISTRES'
      WRITE(IFIL14,'(A)')'  1      $ NSEG'
      WRITE(IFIL14,'(A)')' 97      $ NMESH'
      WRITE(IFIL14,'(A)')'  3      $ NTYPEH'
      WRITE(IFIL14,'(A)')'  1      $ NSHAPE'
      WRITE(IFIL14,'(1P,E14.6,A)') RADIUS, ' $ R1'
      WRITE(IFIL14,'(A)')' 0.      $ Z1'
      WRITE(IFIL14,'(1P,E14.6,A)') RADIUS, ' $ R2'
      AXIAL = 0.5*LENGTH
      IF (INDIC.EQ.0.AND.INDX.EQ.1) THEN
         BLL = 2.73*SQRT(RADIUS*THICK)
         AXIAL = MIN(0.5*LENGTH,BLL)
      ENDIF
      WRITE(IFIL14,'(1P,E14.6,A)') AXIAL, ' $ Z2'
      WRITE(IFIL14,'(A)')'  0      $ IMP'
      WRITE(IFIL14,'(A)')'  3      $ NTYPEZ'
      WRITE(IFIL14,'(1P,E14.6,A)') 0.5*THICK, ' $ THICK'
      WRITE(IFIL14,'(A)')'  N      $ print r(s)...?'
      WRITE(IFIL14,'(A)')'  1      $ NRINGS'
      WRITE(IFIL14,'(A)')'  2      $ NTYPE'
```

```
      WRITE(IFIL14,'(A)')'  0      $ Z(I)'
      WRITE(IFIL14,'(A)')'  0      $ NTYPER'
      WRITE(IFIL14,'(A)')'  0      $ K'
      WRITE(IFIL14,'(A)')'  1      $ LINTYP'
      WRITE(IFIL14,'(A)')'  1      $ NLOAD(1)'
      WRITE(IFIL14,'(A)')'  0      $ NLOAD(2)'
      WRITE(IFIL14,'(A)')'  0      $ NLOAD(3)'
      WRITE(IFIL14,'(A)')'  0      $ NLOAD(4)'
      IF (INDIC.EQ.0.OR.INDIC.EQ.2)
     1WRITE(IFIL14,'(1P,E14.6,A)') NX(ILOADX),  ' $ V(1)'
      IF (INDIC.EQ.1) WRITE(IFIL14,'(A)')'  0.     $ V(1)'
      WRITE(IFIL14,'(A)')'  1      $ NLOAD(1)'
      WRITE(IFIL14,'(A)')'  0      $ NLOAD(2)'
      WRITE(IFIL14,'(A)')'  0      $ NLOAD(3)'
      WRITE(IFIL14,'(A)')'  0      $ NLOAD(4)'
      IF (INDIC.EQ.0)
     1WRITE(IFIL14,'(1P,E14.6,A)') NX(ILOADX),  ' $ DV(1)'
      IF (INDIC.EQ.2) WRITE(IFIL14,'(A)')'  0.   $ DV(1)'
      IF (INDIC.EQ.1)
     1WRITE(IFIL14,'(1P,E14.6,A)') NX(ILOADX)/1000.,  ' $ DV(1)'
      WRITE(IFIL14,'(A)')'  1      $ IDISAB'
      WRITE(IFIL14,'(A)')'  1      $ NLTYPE'
      WRITE(IFIL14,'(A)')'  2      $ NPSTAT'
      WRITE(IFIL14,'(A)')'  0      $ NLOAD(1)'
      WRITE(IFIL14,'(A)')'  0      $ NLOAD(2)'
      WRITE(IFIL14,'(A)')'  1      $ NLOAD(3)'
      WRITE(IFIL14,'(A)')'  1.     $ PN(1)'
      WRITE(IFIL14,'(A)')'  1.     $ PN(2)'
      WRITE(IFIL14,'(A)')'  2      $ NTYPE'
      WRITE(IFIL14,'(A)')'  0.     $ Z(1)'
      WRITE(IFIL14,'(1P,E14.6,A)')  AXIAL,  ' $ Z(2)'
      WRITE(IFIL14,'(A)')'  2      $ NWALL'
      WRITE(IFIL14,'(1P,E14.6,A)') ESTIFF,   ' $ E'
      WRITE(IFIL14,'(1P,E14.6,A)') NU,       ' $ U'
      WRITE(IFIL14,'(1P,E14.6,A)') DENS,     ' $ SM'
      WRITE(IFIL14,'(A)')'  0.     $ ALPHA'
      WRITE(IFIL14,'(A)')'  0      $ NRS'
      WRITE(IFIL14,'(A)')'  0      $ NSUR'
C
      WRITE(IFIL14,'(A)')'  N      $ print refsurf...?'
      WRITE(IFIL14,'(A)')'  N      $ print Cij?'
      WRITE(IFIL14,'(A)')'  N      $ print loads?'
      WRITE(IFIL14,'(A)')'  0      $ NLAST'
      WRITE(IFIL14,'(A)')'  N      $ expanded plots?'
      IF (INDIC.NE.0) THEN
         WRITE(IFIL14,'(A)')'  2      $ N0B'
         WRITE(IFIL14,'(A)')'  2      $ NMINB'
         IF (INDX.NE.3) WRITE(IFIL14,'(A)')' 10      $ NMAXB'
```

```
         IF (INDX.EQ.3) WRITE(IFIL14,'(A)')' 15      $ NMAXB'
         WRITE(IFIL14,'(A)')'  1      $ INCRB'
         WRITE(IFIL14,'(A)')'  1      $ NVEC'
       ENDIF
       IF (INDIC.EQ.0.OR.INDIC.EQ.2)
      1WRITE(IFIL14,'(1P,E14.6,A)') PRESS(ILOADX),  ' $ P'
       IF (INDIC.EQ.1) WRITE(IFIL14,'(A)')'  0.   $ P'
       IF (INDIC.EQ.0.OR.INDIC.EQ.1)
      1WRITE(IFIL14,'(1P,E14.6,A)') PRESS(ILOADX)/1000.,  ' $ DP'
       WRITE(IFIL14,'(A)')'  0.     $ TEMP'
       IF (INDIC.EQ.0.OR.INDIC.EQ.1)
      1WRITE(IFIL14,'(A)')'  0.     $ DTEMP'
       IF (INDIC.EQ.0) WRITE(IFIL14,'(A)')'  1     $ NSTEPS'
       WRITE(IFIL14,'(A)')'  0.     $ OMEGA'
       IF (INDIC.EQ.0.OR.INDIC.EQ.1)
      1WRITE(IFIL14,'(A)')'  0.     $ DOMEGA'
C
       WRITE(IFIL14,'(A)')'  1      $ how many segs?'
C
       WRITE(IFIL14,'(A)')'  0      $ number of poles'
       WRITE(IFIL14,'(A)')'  2      $ how many stations?'
       WRITE(IFIL14,'(A)')'  1      $ INODE'
       WRITE(IFIL14,'(A)')'  0      $ IUSTAR'
       WRITE(IFIL14,'(A)')'  1      $ IVSTAR'
       WRITE(IFIL14,'(A)')'  1      $ IWSTAR'
       ICHI = 0
       IF (IBOUND.EQ.2) ICHI = 1
       WRITE(IFIL14,'(I3,A)')  ICHI,  ' $ ICHI'
       WRITE(IFIL14,'(A)')'  0.     $ D1'
       WRITE(IFIL14,'(A)')'  0.     $ D2'
       WRITE(IFIL14,'(A)')'  N      $ bc same for pre,bif?'
       IUSTARB = 0
       IF (IBOUND.EQ.2) IUSTARB = 1
       WRITE(IFIL14,'(I3,A)') IUSTARB,  ' $ IUSTARB'
       WRITE(IFIL14,'(A)')'  1      $ IVSTARB'
       WRITE(IFIL14,'(A)')'  1      $ IWSTARB'
       ICHIB = 0
       IF (IBOUND.EQ.2) ICHIB = 1
       WRITE(IFIL14,'(I3,A)')  ICHIB,  ' $ ICHIB'
C
       WRITE(IFIL14,'(A)')' 97      $ INODE'
       WRITE(IFIL14,'(A)')'  1      $ IUSTAR'
       WRITE(IFIL14,'(A)')'  0      $ IVSTAR'
       WRITE(IFIL14,'(A)')'  0      $ IWSTAR'
       WRITE(IFIL14,'(A)')'  1      $ ICHI'
       WRITE(IFIL14,'(A)')'  0.     $ D1'
       WRITE(IFIL14,'(A)')'  0.     $ D2'
       IF (INDX.NE.3) THEN
```

```
         WRITE(IFIL14,'(A)')'  Y      $ bc same for pre,bif?'
      ELSE
         WRITE(IFIL14,'(A)')'  N      $ bc same for pre,bif?'
         WRITE(IFIL14,'(A)')'  0      $ IUSTARB'
         WRITE(IFIL14,'(A)')'  1      $ IVSTARB'
         WRITE(IFIL14,'(A)')'  1      $ IWSTARB'
         WRITE(IFIL14,'(A)')'  0      $ ICHIB'
      ENDIF
      WRITE(IFIL14,'(A)')'  N      $ joined to lower segs?'
      WRITE(IFIL14,'(A)')'  N      $ rigid body possible?'
      WRITE(IFIL14,'(A)')'  Y      $ output for seg. 1?'
      WRITE(IFIL14,'(A)')'  Y      $ output for rings?'
C
      RETURN
      END
```

------- END OF MY VERSION of bosdec.cylinder (or bosdec.src)-------

NOTE THE FOLLOWING:

1. There are labeled common blocks inserted at the beginning of bosdec.src. These represent the contents of the file, **/home/progs/genopt/case/cylinder/cylinder.COM,** that is generated automatically during the GENTEXT interactive session corresponding to the generic case, “cylinder”. It must be inserted at the beginning of any bosdec.src source file.

2. The labeled common block,

```
      COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
```

is from the BIGBOSOR4 program and is needed here only for the index, IFILE4, that is, to generate the printed output about BOSDEC near the beginning of the bosdec.src routine listed above.

3. There are "IF" statements in which the conditional index may be IBOUND and/or INDIC and/or INDX. The valid input data for BIGBOSOR4 depend on the values of these indices, which are different corresponding to the different behaviors, BEHX1, BEHX2, BEHX3, BEHX4, that will act as constraints on the optimum design and that were originally chosen by the GENOPT user See the files,

/home/progs/genopt/case/cylinder/cylinder.INP

and

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

Associated with BEHX1 are INDIC=0 and INDX=1 (nonlinear stress analysis)
Associated with BEHX2 are INDIC=1 and INDX=2 (buckling with symmetry)
Associated with BEHX3 are INDIC=1 and INDX=3 (buckling with antisymmetry)
Associated with BEHX4 are INDIC=2 and INDX=4 (modal vibration)

The boundary condition, IBOUND, is provided in the file,
/home/progs/genopt/case/cylinder/cyl.BEG:

IBOUND = 1 means simple support at ends
IBOUND = 2 means clamped at ends

This is a simple case with only one shell segment in the BIGBOSOR4 model.
What if you have many, many shell segments, such as is the case for
/home/progs/genopt/case/wavycyl [7]? Then you may want to program
bosdec.src with a loop over the segment number, such as is done in
/home/progs/genopt/case/wavycyl/bosdec.wavycyl [7].
==========