The postscript file is called "metafile.ps"(Fig. 26).)

```
cleanup          (clean up BIGBOSOR4 files called "test" and
                  generate the version of test.ALL and test.DOC
                  with proper annotation throughout.)
```

(Next, we wish to obtain a plot of the critical GENERAL buckling mode from execution of BIGBOSOR4 and BOSORPLOT for the case with all factors of safety greater than 1.0:)

(Copy the BIGBOSOR4 input file for GENERAL buckling, test.BEHX2, into test.ALL because BIGBOSOR4 input files must always have the three-letter suffix, ".ALL":)

```
cp test.BEHX2 test.ALL

bigbosorall      (Start "batch" run for GENERAL buckling. The output
                  file that you want to inspect is called test.OUT.
                  This will be a very, very long file, so search specifically
                  for the string, "EIGENVALUE(", typed with the
                  "(" at the end of the string. You will find the
                  following output there:
---------------------------------------------------------------------------
 **** CRITICAL EIGENVALUE AND WAVENUMBER ****
 EIGCRT=  2.0048E+00; NO. OF CIRC. WAVES, NWVCRT=  300
 ****************************************************

 ***** EIGENVALUES AND MODE SHAPES *****
  EIGENVALUE(CIRC. WAVES)
 ======================================
    2.5384E+00( 100)
    2.0644E+00( 200)
    2.0048E+00( 300)
    2.1052E+00( 400)
    2.2525E+00( 500)
 ======================================
---------------------------------------------------------------------------

bosorplot        (obtain a plot of the critical GENERAL buckling mode.
                  The postscript file is called "metafile.ps"(Fig. 27).)

cleanup          (clean up BIGBOSOR4 files called "test.*)
```

(Next, return to the "genoptcase" directory, and continue processing the SPECIFIC case called "test".)

```
cd /home/progs/genoptcase
```

(Next, do a convergence study of the prediction of local buckling vs the number of modules, NMODULL, used in the local buckling model). see p. 16

```
cleanspec        (clean up the SPECIFIC files, test.*)
```

(Edit the test.BEG file to change the number of modules used for local buckling from NMODULL = 1 to NMODULL = 2 .)

```
begin            (execute BEGIN, using the file, test.BEG, for input
                  Table 14 with Table 27 changes in factors of safety
                  and with NMODULL changed from 1 to 2)

change           (execute CHANGE, using the file, test.CHG for input (Table 29))

decide           (execute DECIDE, using the file, test.DEC, for input (Table 15))

mainsetup        (execute MAINSETUP, using the file, test.OPT, for input.
                  Make sure that NPRINT = 2 and ITYPE = 2 in test.OPT.)

optimize         (generate the test.OPM file for the fixed design)

cp test.BEHX1 /home/progs/bigbosor4/workspace/test.ALL (LOCAL buckling input)

cd /home/progs/bigbosor4/workspace

bigbosor4log     (activate BIGBOSOR4 commands)

bigbosorall      (execute BIGBOSOR4)
```

Table 8, p.15 /18

(Search for the string, "EIGENVALUE(" and find eigenvalue vx. circumferential waves)

```
bosorplot       (get the file, metafile.ps, for 700 circ. waves (Fig. 28))

cleanup         (clean up BIGBOSOR4 files, test.*)

cd /home/progs/genoptcase

cleanspec       (clean up the SPECIFIC files, test.*)
```

(Edit the test.BEG file to change the number of modules used for local buckling from NMODULL = 2 to NMODULL = 3 .)

```
begin           (execute BEGIN, using the file, test.BEG, for input)

change          (execute CHANGE, using the file, test.CHG for input (Table 29))

decide          (execute DECIDE, using the file, test.DEC, for input (Table 15))

mainsetup       (execute MAINSETUP, using the file, test.OPT, for input.
                 Make sure that NPRINT = 2 and ITYPE = 2 in test.OPT.)

optimize        (generate the test.OPM file for the fixed design)

cp test.BEHX1 /home/progs/bigbosor4/workspace/test.ALL (LOCAL buckling input)

cd /home/progs/bigbosor4/workspace

bigbosor4log    (activate BIGBOSOR4 commands)

bigbosorall     (execute BIGBOSOR4)
```

(Search for the string, "EIGENVALUE(" and find eigenvalue vx. circumferential waves)

```
bosorplot       (get the file, metafile.ps, for 700 circ. waves (Fig. 29))

cleanup         (clean up BIGBOSOR4 files, test.*)

cd /home/progs/genoptcase

cleanspec       (clean up the SPECIFIC files, test.*)
```

(Edit the test.BEG file to change the number of modules used for local buckling from NMODULL = 3 to NMODULL = 13 (maximum permitted).)

```
begin           (execute BEGIN, using the file, test.BEG, for input)

change          (execute CHANGE, using the file, test.CHG for input (Table 29))

decide          (execute DECIDE, using the file, test.DEC, for input (Table 15))

mainsetup       (execute MAINSETUP, using the file, test.OPT, for input.
                 Make sure that NPRINT = 2 and ITYPE = 2 in test.OPT.)

optimize        (generate the test.OPM file for the fixed design)

cp test.BEHX1 /home/progs/bigbosor4/workspace/test.ALL (LOCAL buckling input)

cd /home/progs/bigbosor4/workspace

bigbosor4log    (activate BIGBOSOR4 commands)

bigbosorall     (execute BIGBOSOR4)
```

(Search for the string, "EIGENVALUE(" and find eigenvalue vx. circumferential waves)

```
bosorplot       (get the file, metafile.ps, for 700 circ. waves (Fig. 30))

cleanup         (clean up BIGBOSOR4 files, test.*)
```

[Next we wish to compare predictions for local buckling of the optimized design listed in Table 28 from a BIGBOSOR4 model and from a PANDA2 model. For local buckling of a truss-core sandwich, PANDA2 uses a single discretized see p.16

module model that very much resembles the single general buckling module used in this "trusscomp" project. A sketch of the single module model used in PANDA2 is presented on page 1 of Table 7. The output for local buckling from PANDA2 for the optimum design obtained in the "nasatruss" project appears at the top of page 3 of Table 7. We wish to generate that type of output from PANDA2 for the optimum "trusscomp" design listed in Table 28. Then we want to compare the PANDA2 predictions with those from BIGBOSOR4 for the optimum "trusscomp" configuration modeled as is done for general buckling, that is, a multi-module model with only six shell segments in each module, as depicted in Fig. 4 and at the top of Fig. 6. The following run stream will accomplish this.]

[First, find where the "nasatruss" files, nasatruss.BEG, nasatruss.DEC, and nasatruss.OPT are stored. (e.g. ...genopt/case/truss3) The "nasatruss" files that we want are contained in the nasatruss.tar file. Copy this file to a working space for PANDA2 executions and "untar" it:

```
cp /home/progs/panda2/case/truss3/nasatruss.tar /home/progs/panda2/workspace/.

cd /home/progs/panda2/workspace

tar xvf nasatruss.tar
```

[Find nasatruss.BEG, nasatruss.DEC, nasatruss.OPT, and copy these files to files with a relevant name, such as comptruss.BEG, etc:)

```
cp nasatruss.BEG comptruss.BEG
cp nasatruss.DEC comptruss.DEC
cp nasatruss.OPT comptruss.OPT
```

(Edit the comptruss.BEG file by replacing three values there with the appropriate optimized values listed on the first page of Table 28. The three quantities in the nasatruss.BEG file,

```
  3.700000      $ pitch of truss core, b
 0.7000000      $ width over which truss core contacts each face sheet, b2
  1.400000      $ height of truss, h
```

must be replaced by the following optimum values from the "trusscomp" project:

```
  2.810600      $ pitch of truss core, b
  1.000400      $ width over which truss core contacts each face sheet, b2
  0.888720      $ height of truss, h
```

The rest of the comptruss file needs no further changes. The comptruss.DEC file needs no changing. In the comptruss.OPT file make sure that the initial imperfection components are all zero and that ITYPE = 2 (analysis of a fixed design). Then give the following commnds:]

```
panda2log       (activate the PANDA2 command set)

begin           (use comptruss.BEG as input)

setup           (no input required from the user)

decide          (use comptruss.DEC as input)

mainsetup       (use comptruss.OPT, suitably edited, as input)

pandaopt        (execute PANDA2 in the fixed design (ITYPE=2) mode)
```

[Inspect the test.OPM file. In particular, we are interested in local buckling results for which the PANDA2 single discretized module model is used. Search for the string, "EIGOLD". This will lead you to the following output in the test.OPM file obtained from the execution of pandaopt:]

```
----------------------------------------------------------------------------
 BUCKLING LOAD FACTORS FROM BOSOR4-TYPE DISCRETIZED MODEL...
          (skin-stringer discretized module of local buckling)
  AXIAL     BUCKLING         KNOCKDOWN FOR    KNOCKDOWN FOR      BUCKLING
  HALF-    LOAD FACTOR      TRANSVERSE SHEAR  IN-PLANE SHEAR    LOAD FACTOR
  WAVES  BEFORE KNOCKDOWN     DEFORMATION     LOADING AND/OR  AFTER KNOCKDOWN
                                                ANISOTROPY
    M         EIGOLD              KSTAR          KNOCK        EIGOLD*KSTAR*KNOCK
   57      1.26900E+00        1.00000E+00     9.97192E-01      1.26543E+00
   61      1.24883E+00        1.00000E+00     9.97192E-01      1.24532E+00
   66      1.24861E+00        1.00000E+00     9.97192E-01      1.24510E+00
```

```
   71      1.23999E+00      1.00000E+00      9.97192E-01      1.23651E+00
   76      1.25295E+00      1.00000E+00      9.97192E-01      1.24943E+00
 Buckling load factor before t.s.d.=  1.2365E+00 After t.s.d.=  1.1674E+00
   70      1.23999E+00      9.44586E-01      9.97192E-01      1.16736E+00
 ---------------------------------------------------------------------------
```

(The critical local buckling mode has 70 axial half-waves over the 96-inch length of the cylindrical shell.

This same PANDA2 output appears at the bottom of Table 31. Next, we want to compare these PANDA2 results with "equivalent" results from our use of the "trusscomp" project procedures.)

(Next, return to the "genoptcase" directory, and continue processing the SPECIFIC case called "test".)

```
cd /home/progs/genoptcase

genoptlog        (activate the GENOPT commands)

cleanspec        (clean up the "test" files)
```

[Edit the begin file, test.BEG, listed in Table 14. Set very small values for ENOODL and DNOODL, as follows:

```
0.1851110E+00 $ axial modulus of a corner "noodle": ENOODL
0.5700000E-08 $ weight density of the "noodle" material: DNOODL
```

This gets rid of the effect of the "noodles", which are not present in the PANDA2 model of the composite truss-core sandwich.]

```
begin            (execute begin, using the suitably edited test.BEG file))

change           (execute CHANGE with the test.CHG file
                  listed in Table 29 as input)

decide           (execute DECIDE with test.DEC in Table 15 as input)

mainsetup        (execute MAINSETUP with test.OPT as listed at the
                  top of Table 28 as input)

optimize         (execute OPTIMIZE in the ITYPE=2 mode of operation)
```

[OPTIMIZE with ITYPE = 2 generates the two valid input files for BIGBOSOR4, test.BEHX1 and test.BEHX2. Now we are interested only in the file, test.BEHX2 (general buckling model). That is because the general buckling single module (6 shell segments) is very similar to the discretized single module model used by PANDA2 for computing local buckling.)

```
cp /home/progs/genoptcase/test.BEHX2 /home/progs/bigbosor4/workspace/test.ALL

cd /home/progs/bigbosor4/workspace

bigbosor4log     (activate BIGBOSOR4 commands)

bigbosorall      (execute BIGBOSOR4 with test.ALL as input)
```

[This execution of bigbosorall will yield the output file, test.OUT which contains the following general buckling loads:]

```
 ---------------------------------------------------------------
 **** CRITICAL EIGENVALUE AND WAVENUMBER ****
 EIGCRT=  1.9480E+00; NO. OF CIRC. WAVES, NWVCRT=  300
 **************************************************

 ***** EIGENVALUES AND MODE SHAPES *****
  EIGENVALUE(CIRC. WAVES)
 ======================================
    2.4656E+00( 100)
    2.0161E+00( 200)
    1.9480E+00( 300)
    2.0364E+00( 400)
    2.1631E+00( 500)
 ======================================
-----------------------------------------------------------------
```

[The eigenvalues above are slightly lower than those listed for general buckling on page 2 of Table 28 because we have taken away the "noodles" by suitably editing the test.BEG file, as described earlier.]

[For now we are not interested in the general buckling eigenvalues because we want to simulate local buckling with use of the general buckling "trusscomp" model. To obtain local buckling eigenvalues we must compute results for a much higher range of wave numbers. PANDA2 estimates that there are about 70 axial half-waves along the 96-inch length of the cylindrical shell. 70 axial half-waves over the 96-inch length of the cylindrical shell corresponds to 7000 circumferential waves around the "huge torus" (see Appendix 1 for an explanation of the factor of 100). In order to explore the high-N range of wave numbers, we must edit the test.ALL file. However, the version of test.ALL generated by SUBROUTINE BEHX2, called test.BEHX2, is not properly annotated. Therefore, it is difficult for the user to find where to do the suitable editing. In order to obtain a suitably annotated test.ALL file, you must type the following command:]

```
cleanup          (deletes unneeded BIGBOSOR4 "test" files and
                  generates a properly annotated test.ALL
                  file and a properly annotated test.DOC file.
                  Table 18 is an abridged version, except for
                  the "noodle" axial stiffness, ENOODL, and
                  density, DNOODL, and therefore also the
                  "noodle" prebuckling loads.)
```

[In the properly annotated test.ALL file we find the following lines:]

```
----------------------------------------------------------------------
      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?
      100       $ N0B   = starting number of circ. waves (buckling analysis)
      100       $ NMINB = minimum number of circ. waves  (buckling analysis)
      500       $ NMAXB = maximum number of circ. waves  (buckling analysis)
      100       $ INCRB = increment in number of circ. waves (buckling)
        1       $ NVEC  = number of eigenvalues for each wave number
----------------------------------------------------------------------
```

[We arrive at this section in the very long test.ALL file by searching for the string, "N0B" (N - zero - B). We wish now to change these input data to the values listed near the top of Table 31. Note that nothing else in the test.ALL file need be changed. We are simply exploring a different range of circumferential wave numbers for buckling of exactly the same model as that used for the prediction of general buckling modes of the type shown in Fig. 27. In the new input we want to obtain 3 eigenvalues for each circumferential wave number. Why is that? Because the first two eigenvalues might well correspond to edge buckling, not to real local buckling of a complete (360-degree) cylindrical shell. (See the small inserts in the two figures in Appendix 1 for samples of edge buckling.) We execute BIGBOSORALL again:]

```
bigbosorall      (execute BIGBOSOR4)

bosorplot        (execute BOSORPLOT repeatedly until you see everthing
                  you want to see.)
```

[Inspect the test.OUT file. Search for the string, "EIGENVALUE(" and find the output similar to that listed in the middle of Table 31. Figures 34, 35, and 36 show the critical first, second, and third eigenvalues. The first two eigenvalues correspond to edge buckling, which we are not interested in. The third eigenvalue represents true local buckling.]

==

end of runstream in "trusscomp"/"test"

Table 9 trusscomp.INP (14 pages)

```
      5  $ starting prompt index in the file trusscomp.PRO
      5  $ increment for prompt index
      0  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
 Please read the report:
 y          $ Are there more lines in the "help" paragraph?

 y          $ Are there more lines in the "help" paragraph?
 "Use of GENOPT and BIGBOSOR4 to obtain optimum designs
 y          $ Are there more lines in the "help" paragraph?
 of a cylindrical shell with a composite truss-core
 y          $ Are there more lines in the "help" paragraph?
 sandwich wall", David Bushnell, June 20, 2009
 y          $ Are there more lines in the "help" paragraph?

 y          $ Are there more lines in the "help" paragraph?
 This GENOPT case is for a cylindrical shell the wall of
 y          $ Are there more lines in the "help" paragraph?
 which is a truss-core sandwich. Each segment of a truss-
 y          $ Are there more lines in the "help" paragraph?
 core sandwich module is a composite laminate. Hence, the
 y          $ Are there more lines in the "help" paragraph?
 name, "trusscomp". A truss-core module here consists of
 y          $ Are there more lines in the "help" paragraph?
 22 shell segments of the type used in BIGBOSOR4. General
 y          $ Are there more lines in the "help" paragraph?
 and local buckling load factors are determined from
 y          $ Are there more lines in the "help" paragraph?
 BIGBOSOR4. The shell is modeled as a part of a "huge
 y          $ Are there more lines in the "help" paragraph?
 torus". A membrane prebuckled state is assumed to exist.
 y          $ Are there more lines in the "help" paragraph?
 The axial resultant in each module segment is assumed to
 y          $ Are there more lines in the "help" paragraph?
 be proportional to the axial stiffness of that segment.
 y          $ Are there more lines in the "help" paragraph?
 Local buckling is determined from a single module.
 y          $ Are there more lines in the "help" paragraph?
 General buckling is determined from a number of analogous
 y          $ Are there more lines in the "help" paragraph?
 simpler modules strung together. The stress constraints
 y          $ Are there more lines in the "help" paragraph?
 are computed from a much simplified version of the code
 y          $ Are there more lines in the "help" paragraph?
 in SUBROUTINE STRTHX of BIGBOSOR4, simplified because
 y          $ Are there more lines in the "help" paragraph?
 the prebuckled state is a membrane state. The stresses
 y          $ Are there more lines in the "help" paragraph?
 are computed in each layer in each truss-core sandwich,
 y          $ Are there more lines in the "help" paragraph?
 segment.
 y          $ Are there more lines in the "help" paragraph?

 y          $ Are there more lines in the "help" paragraph?
 Occasionally an optimization run with SUPEROPT may
 y          $ Are there more lines in the "help" paragraph?
 abort early. If this happens, inspect the end of the
 y          $ Are there more lines in the "help" paragraph?
 xxx.OUT file to see if there are any error messages. You
 y          $ Are there more lines in the "help" paragraph?
 may have to modify your inequality constraints in the
 y          $ Are there more lines in the "help" paragraph?
 xxx*DEC file (input to "DECIDE") in order to avoid
 y          $ Are there more lines in the "help" paragraph?
 an early and unexpected termination of the SUPEROPT run.
 y          $ Are there more lines in the "help" paragraph?
 Most of the early terminations of the SUPEROPT run can
 y          $ Are there more lines in the "help" paragraph?
 be avoided by increasing the lower bound of the thickness
 y          $ Are there more lines in the "help" paragraph?
 of the truss-core, that is, increasing the lower bound
 y          $ Are there more lines in the "help" paragraph?
 of the variable called "HEIGHT".
 n          $ Are there more lines in the "help" paragraph?
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt

 LENGTH     $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
```

Input for GENTEXT

Note: GENTEXT destroys old versions of behavior.new struct.new

Therefore, make sure you save old versions of behavior.new & struct.new that you want to have for future use.

```
n         $ Is the variable  LENGTH  an array?
length of the cylindrical shell
y         $ Do you want to include a "help" paragraph?
NOTE: the cylindrical shell is assumed to be simply
y         $ Any more lines in the "help" paragraph?
supported along its two curved ends.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $10
     0  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
Next you will be asked to provide the fraction
y         $ Are there more lines in the "help" paragraph?
of the shell axial length to be used in the
y         $ Are there more lines in the "help" paragraph?
local buckling analysis. If the shell is
y         $ Are there more lines in the "help" paragraph?
subjected to uniform end shortening (axial
y         $ Are there more lines in the "help" paragraph?
compression), then this fraction of the total
y         $ Are there more lines in the "help" paragraph?
shell length can be estimated by assuming that
y         $ Are there more lines in the "help" paragraph?
the critical buckling mode will have approximately
y         $ Are there more lines in the "help" paragraph?
square buckles. The portion of the truss-core
y         $ Are there more lines in the "help" paragraph?
sandwich most likely to buckle may well have the
y         $ Are there more lines in the "help" paragraph?
longest unsupported width. Suppose the total
y         $ Are there more lines in the "help" paragraph?
length of the cylindrical shell is 100 inches.
y         $ Are there more lines in the "help" paragraph?
Suppose the width of the segment of the truss-core
y         $ Are there more lines in the "help" paragraph?
sandwich most likely to buckle locally is 2 inches.
y         $ Are there more lines in the "help" paragraph?
Then, for square buckles, the critical local buckling
y         $ Are there more lines in the "help" paragraph?
mode will have approximately 50 half-waves along
y         $ Are there more lines in the "help" paragraph?
the total axial length of the cylindrical shell.
y         $ Are there more lines in the "help" paragraph?
In order to save computer time you don't want to
y         $ Are there more lines in the "help" paragraph?
have to search over a large range of axial half-waves.
y         $ Are there more lines in the "help" paragraph?
Therefore, in this example, choose a fraction, FACLEN,
y         $ Are there more lines in the "help" paragraph?
of axial length equal to about 1/10th (FACLEN=0.1). Then
y         $ Are there more lines in the "help" paragraph?
the critical local buckling mode will have about 5 axial
y         $ Are there more lines in the "help" paragraph?
half-waves over the length, 0.1 x LENGTH = 10 inches,
y         $ Are there more lines in the "help" paragraph?
and you can establish the range of axial half-waves
y         $ Are there more lines in the "help" paragraph?
to be covered to be from 1 to 10 axial halfwaves.
y         $ Are there more lines in the "help" paragraph?
(MLOWL = 1, MHIGHL = 10). The probable critical
y         $ Are there more lines in the "help" paragraph?
buckling mode will probably be somewhere in the
y         $ Are there more lines in the "help" paragraph?
middle of this range.
n         $ Are there more lines in the "help" paragraph?
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
FACLEN    $ Name of a variable in the users program (defined below)
     2  $ Role of the variable in the users program
     2  $ type of variable:  1 =integer,  2 =floating point
n         $ Is the variable  FACLEN  an array?
fraction of LENGTH for local buckling
y         $ Do you want to include a "help" paragraph?
If the cylindrical shell is subjected to axial compression
y         $ Any more lines in the "help" paragraph?
set FACLEN such that if the local buckles are square the
y         $ Any more lines in the "help" paragraph?
critical local buckling mode will fall in the middle of
y         $ Any more lines in the "help" paragraph?
the range of axial wave numbers, MLOWL to MHIGHL, that
y         $ Any more lines in the "help" paragraph?
```

```
you will later establish during this interactive session.
n          $ Any more lines in the "help" paragraph?
y          $ Any more variables for role types  1  or  2   ?    $20
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
RADIUS     $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
n          $ Is the variable  RADIUS  an array?
radius of the cylindrical shell
y          $ Do you want to include a "help" paragraph?
This is the radius to the INNER face sheet of the
y          $ Any more lines in the "help" paragraph?
truss-core sandwich wall.
n          $ Any more lines in the "help" paragraph?
y          $ Any more variables for role types  1  or  2   ?    $25
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
PITCH      $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
n          $ Is the variable  PITCH  an array?
circumferential width of a single module
y          $ Do you want to include a "help" paragraph?
A single module consists essentially of one "upside-
y          $ Any more lines in the "help" paragraph?
down" trapezoid and its adjacent "rignt-side-up"
y          $ Any more lines in the "help" paragraph?
neighbor. There are 22 shell segments in the single
y          $ Any more lines in the "help" paragraph?
truss-core sandwich module. (See Figs. 1 and 2).
y          $ Any more lines in the "help" paragraph?
PITCH is called "b" in Fig. 1.
n          $ Any more lines in the "help" paragraph?
y          $ Any more variables for role types  1  or  2   ?    $30
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
BCROWN     $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
n          $ Is the variable  BCROWN  an array?
circumferential width of the trapezoid crown
y          $ Do you want to include a "help" paragraph?
BCROWN is the width of the narrow horizontal face of
y          $ Any more lines in the "help" paragraph?
the trapezoid. BCROWN is called "b2" in Fig. 1.
n          $ Any more lines in the "help" paragraph?
y          $ Any more variables for role types  1  or  2   ?    $35
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
HEIGHT     $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
n          $ Is the variable  HEIGHT  an array?
height of the truss-core sandwich
y          $ Do you want to include a "help" paragraph?
HEIGHT is measured from the middle surface of the
y          $ Any more lines in the "help" paragraph?
bottom face sheet to the middle surface of the top
y          $ Any more lines in the "help" paragraph?
face sheet. HEIGHT is called "h" in Fig. 1.
n          $ Any more lines in the "help" paragraph?
y          $ Any more variables for role types  1  or  2   ?    $40
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
RACUTE     $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
n          $ Is the variable  RACUTE  an array?
local radius from base to side of trapezoidal tool
y          $ Do you want to include a "help" paragraph?
RACUTE is the transition radius by means of which
y          $ Any more lines in the "help" paragraph?
the base surface of the trapezoidal tool is smoothly
y          $ Any more lines in the "help" paragraph?
faired into the sloping side walls of the trapezoidal
y          $ Any more lines in the "help" paragraph?
tool. The trapezoidal tool is the object around which
y          $ Any more lines in the "help" paragraph?
the layers of the core laminate are laid up. RACUTE
y          $ Any more lines in the "help" paragraph?
is the radius that spans the acute angle from trapezoid
y          $ Any more lines in the "help" paragraph?
base to trapezoid sloping sides. RACUTE is similar
y          $ Any more lines in the "help" paragraph?
to the dimension called "R1" in Fig. 1.
y          $ Any more lines in the "help" paragraph?
```

Table 9, p. 4 of 14

```
 (R1 = RACUTE + total thickness of shell segment 5).
 n         $ Any more lines in the "help" paragraph?
 y         $ Any more variables for role types  1  or  2   ?    $45
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
 ROBTUS    $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
 n         $ Is the variable  ROBTUS  an array?
 local radius from side to crown of trapezoidal tool
 y         $ Do you want to include a "help" paragraph?
 ROBTUS is the radius that fairs each side wall of the
 y         $ Any more lines in the "help" paragraph?
 trapezoidal tool into the crown of the trapezoidal
 y         $ Any more lines in the "help" paragraph?
 tool. The trapezoidal tool is the object around which
 y         $ Any more lines in the "help" paragraph?
 the core laminate is laid up. ROBTUS is the radius
 y         $ Any more lines in the "help" paragraph?
 that spans the obtuse angle from trapezoid crown to
 y         $ Any more lines in the "help" paragraph?
 trapezoid sloping sides. ROBTUS is similar
 y         $ Any more lines in the "help" paragraph?
 to the dimension called "R2" in Fig. 1.
 y         $ Any more lines in the "help" paragraph?
 (R2 = ROBTUS + total thickness of shell segment 2).
 n         $ Any more lines in the "help" paragraph?
 y         $ Any more variables for role types  1  or  2   ?    $50
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
 ENOODL    $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
 n         $ Is the variable  ENOODL  an array?
 axial modulus of a corner "noodle"
 y         $ Do you want to include a "help" paragraph?
 ENOODL is the axial modulus of the "noodle"
 y         $ Any more lines in the "help" paragraph?
 material. A "noodle" is the material that is
 y         $ Any more lines in the "help" paragraph?
 fills a "corner" between adjacent trapezoidal
 y         $ Any more lines in the "help" paragraph?
 core segments. Each truss-core sandwich module has
 y         $ Any more lines in the "help" paragraph?
 four "noodles", two at the bottom and two at the top
 y         $ Any more lines in the "help" paragraph?
 of the single module. The "noodles" fit into the
 y         $ Any more lines in the "help" paragraph?
 sort of triangular gaps adjacent to the slanted
 y         $ Any more lines in the "help" paragraph?
 core walls and the bottom and top face sheets. For
 y         $ Any more lines in the "help" paragraph?
 example, one of the "triangular" gaps shown in Fig. 1
 y         $ Any more lines in the "help" paragraph?
 is the small region enclosed by shell segments 2, 3,
 y         $ Any more lines in the "help" paragraph?
 4, and 5. ENOODL is the axial modulus of the
 y         $ Any more lines in the "help" paragraph?
 material that fills each of the four "triangular"
 y         $ Any more lines in the "help" paragraph?
 prismatic cavities in the single truss-core sandwich
 y         $ Any more lines in the "help" paragraph?
 module.
 n         $ Any more lines in the "help" paragraph?
 y         $ Any more variables for role types  1  or  2   ?    $55
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
 DNOODL    $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
 n         $ Is the variable  DNOODL  an array?
 weight density of the "noodle" material
 y         $ Do you want to include a "help" paragraph?
 ENOODL is the density analogous to that of
 y         $ Any more lines in the "help" paragraph?
 aluminum's weight density of 0.1 lb/in^3. Usually,
 y         $ Any more lines in the "help" paragraph?
 the "noodle" will be made from the same material
 y         $ Any more lines in the "help" paragraph?
 as one of the layers in the truss-core. If so,
 y         $ Any more lines in the "help" paragraph?
 just use the same value for DNOODL as you will
```

```
y         $ Any more lines in the "help" paragraph?
use later for the material density of one
y         $ Any more lines in the "help" paragraph?
of the layers in the truss-core laminate.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $60
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
EFOUND   $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
n         $ Is the variable  EFOUND  an array?
elastic foam "Winkler" foundation stiffness
y         $ Do you want to include a "help" paragraph?
EFOUND is used only for the LOCAL buckling analysis.
y         $ Any more lines in the "help" paragraph?
The volume inside the truss-core sandwich may
y         $ Any more lines in the "help" paragraph?
be filled with an elastic foam that is bonded
y         $ Any more lines in the "help" paragraph?
to the face sheets and sloping side walls of
y         $ Any more lines in the "help" paragraph?
the core. A Winkler foundation is assumed. A Winkler
y         $ Any more lines in the "help" paragraph?
foundation reacts only normal to the shell surface;
y         $ Any more lines in the "help" paragraph?
it is like an array of closely spaced springs
y         $ Any more lines in the "help" paragraph?
oriented normal to the shell surface. Units of the
y         $ Any more lines in the "help" paragraph?
foundation stiffness, K, are force per length
y         $ Any more lines in the "help" paragraph?
cubed (lb/in**3, for example). IMPORTANT NOTE:
y         $ Any more lines in the "help" paragraph?
The elastic foundation is assumed to be connected
y         $ Any more lines in the "help" paragraph?
to GROUND. Therefore, the elastic foam inside each
y         $ Any more lines in the "help" paragraph?
truss core module CANNOT TRANSLATE AND ROTATE AS
y         $ Any more lines in the "help" paragraph?
A RIGID BODY. Hence, the elastic foam is assumed
y         $ Any more lines in the "help" paragraph?
to be present ONLY IN THE LOCAL BUCKLING ANALYSIS.
y         $ Any more lines in the "help" paragraph?
Still, it is probably somewhat unconservative to
y         $ Any more lines in the "help" paragraph?
include a non-zero EFOUND in your model unless
y         $ Any more lines in the "help" paragraph?
you use a significantly conservative value for
y         $ Any more lines in the "help" paragraph?
EFNOOD, that is, an underestimate of what you think
y         $ Any more lines in the "help" paragraph?
EFNOOD might actually be. It is probably best first
y         $ Any more lines in the "help" paragraph?
to optimize with EFNOOD = 0.0, then re-optimize
y         $ Any more lines in the "help" paragraph?
with some small non-zero value of EFNOOD later.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $65
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
EFNOOD   $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
n         $ Is the variable  EFNOOD  an array?
elastic "noodle" Winkler foundation modulus
y         $ Do you want to include a "help" paragraph?
EFNOOD is used only for the LOCAL buckling analysis.
y         $ Any more lines in the "help" paragraph?
EFNOOD is the effective elastic foundation stiffness
y         $ Any more lines in the "help" paragraph?
that supports the segments of the single module
y         $ Any more lines in the "help" paragraph?
that form sort of triangular shaped gaps adjacent
y         $ Any more lines in the "help" paragraph?
to the slanted walls of the truss-core and the
y         $ Any more lines in the "help" paragraph?
bottom and top face sheets. EFNOOD is a "noodle"
y         $ Any more lines in the "help" paragraph?
property. A Winkler foundation is assumed. A Winkler
```

```
y         $ Any more lines in the "help" paragraph?
foundation reacts only normal to the shell surface;
y         $ Any more lines in the "help" paragraph?
it is like an array of closely spaced springs
y         $ Any more lines in the "help" paragraph?
oriented normal to the shell surface. Units of the
y         $ Any more lines in the "help" paragraph?
foundation stiffness, K, are force per length
y         $ Any more lines in the "help" paragraph?
cubed (lb/in**3, for example). Note that the same
y         $ Any more lines in the "help" paragraph?
comment applies to EFNOOD that occurs in the "help"
y         $ Any more lines in the "help" paragraph?
paragraph associated with the variable, EFOUND.
y         $ Any more lines in the "help" paragraph?
(NOTE: IF YOU WANT TO SEE ALL THE "HELP" PARAGRAPHS
y         $ Any more lines in the "help" paragraph?
 ASSOCIATED WITH ALL THE PROBLEM VARIABLES, INSPECT
y         $ Any more lines in the "help" paragraph?
 THE XXX.PRO FILE, IN WHICH "XXX" REPRESENTS THE
y         $ Any more lines in the "help" paragraph?
 GENERIC NAME OF THIS CASE).
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $70
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
EMOD1     $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  EMOD1  an array?
y         $ Do you want to establish new dimensions for EMOD1 ?
      1  $ Number of dimensions in the array,  EMOD1
material type
     10  $ Max. allowable number of rows NROWS in the array, EMOD1
elastic modulus in the fiber direction
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $80
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
EMOD2     $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  EMOD2  an array?
n         $ Do you want to establish new dimensions for EMOD2 ?
elastic modulus transverse to fibers
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $85
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
G12       $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  G12  an array?
n         $ Do you want to establish new dimensions for G12 ?
in-plane shear modulus
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $90
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
G13       $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  G13  an array?
n         $ Do you want to establish new dimensions for G13 ?
out-of-plane x-z shear modulus
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $95
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
G23       $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  G23  an array?
n         $ Do you want to establish new dimensions for G23 ?
out-of-plain y-z shear modulus
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $100
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
NU        $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  NU  an array?
n         $ Do you want to establish new dimensions for NU ?
```

```
minor (small) Poisson ratio
y         $ Do you want to include a "help" paragraph?
NU is the "NU21" Poisson ratio, governed by the
y         $ Any more lines in the "help" paragraph?
equation, E1*NU21 = E2*NU12. (Same input as for
y         $ Any more lines in the "help" paragraph?
BOSOR4 and PANDA2).
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $105
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
ALPHA1    $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  ALPHA1  an array?
n         $ Do you want to establish new dimensions for ALPHA1 ?
coef. of thermal expansion along the fibers
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $110
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
ALPHA2    $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  ALPHA2  an array?
n         $ Do you want to establish new dimensions for ALPHA2 ?
coef. of thermal expansion transverse to fibers
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $115
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
TEMCUR    $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  TEMCUR  an array?
n         $ Do you want to establish new dimensions for TEMCUR ?
curing temperature difference
y         $ Do you want to include a "help" paragraph?
TEMCUR is the difference between the temperature
y         $ Any more lines in the "help" paragraph?
at which the material "sets" during curing and the
y         $ Any more lines in the "help" paragraph?
service ("room") temperature. Use a POSITIVE number.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $120
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
DENSTY    $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  DENSTY  an array?
n         $ Do you want to establish new dimensions for DENSTY ?
weight density of material
y         $ Do you want to include a "help" paragraph?
For example, Aluminum = 0.1 lb/in^3
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $125
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
THICK     $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
y         $ Is the variable  THICK  an array?
y         $ Do you want to establish new dimensions for THICK ?
      1  $ Number of dimensions in the array,  THICK
layer type (thickness,layup angle,material type)
     30  $ Max. allowable number of rows NROWS in the array, THICK
layer type thickness
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $135
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
ANGLE     $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      2  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  ANGLE  an array?
n         $ Do you want to establish new dimensions for ANGLE ?
layer type layup angle
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $140
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
MATTYP    $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      1  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  MATTYP  an array?
```

Table 9, p.8 of 14

```
n         $ Do you want to establish new dimensions for MATTYP ?
layer type material type
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $145
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
LAYTYP    $ Name of a variable in the users program (defined below)
     2  $ Role of the variable in the users program
     1  $ type of variable:  1 =integer,  2 =floating point
y         $ Is the variable  LAYTYP  an array?
y         $ Do you want to establish new dimensions for LAYTYP ?
     2  $ Number of dimensions in the array,  LAYTYP
layer number
    90  $ Max. allowable number of rows NROWS in the array, LAYTYP
shell segment number
    50  $ Max. allowable number of columns NCOLS in the array, LAYTYP
layer type
y         $ Do you want to include a "help" paragraph?
LAYTYP(layer no., segment no.) identifies the type of
y         $ Any more lines in the "help" paragraph?
that particular layer in "segment no." of the single
y         $ Any more lines in the "help" paragraph?
module model. LAYTYP(layer no., segment no.) points
y         $ Any more lines in the "help" paragraph?
to the layer with a certain thickness, THICK, layup
y         $ Any more lines in the "help" paragraph?
angle, ANGLE, and material type, MATTYP.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $160
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
NLAYRF    $ Name of a variable in the users program (defined below)
     2  $ Role of the variable in the users program
     1  $ type of variable:  1 =integer,  2 =floating point
n         $ Is the variable  NLAYRF  an array?
number of layers in each face sheet
y         $ Do you want to include a "help" paragraph?
This is the number of layers, such that when
y         $ Any more lines in the "help" paragraph?
added to the number of layers wrapped around the
y         $ Any more lines in the "help" paragraph?
trapezoidal tool (NLAYRC, the next role 2 variable),
y         $ Any more lines in the "help" paragraph?
gives the total number of layers in the laminate
y         $ Any more lines in the "help" paragraph?
that forms the wall of the horizontal shell
y         $ Any more lines in the "help" paragraph?
segments that run along the base and crown of
y         $ Any more lines in the "help" paragraph?
each trapezoidal segment of the truss-core
y         $ Any more lines in the "help" paragraph?
sandwich structure.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $165
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
NLAYRC    $ Name of a variable in the users program (defined below)
     2  $ Role of the variable in the users program
     1  $ type of variable:  1 =integer,  2 =floating point
n         $ Is the variable  NLAYRC  an array?
number of layers around trapezoidal tool
y         $ Do you want to include a "help" paragraph?
NLAYRC is the number of layers wrapped around the trapezoidal
y         $ Any more lines in the "help" paragraph?
tool, that is, the number of layers in the truss core.
y         $ Any more lines in the "help" paragraph?
For the upper and lower face sheets NLAYRC is added to
y         $ Any more lines in the "help" paragraph?
the number of layers, NLAYRF, to form the total number
y         $ Any more lines in the "help" paragraph?
of layers in the horizontal shell segment walls.
y         $ Any more lines in the "help" paragraph?
The number of layers in the truss-core webs will be
y         $ Any more lines in the "help" paragraph?
2 x NLAYRC.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $170
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
INTEXT    $ Name of a variable in the users program (defined below)
     2  $ Role of the variable in the users program
     1  $ type of variable:  1 =integer,  2 =floating point
```

```
n         $ Is the variable  INTEXT  an array?
0=pressure on top face sheet; 1=bottom face sheet
y         $ Do you want to include a "help" paragraph?
INTEXT = 0 means pressure acts on the top face sheet.
y         $ Any more lines in the "help" paragraph?
INTEXT = 1 means pressure acts on the bottom face sheet.
y         $ Any more lines in the "help" paragraph?
In the model used here the circumferential resultant
y         $ Any more lines in the "help" paragraph?
in each face sheet is assumed to be (PRESS x RADIUS)/2.
y         $ Any more lines in the "help" paragraph?
in which PRESS is the applied pressure (negative for
y         $ Any more lines in the "help" paragraph?
external pressure), and RADIUS is the radius of the
y         $ Any more lines in the "help" paragraph?
cylindrical shell..
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $175
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
NSEGS     $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      1  $ type of variable:  1 =integer,  2 =floating point
n         $ Is the variable  NSEGS  an array?
number of segments in module
y         $ Do you want to include a "help" paragraph?
NSEGS must be 22 for the truss-core sandwich module.
y         $ Any more lines in the "help" paragraph?
At present NSEGS is set equal to 22 in SUBROUTINE BOSDEC.
y         $ Any more lines in the "help" paragraph?
Therefore, the value you provide here is overwritten.
y         $ Any more lines in the "help" paragraph?
You may feel that NSEGS should not be an input variable
y         $ Any more lines in the "help" paragraph?
because the user must always specify NSEGS = 22 for
y         $ Any more lines in the "help" paragraph?
now. However, perhaps this will change in the future.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $180
      0  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
You will next be asked to provide the number of modules
y         $ Are there more lines in the "help" paragraph?
to be used in the model for general buckling (NMODULG)
y         $ Are there more lines in the "help" paragraph?
and for local buckling (NMODULL). As of now NMODULG is
y         $ Are there more lines in the "help" paragraph?
computed in SUBROUTINE BOSDEC. Therefore, the value of
y         $ Are there more lines in the "help" paragraph?
NMODULG that you provide here will not be used. However,
y         $ Are there more lines in the "help" paragraph?
it is possible that in the future SUBROUTINE BOSDEC may
y         $ Are there more lines in the "help" paragraph?
be changed. Hence, NMODULG is retained as an input datum.
y         $ Are there more lines in the "help" paragraph?
Use a number for NMODULG such as NMODULG = 13 (NOTE: you
y         $ Are there more lines in the "help" paragraph?
are not allowed to use NMODULG > 13). For local buckling
y         $ Are there more lines in the "help" paragraph?
you should use NMODULL = 1 until you have a preliminary
y         $ Are there more lines in the "help" paragraph?
optimum design. Then, to refine the design, possibly
y         $ Are there more lines in the "help" paragraph?
optimize again with NMODULL = 10 or something like that.
y         $ Are there more lines in the "help" paragraph?
NOTE: The computer time increases significantly the
y         $ Are there more lines in the "help" paragraph?
higher the value of NMODULL.
n         $ Are there more lines in the "help" paragraph?
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
NMODULG   $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      1  $ type of variable:  1 =integer,  2 =floating point
n         $ Is the variable  NMODULG  an array?
number of modules for general buckling
y         $ Do you want to include a "help" paragraph?
Use a number such as NMODULG = 13 (NOTE: you are
y         $ Any more lines in the "help" paragraph?
not permitted to use NMODULG > 13). The computer
y         $ Any more lines in the "help" paragraph?
```

Table 9, p.10 of 14

```
time required for a SUPEROPT optimization increases
y          $ Any more lines in the "help" paragraph?
with increase in NMODULG. However, it is probably
y          $ Any more lines in the "help" paragraph?
best always to use the maximum value of NMODULG
y          $ Any more lines in the "help" paragraph?
that is permitted (NMODULG = 13) because you will
y          $ Any more lines in the "help" paragraph?
be less likely to obtain an unconservative design.
y          $ Any more lines in the "help" paragraph?
The value, NMODULG = 13, produces a BIGBOSOR4 model
y          $ Any more lines in the "help" paragraph?
with a total of 286 shell segments (286 = 22 x 13),
y          $ Any more lines in the "help" paragraph?
which is close to the maximum number of shell
y          $ Any more lines in the "help" paragraph?
segments (295) that BIGBOSOR4 can presently handle.
n          $ Any more lines in the "help" paragraph?
y          $ Any more variables for role types  1  or  2   ?    $190
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
NMODULL    $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      1  $ type of variable:  1 =integer,  2 =floating point
n          $ Is the variable  NMODULL  an array?
number of modules for local buckling
y          $ Do you want to include a "help" paragraph?
NMODULL is the number of modules, each of which
y          $ Any more lines in the "help" paragraph?
contains NSEGS shell segments. In this application
y          $ Any more lines in the "help" paragraph?
the best answer is usually 1  (that is, one module).
y          $ Any more lines in the "help" paragraph?
That is because the buckling (or stress) model
y          $ Any more lines in the "help" paragraph?
that consists of NSEGS shell segments is used
y          $ Any more lines in the "help" paragraph?
for obtaining LOCAL behavior, such as local
y          $ Any more lines in the "help" paragraph?
buckling. Only one module of NSEGS shell segments
y          $ Any more lines in the "help" paragraph?
is needed for the determination of LOCAL
y          $ Any more lines in the "help" paragraph?
behavior. However, there seems to be some dependence
y          $ Any more lines in the "help" paragraph?
of the local buckling load on the number of modules,
y          $ Any more lines in the "help" paragraph?
NMODULL, used in the model for local buckling. Hence,
y          $ Any more lines in the "help" paragraph?
after you have optimized with the use of NMODULL = 1
y          $ Any more lines in the "help" paragraph?
you may want to check the optimum design with use of
y          $ Any more lines in the "help" paragraph?
NMODULL = 10 or something like that.
n          $ Any more lines in the "help" paragraph?
y          $ Any more variables for role types  1  or  2   ?    $195
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
MLOWG      $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      1  $ type of variable:  1 =integer,  2 =floating point
n          $ Is the variable  MLOWG  an array?
low end of the M-range for general buckling
y          $ Do you want to include a "help" paragraph?
MLOWG should be 1 (an integer). Later it will be
y          $ Any more lines in the "help" paragraph?
multiplied by 100 because of the "huge torus" model.
y          $ Any more lines in the "help" paragraph?
The "huge torus" model is described in Part 12.2.2
y          $ Any more lines in the "help" paragraph?
of the paper, "Optimization of an axially compressed
y          $ Any more lines in the "help" paragraph?
ring and stringer stiffened cylindrical shell with
y          $ Any more lines in the "help" paragraph?
a general buckling modal imperfection", AIAA Paper
y          $ Any more lines in the "help" paragraph?
2007-2216, 48th AIAA Structures Meeting, Honolulu,
y          $ Any more lines in the "help" paragraph?
Hawaii, April 2007.
n          $ Any more lines in the "help" paragraph?
```

```
y         $ Any more variables for role types  1  or  2   ?    $200
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
MHIGHG     $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      1  $ type of variable:  1 =integer,  2 =floating point
n         $ Is the variable  MHIGHG  an array?
high end of the M-range for general buckling
y         $ Do you want to include a "help" paragraph?
Perhaps one should give a value 10 (or something
y         $ Any more lines in the "help" paragraph?
like that). It depends on how many axial halfwaves
y         $ Any more lines in the "help" paragraph?
you think will correspond to the critical general
y         $ Any more lines in the "help" paragraph?
buckling mode. That number depends on the axial
y         $ Any more lines in the "help" paragraph?
bending stiffness of the cylindrical shell wall,
y         $ Any more lines in the "help" paragraph?
the length of the cylindrical shell, and the
y         $ Any more lines in the "help" paragraph?
radius of the cylindrical shell. Use at least
y         $ Any more lines in the "help" paragraph?
double the value of what you think is the critical
y         $ Any more lines in the "help" paragraph?
value, that is, the value that corresponds to the
y         $ Any more lines in the "help" paragraph?
lowest general buckling load. You may have to
y         $ Any more lines in the "help" paragraph?
adjust the value of MHIGHG as your project proceeds.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $205
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
MLOWL      $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      1  $ type of variable:  1 =integer,  2 =floating point
n         $ Is the variable  MLOWL  an array?
low end of the range of local buckling waves
y         $ Do you want to include a "help" paragraph?
You should use MLOWL = 1 because the axial length of
y         $ Any more lines in the "help" paragraph?
the portion of the cylindrical shell to be used in the
y         $ Any more lines in the "help" paragraph?
LOCAL buckling analysis is equal to LENGTH x FACLEN,
y         $ Any more lines in the "help" paragraph?
in which LENGTH is the actual axial length of the
y         $ Any more lines in the "help" paragraph?
cylindrical shell, and FACLEN is the fraction of
y         $ Any more lines in the "help" paragraph?
LENGTH to be included in the local buckling model.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $210
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
MHIGHL     $ Name of a variable in the users program (defined below)
      2  $ Role of the variable in the users program
      1  $ type of variable:  1 =integer,  2 =floating point
n         $ Is the variable  MHIGHL  an array?
high end of the M-range for local buckling
y         $ Do you want to include a "help" paragraph?
We're referring here to the range of axial halfwaves
y         $ Any more lines in the "help" paragraph?
to be used in the search for the critical LOCAL
y         $ Any more lines in the "help" paragraph?
buckling load of the single truss-core sandwich
y         $ Any more lines in the "help" paragraph?
module. Use a number such as 10 and adjust it later
y         $ Any more lines in the "help" paragraph?
if necessary.
n         $ Any more lines in the "help" paragraph?
n         $ Any more variables for role types  1  or  2   ?    $
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
PX         $ Name of a variable in the users program (defined below)
      3  $ Role of the variable in the users program
total axial load (2 x pi x r x resultant)
y         $ Do you want to include a "help" paragraph?
PX = 2 x pi x r x axial resultant. Units = lb,
y         $ Any more lines in the "help" paragraph?
for example. PX is negative for axial compression.
y         $ Any more lines in the "help" paragraph?
```

Table 9, p.11 of 14

```
The axial resultant in each shell segment is computed
y         $ Any more lines in the "help" paragraph?
as if the shell were subject to uniform end shortening
y         $ Any more lines in the "help" paragraph?
and the prebuckled state is a membrane state (no
y         $ Any more lines in the "help" paragraph?
imperfections, no bending, no end "boundary layers").
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role type  3 ?                 $225
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
PRESS     $ Name of a variable in the users program (defined below)
     3  $ Role of the variable in the users program
pressure (negative for external pressure)
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role type  3 ?                 $230
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
PX0       $ Name of a variable in the users program (defined below)
     3  $ Role of the variable in the users program
total "Load Set B" load
y         $ Do you want to include a "help" paragraph?
PX0 = 2 x pi x r x Nx0, in which Nx0 is an
y         $ Any more lines in the "help" paragraph?
axial stress resultant. PX0 = positive for axial
y         $ Any more lines in the "help" paragraph?
tension. Load Set B is the load set that is NOT to be
y         $ Any more lines in the "help" paragraph?
multiplied by the buckling eigenvalue.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role type  3 ?                 $235
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
PRESS0    $ Name of a variable in the users program (defined below)
     3  $ Role of the variable in the users program
"Load Set B" pressure (external=negative)
y         $ Do you want to include a "help" paragraph?
PRESS0 is positive for internal pressure. Load Set B
y         $ Any more lines in the "help" paragraph?
is the load set that is NOT to be multiplied by
y         $ Any more lines in the "help" paragraph?
the buckling eigenvalue.
n         $ Any more lines in the "help" paragraph?
n         $ Any more variables for role type  3 ?                 $
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
LOCBUK    $ Name of a variable in the users program (defined below)
     4  $ Role of the variable in the users program
n         $ Do you want to reset the number of columns in LOCBUK ?
local buckling load factor
n         $ Do you want to include a "help" paragraph?
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
LOCBUKA   $ Name of a variable in the users program (defined below)
     5  $ Role of the variable in the users program
allowable for local buckling load factor
y         $ Do you want to include a "help" paragraph?
Usually, you supply 1.0 for LOCBUKA because LOCBUK is
y         $ Any more lines in the "help" paragraph?
a buckling load FACTOR, that is, a quantity that is
y         $ Any more lines in the "help" paragraph?
to be multiplied by the design loads, PX and PRESS,
y         $ Any more lines in the "help" paragraph?
in order to obtain the buckling load.
n         $ Any more lines in the "help" paragraph?
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
LOCBUKF   $ Name of a variable in the users program (defined below)
     6  $ Role of the variable in the users program
factor of safety for local buckling
y         $ Do you want to include a "help" paragraph?
In this "huge torus" model of the shell there cannot
y         $ Any more lines in the "help" paragraph?
exist any specified initial imperfection. Therefore,
y         $ Any more lines in the "help" paragraph?
you should probably use a factor of safety of 1.5
y         $ Any more lines in the "help" paragraph?
or something like that.
n         $ Any more lines in the "help" paragraph?
     2  $ Indicator (1 or 2 or 3) for type of constraint
y         $ Any more variables for role type  4 ?                 $255
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
GENBUK    $ Name of a variable in the users program (defined below)
     4  $ Role of the variable in the users program
```

```
n         $ Do you want to reset the number of columns in GENBUK ?
general buckling load factor
n         $ Do you want to include a "help" paragraph?
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
GENBUKA   $ Name of a variable in the users program (defined below)
     5  $ Role of the variable in the users program
allowable for general buckling load factor
y         $ Do you want to include a "help" paragraph?
Usually, you supply 1.0 for GENBUKA because GENBUK is
y         $ Any more lines in the "help" paragraph?
a buckling load FACTOR, that is, a quantity that is
y         $ Any more lines in the "help" paragraph?
to be multiplied by the design loads, PX and PRESS,
y         $ Any more lines in the "help" paragraph?
in order to obtain the buckling load.
n         $ Any more lines in the "help" paragraph?
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
GENBUKF   $ Name of a variable in the users program (defined below)
     6  $ Role of the variable in the users program
general buckling factor of safety
y         $ Do you want to include a "help" paragraph?
In this "huge torus" model of the shell there cannot
y         $ Any more lines in the "help" paragraph?
exist any specified initial imperfection. Therefore,
y         $ Any more lines in the "help" paragraph?
you should probably use a factor of safety of 2.0
y         $ Any more lines in the "help" paragraph?
or something like that.
n         $ Any more lines in the "help" paragraph?
     2  $ Indicator (1 or 2 or 3) for type of constraint
y         $ Any more variables for role type  4 ?          $270
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
STRM1     $ Name of a variable in the users program (defined below)
     4  $ Role of the variable in the users program
y         $ Do you want to reset the number of columns in STRM1 ?
     2  $ Number of dimensions in the array,  STRM1
number of stress constraints
     6  $ Max. allowable number of columns NCOLS in the array, STRM1
stress component in material 1
y         $ Do you want to include a "help" paragraph?
For a composite laminate there are 6 stress components
y         $ Any more lines in the "help" paragraph?
for which stress constraints are generated:
y         $ Any more lines in the "help" paragraph?
1. maximum tensile stess in the fiber direction
y         $ Any more lines in the "help" paragraph?
2. maximum compressive stress in the fiber direction
y         $ Any more lines in the "help" paragraph?
3. maximum tensile stress transverse to the fibers
y         $ Any more lines in the "help" paragraph?
4. maximum compressive stress transverse to the fibers
y         $ Any more lines in the "help" paragraph?
5. maximum in-plane shear stress
y         $ Any more lines in the "help" paragraph?
6. maximum effective (VonMises) stress for an
y         $ Any more lines in the "help" paragraph?
isotropic layer.
n         $ Any more lines in the "help" paragraph?
     1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
STRM1A    $ Name of a variable in the users program (defined below)
     5  $ Role of the variable in the users program
allowable stress in material 1
y         $ Do you want to include a "help" paragraph?
For a composite laminate there are 6 stress components
y         $ Any more lines in the "help" paragraph?
for which stress constraints are generated:
y         $ Any more lines in the "help" paragraph?
1. maximum tensile stess in the fiber direction:
y         $ Any more lines in the "help" paragraph?
  STRM1A(i,1), in which "i" is the load set number
y         $ Any more lines in the "help" paragraph?
2. maximum compressive stress in the fiber direction:
y         $ Any more lines in the "help" paragraph?
  STRM1A(i,2), in which "i" is the load set number
y         $ Any more lines in the "help" paragraph?
3. maximum tensile stress transverse to the fibers:
y         $ Any more lines in the "help" paragraph?
  STRM1A(i,3), in which "i" is the load set number
```

Table 9, p. 13 of 14

```
y         $ Any more lines in the "help" paragraph?
4. maximum compressive stress transverse to the fibers:
y         $ Any more lines in the "help" paragraph?
   STRM1A(i,4), in which "i" is the load set number
y         $ Any more lines in the "help" paragraph?
5. maximum in-plane shear stress:
y         $ Any more lines in the "help" paragraph?
   STRM1A(i,5), in which "i" is the load set number
y         $ Any more lines in the "help" paragraph?
6. maximum effective (VonMises) stress for an
y         $ Any more lines in the "help" paragraph?
isotropic layer:
y         $ Any more lines in the "help" paragraph?
   STRM1A(i,6), in which "i" is the load set number
n         $ Any more lines in the "help" paragraph?
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
STRM1F    $ Name of a variable in the users program (defined below)
      6  $ Role of the variable in the users program
factor of safety for stress in material 1
y         $ Do you want to include a "help" paragraph?
In this "huge torus" model of the shell there cannot
y         $ Any more lines in the "help" paragraph?
exist any specified initial imperfection. Therefore,
y         $ Any more lines in the "help" paragraph?
you should probably use a factor of safety of 1.5
y         $ Any more lines in the "help" paragraph?
or something like that.
n         $ Any more lines in the "help" paragraph?
      3  $ Indicator (1 or 2 or 3) for type of constraint
y         $ Any more variables for role type  4 ?              $290
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
STRM2     $ Name of a variable in the users program (defined below)
      4  $ Role of the variable in the users program
n         $ Do you want to reset the number of columns in STRM2 ?
stress component in material 2
n         $ Do you want to include a "help" paragraph?
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
STRM2A    $ Name of a variable in the users program (defined below)
      5  $ Role of the variable in the users program
allowable for stress in material 2
n         $ Do you want to include a "help" paragraph?
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
STRM2F    $ Name of a variable in the users program (defined below)
      6  $ Role of the variable in the users program
factor of safety for stress in material 2
n         $ Do you want to include a "help" paragraph?
      3  $ Indicator (1 or 2 or 3) for type of constraint
n         $ Any more variables for role type  4 ?              $
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
WEIGHT    $ Name of a variable in the users program (defined below)
      7  $ Role of the variable in the users program
weight/area of the truss-core sandwich wall
y         $ Do you want to include a "help" paragraph?
WEIGHT = weight per surface area of the truss-core
y         $ Any more lines in the "help" paragraph?
sandwich shell wall.
n         $ Any more lines in the "help" paragraph?
```

Input for GENTEXT

Table 10 part of the trusscomp.DEF file

```
C==========================================================
C  ARRAY  NUMBER OF             PROMPT
C    ?    (ROWS,COLS)  ROLE  NUMBER   NAME                DEFINITION OF VARIABLE
C                            (trusscomp.PRO)
C==========================================================
C    n    (   0,    0)    2      10    LENGTH    = length of the cylindrical shell
C    n    (   0,    0)    2      20    FACLEN    = fraction of LENGTH for local buckling
C    n    (   0,    0)    2      25    RADIUS    = radius of the cylindrical shell
C    n    (   0,    0)    1      30    PITCH     = circumferential width of a single module
C    n    (   0,    0)    1      35    BCROWN    = circumferential width of the trapezoid crown
C    n    (   0,    0)    1      40    HEIGHT    = height of the truss-core sandwich
C    n    (   0,    0)    1      45    RACUTE    = local radius from base to side of trapezoidal too»
l
C    n    (   0,    0)    1      50    ROBTUS    = local radius from side to crown of trapezoidal to»
ol
C    n    (   0,    0)    2      55    ENOODL    = axial modulus of a corner "noodle"
C    n    (   0,    0)    2      60    DNOODL    = weight density of the "noodle" material
C    n    (   0,    0)    2      65    EFOUND    = elastic foam foundation stiffness
C    n    (   0,    0)    2      70    EFNOOD    = elastic "noodle" Winkler foundation modulus
C    n    (   0,    0)    2      75    IEMOD1    = material type in EMOD1(IEMOD1)
C    y    (  10,    0)    2      80    EMOD1     = elastic modulus in the fiber direction
C    y    (  10,    0)    2      85    EMOD2     = elastic modulus transverse to fibers
C    y    (  10,    0)    2      90    G12       = in-plane shear modulus
C    y    (  10,    0)    2      95    G13       = out-of-plane x-z shear modulus
C    y    (  10,    0)    2     100    G23       = out-of-plain y-z shear modulus
C    y    (  10,    0)    2     105    NU        = minor (small) Poisson ratio
C    y    (  10,    0)    2     110    ALPHA1    = coef. of thermal expansion along the fibers
C    y    (  10,    0)    2     115    ALPHA2    = coef. of thermal expansion transverse to fibers
C    y    (  10,    0)    2     120    TEMCUR    = curing temperature difference
C    y    (  10,    0)    2     125    DENSTY    = weight density of material
C    n    (   0,    0)    2     130    ITHICK    = layer type (thickness,layup angle,material type) »
in THICK(ITHICK)
C    y    (  30,    0)    1     135    THICK     = layer type thickness
C    y    (  30,    0)    2     140    ANGLE     = layer type layup angle
C    y    (  30,    0)    2     145    MATTYP    = layer type material type
C    n    (   0,    0)    2     150    JLAYTYP   = shell segment number in LAYTYP(ILAYTYP,JLAYTYP)
C    n    (   0,    0)    2     155    ILAYTYP   = layer number in LAYTYP(ILAYTYP,JLAYTYP)
C    y    (  90,   50)    2     160    LAYTYP    = layer type
C    n    (   0,    0)    2     165    NLAYRF    = number of layers in each face sheet
C    n    (   0,    0)    2     170    NLAYRC    = number of layers around trapezoidal tool
C    n    (   0,    0)    2     175    INTEXT    = 0 =pressure on top face sheet; 1 = bottom face sh»
eet
C    n    (   0,    0)    2     180    NSEGS     = number of segments in module
C    n    (   0,    0)    2     190    NMODULG   = number of modules for general buckling
C    n    (   0,    0)    2     195    NMODULL   = number of modules for local buckling
C    n    (   0,    0)    2     200    MLOWG     = low end of the M-range for general buckling
C    n    (   0,    0)    2     205    MHIGHG    = high end of the M-range for general buckling
C    n    (   0,    0)    2     210    MLOWL     = low end of the range of local buckling waves
C    n    (   0,    0)    2     215    MHIGHL    = high end of the M-range for local buckling
C    n    (   0,    0)    2     220    NCASES    = Number of load cases (number of environments)  in»
 PX(NCASES)
C    y    (  20,    0)    3     225    PX        = total axial load (2 x pi x r x resultant)
C    y    (  20,    0)    3     230    PRESS     = pressure (negative for external pressure)
C    y    (  20,    0)    3     235    PX0       = total "Load Set B" load
C    y    (  20,    0)    3     240    PRESS0    = "Load Set B" pressure (external=negative)
C    y    (  20,    0)    4     245    LOCBUK    = local buckling load factor
C    y    (  20,    0)    5     250    LOCBUKA   = allowable for local buckling load factor
C    y    (  20,    0)    6     255    LOCBUKF   = factor of safety for local buckling
C    y    (  20,    0)    4     260    GENBUK    = general buckling load factor
C    y    (  20,    0)    5     265    GENBUKA   = allowable for general buckling load factor
C    y    (  20,    0)    6     270    GENBUKF   = general buckling factor of safety
C    n    (   0,    0)    2     275    JSTRM1    = number of stress constraints in STRM1(NCASES,JSTR»
M1)
C    y    (  20,    6)    4     280    STRM1     = stress component in material 1
C    y    (  20,    6)    5     285    STRM1A    = allowable stress in material 1
C    y    (  20,    6)    6     290    STRM1F    = factor of safety for stress in matl 1
C    y    (  20,    6)    4     295    STRM2     = stress component in material 2
C    y    (  20,    6)    5     300    STRM2A    = allowable for stress in material 2
C    y    (  20,    6)    6     305    STRM2F    = factor of safety for stress in material 2
C    n    (   0,    0)    7     310    WEIGHT    = weight/area of the truss-core sandwich wall
C
C==========================================================
```

This list (& the entire trusscomp.DEF file) are created automatically by GENTEXT.

Table 11 trusscomp.PRO file (6 pages)

5.0

Please read the report:

"Use of GENOPT and BIGBOSOR4 to obtain optimum designs of a cylindrical shell with a composite truss-core sandwich wall", David Bushnell, June 20, 2009

This GENOPT case is for a cylindrical shell the wall of which is a truss-core sandwich. Each segment of a truss-core sandwich module is a composite laminate. Hence, the name, "trusscomp". A truss-core module here consists of 22 shell segments of the type used in BIGBOSOR4. General and local buckling load factors are determined from BIGBOSOR4. The shell is modeled as a part of a "huge torus". A membrane prebuckled state is assumed to exist. The axial resultant in each module segment is assumed to be proportional to the axial stiffness of that segment. Local buckling is determined from a single module. General buckling is determined from a number of analogous simpler modules strung together. The stress constraints are computed from a much simplified version of the code in SUBROUTINE STRTHX of BIGBOSOR4, simplified because the prebuckled state is a membrane state. The stresses are computed in each layer in each truss-core sandwich, segment.

Occasionally an optimization run with SUPEROPT may abort early. If this happens, inspect the end of the xxx.OUT file to see if there are any error messages. You may have to modify your inequality constraints in the xxx*DEC file (input to "DECIDE") in order to avoid an early and unexpected termination of the SUPEROPT run. Most of the early terminations of the SUPEROPT run can be avoided by increasing the lower bound of the thickness of the truss-core, that is, increasing the lower bound of the variable called "HEIGHT".

10.1 length of the cylindrical shell: LENGTH

10.2

NOTE: the cylindrical shell is assumed to be simply supported along its two curved ends.

15.0

Next you will be asked to provide the fraction of the shell axial length to be used in the local buckling analysis. If the shell is subjected to uniform end shortening (axial compression), then this fraction of the total shell length can be estimated by assuming that the critical buckling mode will have approximately square buckles. The portion of the truss-core sandwich most likely to buckle may well have the longest unsupported width. Suppose the total length of the cylindrical shell is 100 inches. Suppose the width of the segment of the truss-core sandwich most likely to buckle locally is 2 inches. Then, for square buckles, the critical local buckling mode will have approximately 50 half-waves along the total axial length of the cylindrical shell. In order to save computer time you don't want to have to search over a large range of axial half-waves. Therefore, in this example, choose a fraction, FACLEN, of axial length equal to about 1/10th (FACLEN=0.1). Then the critical local buckling mode will have about 5 axial half-waves over the length, 0.1 x LENGTH = 10 inches, and you can establish the range of axial half-waves to be covered to be from 1 to 10 axial halfwaves. (MLOWL = 1, MHIGHL = 10). The probable critical buckling mode will probably be somewhere in the middle of this range.

20.1 fraction of LENGTH for local buckling: FACLEN

20.2

If the cylindrical shell is subjected to axial compression set FACLEN such that if the local buckles are square the critical local buckling mode will fall in the middle of

This is the prompting file used during the execution of BEGIN. It is produced automatically by GENTEXT. The text in it is created by the GENOPT user, but this text is arranged in this format by GENTEXT.

the range of axial wave numbers, MLOWL to MHIGHL, that
you will later establish during this interactive session.

25.1 radius of the cylindrical shell: RADIUS
25.2
This is the radius to the INNER face sheet of the truss-core sandwich wall.

30.1 circumferential width of a single module: PITCH
30.2
A single module consists essentially of one "upside-down" trapezoid and its adjacent "rignt-side-up" neighbor. There are 22 shell segments in the single truss-core sandwich module. (See Figs. 1 and 2). PITCH is called "b" in Fig. 1.

35.1 circumferential width of the trapezoid crown: BCROWN
35.2
BCROWN is the width of the narrow horizontal face of the trapezoid. BCROWN is called "b2" in Fig. 1.

40.1 height of the truss-core sandwich: HEIGHT
40.2
HEIGHT is measured from the middle surface of the bottom face sheet to the middle surface of the top face sheet. HEIGHT is called "h" in Fig. 1.

45.1 local radius from base to side of trapezoidal tool: RACUTE
45.2
RACUTE is the transition radius by means of which the base surface of the trapezoidal tool is smoothly faired into the sloping side walls of the trapezoidal tool. The trapezoidal tool is the object around which the layers of the core laminate are laid up. RACUTE is the radius that spans the acute angle from trapezoid base to trapezoid sloping sides. RACUTE is similar to the dimension called "R1" in Fig. 1.
(R1 = RACUTE + total thickness of shell segment 5).

50.1 local radius from side to crown of trapezoidal tool: ROBTUS
50.2
ROBTUS is the radius that fairs each side wall of the trapezoidal tool into the crown of the trapezoidal tool. The trapezoidal tool is the object around which the core laminate is laid up. ROBTUS is the radius that spans the obtuse angle from trapezoid crown to trapezoid sloping sides. ROBTUS is similar to the dimension called "R2" in Fig. 1.
(R2 = ROBTUS + total thickness of shell segment 2).

55.1 axial modulus of a corner "noodle": ENOODL
55.2
ENOODL is the axial modulus of the "noodle" material. A "noodle" is the material that is fills a "corner" between adjacent trapezoidal core segments. Each truss-core sandwich module has four "noodles", two at the bottom and two at the top of the single module. The "noodles" fit into the sort of triangular gaps adjacent to the slanted core walls and the bottom and top face sheets. For example, one of the "triangular" gaps shown in Fig. 1 is the small region enclosed by shell segments 2, 3, 4, and 5. ENOODL is the axial modulus of the material that fills each of the four "triangular" prismatic cavities in the single truss-core sandwich module.

60.1 weight density of the "noodle" material: DNOODL
60.2
ENOODL is the density analogous to that of aluminum's weight density of 0.1 lb/in^3. Usually, the "noodle" will be made from the same material as one of the layers in the truss-core. If so, just use the same value for DNOODL as you will use later for the material density of one of the layers in the truss-core laminate.

65.1 elastic foam "Winkler" foundation stiffness: EFOUND

65.2
EFOUND is used only for the LOCAL buckling analysis.
The volume inside the truss-core sandwich may
be filled with an elastic foam that is bonded
to the face sheets and sloping side walls of
the core. A Winkler foundation is assumed. A Winkler
foundation reacts only normal to the shell surface;
it is like an array of closely spaced springs
oriented normal to the shell surface. Units of the
foundation stiffness, K, are force per length
cubed (lb/in**3, for example). IMPORTANT NOTE:
The elastic foundation is assumed to be connected
to GROUND. Therefore, the elastic foam inside each
truss core module CANNOT TRANSLATE AND ROTATE AS
A RIGID BODY. Hence, the elastic foam is assumed
to be present ONLY IN THE LOCAL BUCKLING ANALYSIS.
Still, it is probably somewhat unconservative to
include a non-zero EFOUND in your model unless
you use a significantly conservative value for
EFNOOD, that is, an underestimate of what you think
EFNOOD might actually be. It is probably best first
to optimize with EFNOOD = 0.0, then re-optimize
with some small non-zero value of EFNOOD later.

70.1 elastic "noodle" Winkler foundation modulus: EFNOOD
70.2
EFNOOD is used only for the LOCAL buckling analysis.
EFNOOD is the effective elastic foundation stiffness
that supports the segments of the single module
that form sort of triangular shaped gaps adjacent
to the slanted walls of the truss-core and the
bottom and top face sheets. EFNOOD is a "noodle"
property. A Winkler foundation is assumed. A Winkler
foundation reacts only normal to the shell surface;
it is like an array of closely spaced springs
oriented normal to the shell surface. Units of the
foundation stiffness, K, are force per length
cubed (lb/in**3, for example). Note that the same
comment applies to EFNOOD that occurs in the "help"
paragraph associated with the variable, EFOUND.
(NOTE: IF YOU WANT TO SEE ALL THE "HELP" PARAGRAPHS
ASSOCIATED WITH ALL THE PROBLEM VARIABLES, INSPECT
THE XXX.PRO FILE, IN WHICH "XXX" REPRESENTS THE
GENERIC NAME OF THIS CASE).

75.1 Number IEMOD1 of rows in the array EMOD1: IEMOD1
80.1 elastic modulus in the fiber direction: EMOD1
85.1 elastic modulus transverse to fibers: EMOD2
90.1 in-plane shear modulus: G12
95.1 out-of-plane x-z shear modulus: G13
100.1 out-of-plain y-z shear modulus: G23
105.1 minor (small) Poisson ratio: NU
105.2
NU is the "NU21" Poisson ratio, governed by the
equation, E1*NU21 = E2*NU12. (Same input as for
BOSOR4 and PANDA2).

110.1 coef. of thermal expansion along the fibers: ALPHA1
115.1 coef. of thermal expansion transverse to fibers: ALPHA2
120.1 curing temperature difference: TEMCUR
120.2
TEMCUR is the difference between the temperature
at which the material "sets" during curing and the
service ("room") temperature. Use a POSITIVE number.

125.1 weight density of material: DENSTY
125.2
For example, Aluminum = 0.1 lb/in^3

130.1 Number ITHICK of rows in the array THICK: ITHICK
135.1 layer type thickness: THICK
140.1 layer type layup angle: ANGLE
145.1 layer type material type: MATTYP
150.1 Number JLAYTYP of columns in the array, LAYTYP: JLAYTYP
155.1 Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
160.1 layer type: LAYTYP
160.2
LAYTYP(layer no., segment no.) identifies the type of

that particular layer in "segment no." of the single
module model. LAYTYP(layer no., segment no.) points
to the layer with a certain thickness, THICK, layup
angle, ANGLE, and material type, MATTYP.

165.1 number of layers in each face sheet: NLAYRF
165.2
This is the number of layers, such that when
added to the number of layers wrapped around the
trapezoidal tool (NLAYRC, the next role 2 variable),
gives the total number of layers in the laminate
that forms the wall of the horizontal shell
segments that run along the base and crown of
each trapezoidal segment of the truss-core
sandwich structure.

170.1 number of layers around trapezoidal tool: NLAYRC
170.2
NLAYRC is the number of layers wrapped around the trapezoidal
tool, that is, the number of layers in the truss core.
For the upper and lower face sheets NLAYRC is added to
the number of layers, NLAYRF, to form the total number
of layers in the horizontal shell segment walls.
The number of layers in the truss-core webs will be
2 x NLAYRC.

175.1 0=pressure on top face sheet; 1=bottom face sheet: INTEXT
175.2
INTEXT = 0 means pressure acts on the top face sheet.
INTEXT = 1 means pressure acts on the bottom face sheet.
In the model used here the circumferential resultant
in each face sheet is assumed to be (PRESS x RADIUS)/2.
in which PRESS is the applied pressure (negative for
external pressure), and RADIUS is the radius of the
cylindrical shell..

180.1 number of segments in module: NSEGS
180.2
NSEGS must be 22 for the truss-core sandwich module.
At present NSEGS is set equal to 22 in SUBROUTINE BOSDEC.
Therefore, the value you provide here is overwritten.
You may feel that NSEGS should not be an input variable
because the user must always specify NSEGS = 22 for
now. However, perhaps this will change in the future.

185.0
You will next be asked to provide the number of modules
to be used in the model for general buckling (NMODULG)
and for local buckling (NMODULL). As of now NMODULG is
computed in SUBROUTINE BOSDEC. Therefore, the value of
NMODULG that you provide here will not be used. However,
it is possible that in the future SUBROUTINE BOSDEC may
be changed. Hence, NMODULG is retained as an input datum.
Use a number for NMODULG such as NMODULG = 13 (NOTE: you
are not allowed to use NMODULG > 13). For local buckling
you should use NMODULL = 1 until you have a preliminary
optimum design. Then, to refine the design, possibly
optimize again with NMODULL = 10 or something like that.
NOTE: The computer time increases significantly the
higher the value of NMODULL.

190.1 number of modules for general buckling: NMODULG
190.2
Use a number such as NMODULG = 13 (NOTE: you are
not permitted to use NMODULG > 13). The computer
time required for a SUPEROPT optimization increases
with increase in NMODULG. However, it is probably
best always to use the maximum value of NMODULG
that is permitted (NMODULG = 13) because you will
be less likely to obtain an unconservative design.
The value, NMODULG = 13, produces a BIGBOSOR4 model
with a total of 286 shell segments (286 = 22 x 13),
which is close to the maximum number of shell
segments (295) that BIGBOSOR4 can presently handle.

195.1 number of modules for local buckling: NMODULL
195.2

NMODULL is the number of modules, each of which contains NSEGS shell segments. In this application the best answer is usually 1 (that is, one module). That is because the buckling (or stress) model that consists of NSEGS shell segments is used for obtaining LOCAL behavior, such as local buckling. Only one module of NSEGS shell segments is needed for the determination of LOCAL behavior. However, there seems to be some dependence of the local buckling load on the number of modules, NMODULL, used in the model for local buckling. Hence, after you have optimized with the use of NMODULL = 1 you may want to check the optimum design with use of NMODULL = 10 or something like that.

200.1 low end of the M-range for general buckling: MLOWG
200.2

MLOWG should be 1 (an integer). Later it will be multiplied by 100 because of the "huge torus" model. The "huge torus" model is described in Part 12.2.2 of the paper, "Optimization of an axially compressed ring and stringer stiffened cylindrical shell with a general buckling modal imperfection", AIAA Paper 2007-2216, 48th AIAA Structures Meeting, Honolulu, Hawaii, April 2007.

205.1 high end of the M-range for general buckling: MHIGHG
205.2

Perhaps one should give a value 10 (or something like that). It depends on how many axial halfwaves you think will correspond to the critical general buckling mode. That number depends on the axial bending stiffness of the cylindrical shell wall, the length of the cylindrical shell, and the radius of the cylindrical shell. Use at least double the value of what you think is the critical value, that is, the value that corresponds to the lowest general buckling load. You may have to adjust the value of MHIGHG as your project proceeds.

210.1 low end of the range of local buckling waves: MLOWL
210.2

You should use MLOWL = 1 because the axial length of the portion of the cylindrical shell to be used in the LOCAL buckling analysis is equal to LENGTH x FACLEN, in which LENGTH is the actual axial length of the cylindrical shell, and FACLEN is the fraction of LENGTH to be included in the local buckling model.

215.1 high end of the M-range for local buckling: MHIGHL
215.2

We're referring here to the range of axial halfwaves to be used in the search for the critical LOCAL buckling load of the single truss-core sandwich module. Use a number such as 10 and adjust it later if necessary.

220.1 Number NCASES of load cases (environments): NCASES
225.1 total axial load (2 x pi x r x resultant): PX
225.2

PX = 2 x pi x r x axial resultant. Units = lb, for example. PX is negative for axial compression. The axial resultant in each shell segment is computed as if the shell were subject to uniform end shortening and the prebuckled state is a membrane state (no imperfections, no bending, no end "boundary layers").

230.1 pressure (negative for external pressure): PRESS
235.1 total "Load Set B" load: PX0
235.2

PX0 = 2 x pi x r x Nx0, in which Nx0 is an axial stress resultant. PX0 = positive for axial tension. Load Set B is the load set that is NOT to be multiplied by the buckling eigenvalue.

240.1 "Load Set B" pressure (external=negative): PRESS0
240.2

PRESS0 is positive for internal pressure. Load Set B

is the load set that is NOT to be multiplied by
the buckling eigenvalue.

Table 11, p. 6 of 6

245.0 local buckling load factor: LOCBUK
250.1 allowable for local buckling load factor: LOCBUKA
250.2
Usually, you supply 1.0 for LOCBUKA because LOCBUK is
a buckling load FACTOR, that is, a quantity that is
to be multiplied by the design loads, PX and PRESS,
in order to obtain the buckling load.

255.1 factor of safety for local buckling: LOCBUKF
255.2
In this "huge torus" model of the shell there cannot
exist any specified initial imperfection. Therefore,
you should probably use a factor of safety of 1.5
or something like that.

260.0 general buckling load factor: GENBUK
265.1 allowable for general buckling load factor: GENBUKA
265.2
Usually, you supply 1.0 for GENBUKA because GENBUK is
a buckling load FACTOR, that is, a quantity that is
to be multiplied by the design loads, PX and PRESS,
in order to obtain the buckling load.

270.1 general buckling factor of safety: GENBUKF
270.2
In this "huge torus" model of the shell there cannot
exist any specified initial imperfection. Therefore,
you should probably use a factor of safety of 2.0
or something like that.

275.1 Number JSTRM1 of columns in the array, STRM1: JSTRM1
280.0 stress component in material 1: STRM1
280.2
For a composite laminate there are 6 stress components
for which stress constraints are generated:
1. maximum tensile stess in the fiber direction
2. maximum compressive stress in the fiber direction
3. maximum tensile stress transverse to the fibers
4. maximum compressive stress transverse to the fibers
5. maximum in-plane shear stress
6. maximum effective (VonMises) stress for an
isotropic layer.

285.1 allowable stress in material 1: STRM1A
285.2
For a composite laminate there are 6 stress components
for which stress constraints are generated:
1. maximum tensile stess in the fiber direction:
STRM1A(i,1), in which "i" is the load set number
2. maximum compressive stress in the fiber direction:
STRM1A(i,2), in which "i" is the load set number
3. maximum tensile stress transverse to the fibers:
STRM1A(i,3), in which "i" is the load set number
4. maximum compressive stress transverse to the fibers:
STRM1A(i,4), in which "i" is the load set number
5. maximum in-plane shear stress:
STRM1A(i,5), in which "i" is the load set number
6. maximum effective (VonMises) stress for an
isotropic layer:
STRM1A(i,6), in which "i" is the load set number

290.1 factor of safety for stress in material 1: STRM1F
290.2
In this "huge torus" model of the shell there cannot
exist any specified initial imperfection. Therefore,
you should probably use a factor of safety of 1.5
or something like that.

295.0 stress component in material 2: STRM2
300.1 allowable for stress in material 2: STRM2A
305.1 factor of safety for stress in material 2: STRM2F
310.0 weight/area of the truss-core sandwich wall: WEIGHT
310.2
WEIGHT = weight per surface area of the truss-core
sandwich shell wall.

Table 12 behavior.trusscomp (19 pages)

```
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  The two test cases provided with GENOPT
C  provide examples of each method:
C   PLATE (test case 1): use of BEHAVIOR.NEW
C   PANEL (test case 2): use of ADDCODEn.NEW
C                        and STRUCT.NEW.
C
C   SEVEN ROLES THAT VARIABLES IN THIS SYSTEM OF PROGRAMS PLAY
C
C    A variable can have one of the following roles:
C
C    1 = a possible decision variable for optimization,
C        typically a dimension of a structure.
C    2 = a constant parameter (cannot vary as design evolves),
C        typically a control integer or material property,
C        but not a load, allowable, or factor of safety,
C        which are asked for later.
C    3 = a parameter characterizing the environment, such
C        as a load component or a temperature.
C    4 = a quantity that describes the response of the
C        structure, (e.g. stress, buckling load, frequency)
C    5 = an allowable, such as maximum allowable stress,
C        minimum allowable frequency, etc.
C    6 = a factor of safety
C    7 = the quantity that is to be minimized or maximized,
C        called the "objective function" (e.g. weight).
C ==========================================================
C
C  NAMES, DEFINITIONS, AND ROLES OF THE VARIABLES:

C YOU ARE USING WHAT I HAVE CALLED "GENOPT" TO GENERATE AN
```

Table 12, p. 2 of 19

```
C OPTIMIZATION PROGRAM FOR A PARTICULAR CLASS OF PROBLEMS.
C THE NAME YOU HAVE CHOSEN FOR THIS CLASS OF PROBLEMS IS: trusscomp

C "GENOPT" (GENeral OPTimization) was written during 1987-1988
C by Dr. David Bushnell, Dept. 93-30, Bldg. 251, (415)424-3237
C    Lockheed Missiles and Space Co., 3251 Hanover St.,
C    Palo Alto, California, USA  94304

C The optimizer used in GENOPT is called ADS, and was
C written by G. Vanderplaats [3]. It is based on the method
C of feasible directions [4].

C                              ABSTRACT

C "GENOPT" has the following purposes and properties:
C     1. Any relatively simple analysis is "automatically"
C        converted into an optimization of whatever system
C        can be analyzed with fixed properties. Please note
C        that GENOPT is not intended to be used for problems
C        that require elaborate data-base management systems
C        or large numbers of degrees of freedom.

C     2. The optimization problems need not be in fields nor
C        jargon familiar to me, the developer of GENOPT.
C        Although all of the example cases (See the cases
C        in the directories under genopt/case)
C        are in the field of structural analysis, GENOPT is
C        not limited to that field.

C     3. GENOPT is a program that writes other programs. These
C        programs, WHEN AUGMENTED BY USER-SUPPLIED CODING,
C        form a program system that should be user-friendly in
C        the GENOPT-user"s field. In this instance the user
C        of GENOPT must later supply FORTRAN coding that
C        calculates behavior in the problem class called "trusscomp".

C     4. Input data and textual material are elicited from
C        the user of GENOPT in a general enough way so that
C        he or she may employ whatever data, definitions, and
C        "help" paragraphs will make subsequent use of the
C        program system thus generated easy by those less
C        familiar with the class of problems "trusscomp" than
C        the GENOPT user.

C     5. The program system generated by GENOPT has the same
C        general architecture as previous programs written for
C        specific applications by the developer [7 - 16]. That
C        is, the command set is:

C           BEGIN      (User supplies starting design, loads,
C                       control integers, material properties,
C                       etc. in an interactive-help mode.)

C           DECIDE     (User chooses decision and linked
C                       variables and inequality constraints
C                       that are not based on behavior.)

C           MAINSETUP (User chooses output option, whether
C                       to perform analysis of a fixed design
C                       or to optimize, and number of design
C                       iterations.)

C           OPTIMIZE   (The program system performs, in a batch
C                       mode, the work specified in MAINSETUP.)

C           SUPEROPT   (Program tries to find the GLOBAL optimum
C                       design as described in Ref.[11] listed
C                       below (Many OPTIMIZEs in one run.)

C           CHANGE     (User changes certain parameters)

C           CHOOSEPLOT (User selects which quantities to plot
C                       vs. design iterations.)

C           DIPLOT     (User generates plots)

C           CLEANSPEC (User cleans out unwanted files.)
```

Table 2, p.3 of 19

```
C     A typical runstream is:
C       GENOPTLOG    (activate command set)
C       BEGIN        (provide starting design, loads, etc.)
C       DECIDE       (choose decision variables and bounds)
C       MAINSETUP    (choose print option and analysis type)
C       OPTIMIZE     (launch batch run for n design iterations)
C       OPTIMIZE     (launch batch run for n design iterations)
C       OPTIMIZE     (launch batch run for n design iterations)
C       OPTIMIZE     (launch batch run for n design iterations)
C       OPTIMIZE     (launch batch run for n design iterations)
C       CHANGE       (change some variables for new starting pt)
C       OPTIMIZE     (launch batch run for n design iterations)
C       OPTIMIZE     (launch batch run for n design iterations)
C       OPTIMIZE     (launch batch run for n design iterations)
C       OPTIMIZE     (launch batch run for n design iterations)
C       OPTIMIZE     (launch batch run for n design iterations)
C       CHOOSEPLOT   (choose which variables to plot)
C       DIPLOT       (plot variables v. iterations)
C       CHOOSEPLOT   (choose additional variables to plot)
C       DIPLOT       (plot more variables v design iterations)
C       CLEANSPEC    (delete extraneous files for specific case)

C  IMPORTANT:  YOU MUST ALWAYS GIVE THE COMMAND "OPTIMIZE"
C              SEVERAL TIMES IN SUCCESSION IN ORDER TO OBTAIN
C              CONVERGENCE! AN EXPLANATION OF WHY YOU MUST DO
C              THIS IS GIVEN ON P 580-582 OF THE PAPER "PANDA2,
C              PROGRAM FOR MINIMUM WEIGHT DESIGN OF STIFFENED,
C              COMPOSITE LOCALLY BUCKLED PANELS", Computers and
C              Structures, Vol. 25, No. 4, pp 469-605 (1987).

C Due to introduction of a "global" optimizer, SUPEROPT,
C described in Ref.[11], you can now use the runstream

C      BEGIN        (provide starting design, loads, etc.)
C      DECIDE       (choose decision variables and bounds)
C      MAINSETUP    (choose print option and analysis type)
C      SUPEROPT     (launch batch run for "global" optimization)
C      CHOOSEPLOT   (choose which variables to plot)
C      DIPLOT       (plot variables v. iterations)

C "Global" is in quotes because SUPEROPT does its best to find
C a true global optimum design. The user is strongly urged to
C execute SUPEROPT/CHOOSEPLOT several times in succession in
C order to determine an optimum that is essentially just as
C good as the theoretical true global optimum. Each execution
C of the series,
C      SUPEROPT
C      CHOOSEPLOT

C does the following:

C 1. SUPEROPT executes many sets of the two processors,
C    OPTIMIZE and AUTOCHANGE (AUTOCHANGE gets a new random
C    "starting" design), in which each set does the following:

C      OPTIMIZE          (perform k design iterations)
C      OPTIMIZE          (perform k design iterations)
C      OPTIMIZE          (perform k design iterations)
C      OPTIMIZE          (perform k design iterations)
C      OPTIMIZE          (perform k design iterations)
C      AUTOCHANGE        (get new starting design randomly)

C    SUPEROPT keeps repeating the above sequence until the
C    total number of design iterations reaches about 270.
C    The number of OPTIMIZEs per AUTOCHANGE is user-provided.

C 2. CHOOSEPLOT allows the user to plot stuff and resets the
C    total number of design iterations from SUPEROPT to zero.
C    After each execution of SUPEROPT the user MUST execute
C    CHOOSEPLOT: before the next execution of SUPEROPT the
C    total number of design iterations MUST be reset to zero.

C                      REFERENCES

C [1] Bushnell, D., "GENOPT--A program that writes
C user-friendly optimization code", International
C Journal of Solids and Structures, Vol. 26, No. 9/10,
```

```
C pp. 1173-1210, 1990. The same paper is contained in a
C bound volume of papers from the International Journal of
C Solids and Structures published in memory of Professor
C Charles D. Babcock, formerly with the California Institute
C of Technology.

C [2] Bushnell, D., "Automated optimum design of shells of
C revolution with application to ring-stiffened cylindrical
C shells with wavy walls", AIAA paper 2000-1663, 41st
C AIAA Structures Meeting, Atlanta, GA, April 2000. Also see
C Lockheed Martin report, same title, LMMS P525674, November
C 1999

C [2b] Bushnell, D., "Minimum weight design of imperfect
C isogrid-stiffened ellipsoidal shells under uniform external
C pressure", AIAA paper 2009-2702, 50th AIAA Structures
C Meeting, Palm Springs, CA, May 4-7, 2009

C [3] Vanderplaats, G. N., "ADS--a FORTRAN program for
C automated design synthesis, Version 2.01", Engineering
C Design Optimization, Inc, Santa Barbara, CA, January, 1987

C [4] Vanderplaats, G. N. and Sugimoto, H., "A general-purpose
C optimization program for engineering design", Computers
C and Structures, Vol. 24, pp 13-21, 1986

C [5] Bushnell, D., "BOSOR4: Program for stress, stability,
C and vibration of complex, branched shells of revolution",
C in STRUCTURAL ANALYSIS SYSTEMS, Vol. 2, edited by A.
C Niku-Lari, pp. 25-54, (1986)

C [6] Bushnell, D., "BOSOR5: Program for buckling of complex,
C branched shells of revolution including large deflections,
C plasticity and creep," in STRUCTURAL ANALYSIS SYSTEMS, Vol.
C 2,  edited by A. Niku-Lari, pp. 55-67, (1986)

C [7] Bushnell, D., "PANDA2--program for minimum weight
C design of stiffened, composite, locally buckled panels",
C COMPUTERS AND STRUCTURES, vol. 25, No. 4, pp 469-605, 1987

C [8] Bushnell, D., "Improved optimum design of dewar
C supports", COMPUTERS and STRUCTURES, Vol. 29, No. 1,
C pp. 1-56 (1988)

C [9] Bushnell, D., "SPHERE - Program for minimum weight
C design of isogrid-stiffened spherical shells under uniform
C external pressure", Lockheed Report F372046, January, 1990

C [10] Bushnell, D.,"Optimum design of imperf.isogrid-stiffened
C ellipsoidal shells...", written and placed in the file
C ..genopt/case/torisph/sdm50.report.pdf

C [11] Bushnell, D., "Recent enhancements to PANDA2", AIAA
C paper 96-1337-CP, Proc. 37th AIAA SDM Meeting, April 1996
C pp. 126-182, in particular, pp. 127-130

C [12] Bushnell, D., the file ..genopt/doc/getting.started

C [13] Bushnell, D., the case ..genopt/case/torisph

C [14] Bushnell, D., the case ..genopt/case/cylinder

C [15] Bushnell, D., the case ..genopt/case/wavycyl

C [16] Bushnell, D., the case ..genopt/case/plate

C [17] Bushnell, D., the case ..genopt/case/sphere

C=============================================================
C                TABLE 1       "GENOPT" COMMANDS
C=============================================================
C     HELPG        (get information on GENOPT.)
C     GENTEXT      (GENOPT user generate a prompt file, program
C                   fragments [see TABLE 5], programs [see
C                   TABLE 4]., and this and other files
C                   [see TABLE 5 and the rest of this file.])
C     GENPROGRAMS  (GENOPT user generate absolute elements:
C                   BEGIN.EXE, DECIDE.EXE, MAINSETUP.EXE,
```

Table 12, p. 5 of 19

```
C                   OPTIMIZE.EXE, CHANGE.EXE, STORE.EXE,
C                   CHOOSEPLOT.EXE, DIPLOT.EXE.)

C      BEGIN        (end user provide starting data.)
C      DECIDE       (end user choose decision variables, bounds,
C                    linked variables,inequality constraints.)
C      MAINSETUP    (end user set up strategy parameters.)
C      OPTIMIZE     (end user perform optimization, batch mode.)
C      SUPEROPT     (Program tries to find the GLOBAL optimum
C                    design as described in Ref.[11] listed
C                    above (Many OPTIMIZEs in one run.)

C      CHANGE       (end user change some parameters.)
C      CHOOSEPLOT   (end user choose which variables to plot v.
C                    design iterations.)
C      DIPLOT       (end user obtain plots.)
C      INSERT       (GENOPT user add parameters to the problem.)
C      CLEANGEN     (GENOPT user cleanup your GENeric files.)
C      CLEANSPEC    (end user cleanup your SPECific case files)

C   Please consult the following sources for more
C   information about GENOPT:
C        1.  GENOPT.STORY  and  HOWTO.RUN  and  GENOPT.NEWS
C        2.  Sample cases: (in the directory, genopt/case)
C        3.  NAME.DEF file, where NAME is the name chosen by
C            the GENOPT-user for a class of problems. (In this
C            case  NAME = trusscomp)
C        4.  GENOPT.HLP file    (type HELPG)
C=============================================================

C==============================================================
C   TABLE 2   GLOSSARY OF VARIABLES USED IN "trusscomp"
C==============================================================
C  ARRAY  NUMBER OF        PROMPT
C    ?   (ROWS,COLS)  ROLE  NUMBER   NAME                 DEFINITION OF VARIABLE
C                         (trusscomp.PRO)
C==============================================================
C    n    (   0,    0)    2      10    LENGTH   = length of the cylindrical shell
C    n    (   0,    0)    2      20    FACLEN   = fraction of LENGTH for local buck
C    n    (   0,    0)    2      25    RADIUS   = radius of the cylindrical shell
C    n    (   0,    0)    1      30    PITCH    = circumferential width of a single
C    n    (   0,    0)    1      35    BCROWN   = circumferential width of the trap
C    n    (   0,    0)    1      40    HEIGHT   = height of the truss-core sandwich
C    n    (   0,    0)    1      45    RACUTE   = local radius from base to side of
C    n    (   0,    0)    1      50    ROBTUS   = local radius from side to crown o
C    n    (   0,    0)    2      55    ENOODL   = axial modulus of a corner "noodle
C    n    (   0,    0)    2      60    DNOODL   = weight density of the "noodle" ma
C    n    (   0,    0)    2      65    EFOUND   = elastic foam foundation stiffness
C    n    (   0,    0)    2      70    EFNOOD   = elastic "noodle" Winkler foundati
C    n    (   0,    0)    2      75    IEMOD1   = material type in EMOD1(IEMOD1)
C    y    (  10,    0)    2      80    EMOD1    = elastic modulus in the fiber dire
C    y    (  10,    0)    2      85    EMOD2    = elastic modulus transverse to fib
C    y    (  10,    0)    2      90    G12      = in-plane shear modulus
C    y    (  10,    0)    2      95    G13      = out-of-plane x-z shear modulus
C    y    (  10,    0)    2     100    G23      = out-of-plain y-z shear modulus
C    y    (  10,    0)    2     105    NU       = minor (small) Poisson ratio
C    y    (  10,    0)    2     110    ALPHA1   = coef. of thermal expansion along
C    y    (  10,    0)    2     115    ALPHA2   = coef. of thermal expansion transv
C    y    (  10,    0)    2     120    TEMCUR   = curing temperature difference
C    y    (  10,    0)    2     125    DENSTY   = weight density of material
C    n    (   0,    0)    2     130    ITHICK   = layer type (thickness,layup angle
C    y    (  30,    0)    1     135    THICK    = layer type thickness
C    y    (  30,    0)    2     140    ANGLE    = layer type layup angle
C    y    (  30,    0)    2     145    MATTYP   = layer type material type
C    n    (   0,    0)    2     150    JLAYTYP  = shell segment number in LAYTYP(IL
C    n    (   0,    0)    2     155    ILAYTYP  = layer number in LAYTYP(ILAYTYP,JL
C    y    (  90,   50)    2     160    LAYTYP   = layer type
C    n    (   0,    0)    2     165    NLAYRF   = number of layers in each face she
C    n    (   0,    0)    2     170    NLAYRC   = number of layers around trapezoid
C    n    (   0,    0)    2     175    INTEXT   = 0 =pressure on top face sheet; 1
C    n    (   0,    0)    2     180    NSEGS    = number of segments in module
C    n    (   0,    0)    2     190    NMODULG  = number of modules for general buc
C    n    (   0,    0)    2     195    NMODULL  = number of modules for local buckl
C    n    (   0,    0)    2     200    MLOWG    = low end of the M-range for genera
C    n    (   0,    0)    2     205    MHIGHG   = high end of the M-range for gener
C    n    (   0,    0)    2     210    MLOWL    = low end of the range of local buc
C    n    (   0,    0)    2     215    MHIGHL   = high end of the M-range for local
C    n    (   0,    0)    2     220    NCASES   = Number of load cases (number of e
```

```
C     y    (  20,    0)    3      225    PX         = total axial load (2 x pi x r x re
C     y    (  20,    0)    3      230    PRESS      = pressure (negative for external p
C     y    (  20,    0)    3      235    PX0        = total "Load Set B" load
C     y    (  20,    0)    3      240    PRESS0     = "Load Set B" pressure (external=n
C     y    (  20,    0)    4      245    LOCBUK     = local buckling load factor
C     y    (  20,    0)    5      250    LOCBUKA    = allowable for local buckling load
C     y    (  20,    0)    6      255    LOCBUKF    = factor of safety for local buckli
C     y    (  20,    0)    4      260    GENBUK     = general buckling load factor
C     y    (  20,    0)    5      265    GENBUKA    = allowable for general buckling lo
C     y    (  20,    0)    6      270    GENBUKF    = general buckling factor of safety
C     n    (   0,    0)    2      275    JSTRM1     = number of stress constraints in S
C     y    (  20,    6)    4      280    STRM1      = stress component
C     y    (  20,    6)    5      285    STRM1A     = allowable stress in material 1
C     y    (  20,    6)    6      290    STRM1F     = factor of safety for stress in ma
C     y    (  20,    6)    4      295    STRM2      = stress in material 2
C     y    (  20,    6)    5      300    STRM2A     = allowable for stress in material
C     y    (  20,    6)    6      305    STRM2F     = factor of safety for stress in ma
C     n    (   0,    0)    7      310    WEIGHT     = weight of the truss-core sandwich
C
C=DECK      BEHX1
      SUBROUTINE BEHX1
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN local 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     LOCBUK(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 = local buckling load factor
C
C   OUTPUT:
C
C     LOCBUK(ILOADX)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
      REAL LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
      COMMON/FV13/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV14/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV20/ALPHA2(10),TEMCUR(10),DENSTY(10)
      REAL ALPHA2,TEMCUR,DENSTY
      COMMON/FV23/THICK(30),ITHICK
      REAL THICK
      COMMON/IV01/MATTYP(30)
      INTEGER MATTYP
      COMMON/IV02/LAYTYP(90,50),ILAYTYP,JLAYTYP
      INTEGER LAYTYP
      COMMON/IV03/NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      INTEGER NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      COMMON/FV24/ANGLE(30)
      REAL ANGLE
      COMMON/FV25/PX(20)
      REAL PX
```

Table 12, p. 7 of 19

```
      COMMON/FV31/LOCBUK(20),LOCBUKA(20),LOCBUKF(20)
      REAL LOCBUK,LOCBUKA,LOCBUKF
      COMMON/FV34/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV37/STRM1(20,6 ),JSTRM1 ,STRM1A(20,6 ),STRM1F(20,6 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV40/STRM2(20,6 ),STRM2A(20,6 ),STRM2F(20,6 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/IV10/MHIGHG,MLOWL,MHIGHL
      INTEGER MHIGHG,MLOWL,MHIGHL
      COMMON/FV09/ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      REAL ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      COMMON/FV26/PRESS(20),PX0(20),PRESS0(20)
      REAL PRESS,PX0,PRESS0
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/BUCKN0/N0B,NMAXB
      COMMON/RBEGX/RBIG0,RBIGL,RBIGG
      COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
      COMMON/EIGALL/EIG0,EIG1,EIG2,EIG3,EIG4
      COMMON/WAVALL/NWAV0,NWAV1,NWAV2,NWAV3,NWAV4
      COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
      common/caseblock/CASE
      CHARACTER*28 CASE
      CHARACTER*35 CASA
C
      PI = 3.1415927
C
      INDIC = 4
      RAVE = 100.*LENGTH*FACLEN/PI
      RBIG0 = RAVE
      RBIGL = RAVE -FLOAT(NMODULL)*PITCH/2.
      N0B = MLOWL*100
      NMAXB = MHIGHL*100
C
      CALL BOSDEC(0,24,ILOADX,INDIC)
      CALL B4READ
      CALL GASP(DUM1,DUM2,-2,DUM3)
      CALL BOSDEC(1,24,ILOADX,INDIC)
C
      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       local buckling load
          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 ' local buckling load',
     1   CASA
       ENDIF
C
       CALL B4READ
       IF (IMODX.EQ.0) THEN
          N0BX = N0B
          NMINBX = N0B
          NMAXBX = NMAXB
          INCRBX = 100
       ELSE
          NOBX = NWAV1
          NMINBX = NWAV1
          NMAXBX = NWAV1
          INCRBX = 100
```

BOSDEC creates an input file for BIGBOSOR4

BIGBOSOR4 preprocessor

BIGBOSOR4 preprocessor

GENOPT user added this to the "skeletal" BEHX1 created automatically by GENTEXT.

Table 12, p. 8 of 19

BIGBOSCR4 mainprocessor

```
      ENDIF
      REWIND IFILE9
      CALL STOCM1(IFILE9)
      CALL STOCM2(IFILE9)
      CALL B4MAIN
      CALL GASP(DUM1,DUM2,-2,DUM3)
      IF (IMODX.EQ.0) THEN
         EIG1 = EIGCRT
         NWAV1= NWVCRT
      ENDIF
C
      IF (IMODX.EQ.0) THEN
        WRITE(IFILE,'(/,A)')
     1 ' LOCAL BUCKLING LOAD FACTORS AND MODES (BEHX1)'
        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, LOCBUK=',EIGCRT
        WRITE(IFILE,'(A,I5)')
     1' Critical number of circumferential waves, NWVCRT=',NWVCRT
      ENDIF
C23456789012345678901234567890123456789012345678901234567890123456789012
C
      LOCBUK(ILOADX) = EIGCRT
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX2
      SUBROUTINE BEHX2
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)
C
C   PURPOSE: OBTAIN general 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     GENBUK(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 = general buckling load factor
C
C   OUTPUT:
C
C     GENBUK(ILOADX)
C
      CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
     COMMON/FV01/LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
     REAL LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
     COMMON/FV13/EMOD1(10),IEMOD1
     REAL EMOD1
     COMMON/FV14/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
     REAL EMOD2,G12,G13,G23,NU,ALPHA1
```

Table 12, p. 9 of 19

```
      COMMON/FV20/ALPHA2(10),TEMCUR(10),DENSTY(10)
      REAL ALPHA2,TEMCUR,DENSTY
      COMMON/FV23/THICK(30),ITHICK
      REAL THICK
      COMMON/IV01/MATTYP(30)
      INTEGER MATTYP
      COMMON/IV02/LAYTYP(90,50),ILAYTYP,JLAYTYP
      INTEGER LAYTYP
      COMMON/IV03/NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      INTEGER NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      COMMON/FV24/ANGLE(30)
      REAL ANGLE
      COMMON/FV25/PX(20)
      REAL PX
      COMMON/FV31/LOCBUK(20),LOCBUKA(20),LOCBUKF(20)
      REAL LOCBUK,LOCBUKA,LOCBUKF
      COMMON/FV34/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV37/STRM1(20,6 ),JSTRM1 ,STRM1A(20,6 ),STRM1F(20,6 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV40/STRM2(20,6 ),STRM2A(20,6 ),STRM2F(20,6 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/IV10/MHIGHG,MLOWL,MHIGHL
      INTEGER MHIGHG,MLOWL,MHIGHL
      COMMON/FV09/ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      REAL ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      COMMON/FV26/PRESS(20),PX0(20),PRESS0(20)
      REAL PRESS,PX0,PRESS0
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/BUCKN0/N0B,NMAXB
      COMMON/RBEGX/RBIG0,RBIGL,RBIGG
      COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
      COMMON/EIGALL/EIG0,EIG1,EIG2,EIG3,EIG4
      COMMON/WAVALL/NWAV0,NWAV1,NWAV2,NWAV3,NWAV4
      COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
      common/caseblock/CASE
      CHARACTER*28 CASE
      CHARACTER*35 CASA
C
      PI = 3.1415927
C
      INDIC = 4
      RAVE = 100.*LENGTH/PI
      RBIGG = RAVE -0.707*RADIUS
      N0B = MLOWG*100
      NMAXB = MHIGHG*100
      CALL BOSDEC(0,24,ILOADX,INDIC)
      CALL B4READ
      CALL GASP(DUM1,DUM2,-2,DUM3)
C
      CALL BOSDEC(2,24,ILOADX,INDIC)
C
      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      general buckling load
         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 ' general buckling load',
     1   CASA
```

GENOPT user added this to the "skeletal" BEHX2 created automatically by GENTEXT

Table 12, p. 10 of 19

```
      ENDIF
C
      CALL B4READ
      IF (IMODX.EQ.0) THEN
         N0BX = N0B
         NMINBX = N0B
         NMAXBX = NMAXB
         INCRBX = 100
      ELSE
         NOBX = NWAV2
         NMINBX = NWAV2
         NMAXBX = NWAV2
         INCRBX = 100
      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
C
      IF (IMODX.EQ.0) THEN
        WRITE(IFILE,'(/,A)')
     1 ' GENERAL 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, GENBUK=',EIGCRT
        WRITE(IFILE,'(A,I5)')
     1' Critical number of circumferential waves, NWVCRT=',NWVCRT
      ENDIF
C
      GENBUK(ILOADX) = EIGCRT
C
      RETURN
      END
C
C
C
C
C=DECK      BEHX3
      SUBROUTINE BEHX3
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)
C
C   PURPOSE: OBTAIN stress component
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     STRM1(ILOADX,JCOL)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C   THE jth COLUMN  (JCOL)
C   INDEX IS DEFINED AS FOLLOWS:
C     number of stress constraints
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
```

GENOPT user code

```
C     ILOADX = ith LOADING COMBINATION
C     JCOL   = jth column of STRM1
C     JCOL   = number of stress constraints
C     PHRASE = stress component
C
C   OUTPUT:
C
C     STRM1(ILOADX,JCOL)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
      REAL LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
      COMMON/FV13/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV14/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV20/ALPHA2(10),TEMCUR(10),DENSTY(10)
      REAL ALPHA2,TEMCUR,DENSTY
      COMMON/FV23/THICK(30),ITHICK
      REAL THICK
      COMMON/IV01/MATTYP(30)
      INTEGER MATTYP
      COMMON/IV02/LAYTYP(90,50),ILAYTYP,JLAYTYP
      INTEGER LAYTYP
      COMMON/IV03/NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      INTEGER NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      COMMON/FV24/ANGLE(30)
      REAL ANGLE
      COMMON/FV25/PX(20)
      REAL PX
      COMMON/FV31/LOCBUK(20),LOCBUKA(20),LOCBUKF(20)
      REAL LOCBUK,LOCBUKA,LOCBUKF
      COMMON/FV34/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV37/STRM1(20,6 ),JSTRM1 ,STRM1A(20,6 ),STRM1F(20,6 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV40/STRM2(20,6 ),STRM2A(20,6 ),STRM2F(20,6 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/IV10/MHIGHG,MLOWL,MHIGHL
      INTEGER MHIGHG,MLOWL,MHIGHL
      COMMON/FV09/ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      REAL ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      COMMON/FV26/PRESS(20),PX0(20),PRESS0(20)
      REAL PRESS,PX0,PRESS0
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
      COMMON/NLAYRX/NLAYER(50)
      COMMON/STRANX/EPSX,EPSY1,EPSY3,EPSY5
      DIMENSION EPSY(50)
      CHARACTER*15 MODE(7)
C
      MODE(1) = ' 0 deg. tension'
      MODE(2) = ' 0 deg. comp.  '
      MODE(3) = '90 deg. tension'
      MODE(4) = '90 deg. comp.  '
      MODE(5) = ' in-plane shear'
      MODE(6) = 'von Mises yield'
      MODE(7) = '  no failure   '
C
      PI = 3.1415927
      DTR = PI/180.
C
      EPSY(1)  = EPSY1
      EPSY(2)  = EPSY5
      EPSY(3)  = EPSY3
      EPSY(4)  = EPSY5
      EPSY(5)  = EPSY5
      EPSY(6)  = EPSY1
      EPSY(7)  = EPSY5
      EPSY(8)  = EPSY3
      EPSY(9)  = EPSY5
      EPSY(10)  = EPSY5
      EPSY(11)  = EPSY5
      EPSY(12)  = EPSY5
      EPSY(13)  = EPSY5
```

GENOPT user added this to the "skeletal" BEHX3 created automatically by GENTEXT.

Table 12, p. 12 of 19

GENOPT user wrote this part of BEHX3

```
      EPSY(14) = EPSY5
      EPSY(15) = EPSY1
      EPSY(16) = EPSY5
      EPSY(17) = EPSY3
      EPSY(18) = EPSY1
      EPSY(19) = EPSY5
      EPSY(20) = EPSY5
      EPSY(21) = EPSY5
      EPSY(22) = EPSY3
C
      S1MAX = 0.
      S1MIN = 0.
      S2MAX = 0.
      S2MIN = 0.
      S12MAX= 0.
      SEFFMX= 0.
C
      DO 300 ISEG = 1,NSEGS
C
      EX = EPSX
      EY = EPSY(ISEG)
      EXY = 0.
C
      NLAY = NLAYER(ISEG)
C
      DO 200 ILAYER = 1,NLAY
C
        LTYPE = LAYTYP(ILAYER,ISEG)
        M = MATTYP(LTYPE)
        IF (M.NE.1) GO TO 200
        A1T = -ALPHA1(M)*TEMCUR(M)
        A2T = -ALPHA2(M)*TEMCUR(M)
        ARG = ANGLE(LTYPE)*DTR
        CARG = COS(ARG)
        SARG = SIN(ARG)
        C2 = CARG**2
        S2 = SARG**2
        SC = CARG*SARG
C
C  E1, E2, E12 ARE STRAIN COMPONENTS IN MATERIAL COORDINATES...
C
        E1 = C2*EX + S2*EY + SC*EXY
        E2 = S2*EX + C2*EY - SC*EXY
        E12= -2.*SC*(EX - EY) + (C2-S2)*EXY
C
        U12 = NU(M)
        U21 = 0.
        IF (EMOD2(M).GT.0.) U21 = U12*EMOD1(M)/EMOD2(M)
        UD = 1. - U12*U21
C
C  SIG1, SIG2, SIG12 = STRESS COMPONENTS IN MATERIAL COORDINATES...
C
        SIG1 = (     EMOD1(M)*(E1-A1T) +U12*EMOD1(M)*(E2-A2T))/UD
        SIG2 = (U12*EMOD1(M)*(E1-A1T) +     EMOD2(M)*(E2-A2T))/UD
        SIG12= G12(M)*E12
C
C  Determine if the material M is isotropic, and if so, compute
C  the effective (VonMises) stress:
        DIFF1 = 1.
        DIFF2 = 1.
        SIGEFF = 0.
        ISOMAT = 0
C23456789012345678901234567890123456789012345678901234567890123456789012
        IF (EMOD1(M).GT.0.) DIFF1 =ABS(EMOD1(M) -EMOD2(M))/ABS(EMOD1(M))
        GISO = EMOD1(M)/(2.*(1.+U12))
        IF (G12(M).GT.0.)  DIFF2 = ABS(G12(M) - GISO)/ABS(G12(M))
        IF (DIFF1.LT.0.05.AND.DIFF2.LT.0.1) ISOMAT = 1
        IF (ISOMAT.EQ.1)
     1   SIGEFF  = SQRT(SIG1**2 +SIG2**2 -SIG1*SIG2 +3.*SIG12**2)
C
        IF (ISOMAT.EQ.0) THEN
           S1MAX = MAX(S1MAX,SIG1)
           S1MIN = MIN(S1MIN,SIG1)
           S2MAX = MAX(S2MAX,SIG2)
           S2MIN = MIN(S2MIN,SIG2)
           S12MAX = MAX(S12MAX,ABS(SIG12))
        ELSE
           SEFFMX = MAX(SEFFMX,SIGEFF)
```

Table 12, p.13 of 19

GENOPT user written code

```
        ENDIF
C
  200 CONTINUE
  300 CONTINUE
C
      IF (S1MIN.GT.0.0) S1MIN = 0.0
      IF (S2MIN.GT.0.0) S2MIN = 0.0
      S1MIN = ABS(S1MIN)
      S2MIN = ABS(S2MIN)
C
      IF (SEFFMX.EQ.0.0) THEN
         STRM1(ILOADX,1) = S1MAX
         STRM1(ILOADX,2) = S1MIN
         STRM1(ILOADX,3) = S2MAX
         STRM1(ILOADX,4) = S2MIN
         STRM1(ILOADX,5) = S12MAX
      ELSE
         STRM1(ILOADX,6) = SEFFMX
      ENDIF
C
      IF (IMODX.EQ.0.AND.JCOL.EQ.1) THEN
         WRITE(IFILE,'(/,A)')
     1 ' Maximum stress components from BEHX3 (Material type 1):'
         IF (SEFFMX.EQ.0.0) THEN
            DO 400 I = 1,5
            WRITE(IFILE,'(A,1P,E14.6)') MODE(I),STRM1(ILOADX,I)
  400       CONTINUE
         ELSE
            WRITE(IFILE,'(A,1P,E14.6)') MODE(6),STRM1(ILOADX,6)
         ENDIF
      ENDIF
C
      RETURN
      END
C
C
C
C
C=DECK        BEHX4
      SUBROUTINE BEHX4
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)
C
C   PURPOSE: OBTAIN stress in material 2
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     STRM2(ILOADX,JCOL)
C
C   AS OUTPUT. THE ith CASE REFERS
C   TO ith ENVIRONMENT (e.g. load com-
C   bination).
C   THE jth COLUMN  (JCOL)
C   INDEX IS DEFINED AS FOLLOWS:
C     number of stress constraints
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     JCOL   = jth column of STRM2
C     JCOL   = number of stress constraints
C     PHRASE = stress in material 2
C
C  OUTPUT:
C
```

```
C      STRM2(ILOADX,JCOL)
C
        CHARACTER*80 PHRASE
C   INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
      REAL LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
      COMMON/FV13/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV14/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV20/ALPHA2(10),TEMCUR(10),DENSTY(10)
      REAL ALPHA2,TEMCUR,DENSTY
      COMMON/FV23/THICK(30),ITHICK
      REAL THICK
      COMMON/IV01/MATTYP(30)
      INTEGER MATTYP
      COMMON/IV02/LAYTYP(90,50),ILAYTYP,JLAYTYP
      INTEGER LAYTYP
      COMMON/IV03/NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      INTEGER NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      COMMON/FV24/ANGLE(30)
      REAL ANGLE
      COMMON/FV25/PX(20)
      REAL PX
      COMMON/FV31/LOCBUK(20),LOCBUKA(20),LOCBUKF(20)
      REAL LOCBUK,LOCBUKA,LOCBUKF
      COMMON/FV34/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV37/STRM1(20,6 ),JSTRM1 ,STRM1A(20,6 ),STRM1F(20,6 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV40/STRM2(20,6 ),STRM2A(20,6 ),STRM2F(20,6 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/IV10/MHIGHG,MLOWL,MHIGHL
      INTEGER MHIGHG,MLOWL,MHIGHL
      COMMON/FV09/ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      REAL ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      COMMON/FV26/PRESS(20),PX0(20),PRESS0(20)
      REAL PRESS,PX0,PRESS0
C
C
C   INSERT SUBROUTINE STATEMENTS HERE.
C
      COMMON/NLAYRX/NLAYER(50)
      COMMON/STRANX/EPSX,EPSY1,EPSY3,EPSY5
      DIMENSION EPSY(50)
      CHARACTER*15 MODE(7)
C
      MODE(1) = ' 0 deg. tension'
      MODE(2) = ' 0 deg. comp.  '
      MODE(3) = '90 deg. tension'
      MODE(4) = '90 deg. comp.  '
      MODE(5) = ' in-plane shear'
      MODE(6) = 'von Mises yield'
      MODE(7) = '  no failure   '
C
      PI = 3.1415927
      DTR = PI/180.
C
      EPSY(1)  = EPSY1
      EPSY(2)  = EPSY5
      EPSY(3)  = EPSY3
      EPSY(4)  = EPSY5
      EPSY(5)  = EPSY5
      EPSY(6)  = EPSY1
      EPSY(7)  = EPSY5
      EPSY(8)  = EPSY3
      EPSY(9)  = EPSY5
      EPSY(10)  = EPSY5
      EPSY(11)  = EPSY5
      EPSY(12)  = EPSY5
      EPSY(13)  = EPSY5
      EPSY(14)  = EPSY5
      EPSY(15)  = EPSY1
      EPSY(16)  = EPSY5
      EPSY(17)  = EPSY3
      EPSY(18)  = EPSY1
      EPSY(19)  = EPSY5
      EPSY(20)  = EPSY5
```

GENOPT user-written (almost a copy of the code written in BEHX3)

Table 12, p. 15 of 19

```
      EPSY(21) = EPSY5
      EPSY(22) = EPSY3
C
      S1MAX = 0.
      S1MIN = 0.
      S2MAX = 0.
      S2MIN = 0.
      S12MAX= 0.
      SEFFMX= 0.
C
      DO 300 ISEG = 1,NSEGS
C
      EX = EPSX
      EY = EPSY(ISEG)
      EXY = 0.
C
      NLAY = NLAYER(ISEG)
C
      DO 200 ILAYER = 1,NLAY
C
        LTYPE = LAYTYP(ILAYER,ISEG)
        M = MATTYP(LTYPE)
        IF (M.NE.2) GO TO 200
        A1T = -ALPHA1(M)*TEMCUR(M)
        A2T = -ALPHA2(M)*TEMCUR(M)
        ARG = ANGLE(LTYPE)*DTR
        CARG = COS(ARG)
        SARG = SIN(ARG)
        C2 = CARG**2
        S2 = SARG**2
        SC = CARG*SARG
C
C  E1, E2, E12 ARE STRAIN COMPONENTS IN MATERIAL COORDINATES...
C
        E1 = C2*EX + S2*EY + SC*EXY
        E2 = S2*EX + C2*EY - SC*EXY
        E12= -2.*SC*(EX - EY) + (C2-S2)*EXY
C
        U12 = NU(M)
        U21 = 0.
        IF (EMOD2(M).GT.0.) U21 = U12*EMOD1(M)/EMOD2(M)
        UD = 1. - U12*U21
C
C  SIG1, SIG2, SIG12 = STRESS COMPONENTS IN MATERIAL COORDINATES...
C
        SIG1 = (     EMOD1(M)*(E1-A1T) +U12*EMOD1(M)*(E2-A2T))/UD
        SIG2 = (U12*EMOD1(M)*(E1-A1T) +     EMOD2(M)*(E2-A2T))/UD
        SIG12= G12(M)*E12
C
C  Determine if the material M is isotropic, and if so, compute
C  the effective (VonMises) stress:
        DIFF1 = 1.
        DIFF2 = 1.
        SIGEFF = 0.
        ISOMAT = 0
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
        IF (EMOD1(M).GT.0.) DIFF1 =ABS(EMOD1(M) -EMOD2(M))/ABS(EMOD1(M))
        GISO = EMOD1(M)/(2.*(1.+U12))
        IF (G12(M).GT.0.)  DIFF2 = ABS(G12(M) - GISO)/ABS(G12(M))
        IF (DIFF1.LT.0.05.AND.DIFF2.LT.0.1) ISOMAT = 1
        IF (ISOMAT.EQ.1)
     1   SIGEFF  = SQRT(SIG1**2 +SIG2**2 -SIG1*SIG2 +3.*SIG12**2)
C
        IF (ISOMAT.EQ.0) THEN
           S1MAX = MAX(S1MAX,SIG1)
           S1MIN = MIN(S1MIN,SIG1)
           S2MAX = MAX(S2MAX,SIG2)
           S2MIN = MIN(S2MIN,SIG2)
           S12MAX = MAX(S12MAX,ABS(SIG12))
        ELSE
           SEFFMX = MAX(SEFFMX,SIGEFF)
        ENDIF
C
  200 CONTINUE
  300 CONTINUE
C
      IF (S1MIN.GT.0.0) S1MIN = 0.0
      IF (S2MIN.GT.0.0) S2MIN = 0.0
```

GENOPT user wrote this (almost a copy of GENOPT user written code in BEHX3)

Table 12, p.16 of 19

```
      S1MIN = ABS(S1MIN)
      S2MIN = ABS(S2MIN)
C
      IF (SEFFMX.EQ.0.0) THEN
         STRM2(ILOADX,1) = S1MAX
         STRM2(ILOADX,2) = S1MIN
         STRM2(ILOADX,3) = S2MAX
         STRM2(ILOADX,4) = S2MIN
         STRM2(ILOADX,5) = S12MAX
      ELSE
         STRM2(ILOADX,6) = SEFFMX
      ENDIF
C
      IF (IMODX.EQ.0.AND.JCOL.EQ.1) THEN
         WRITE(IFILE,'(/,A)')
     1 ' Maximum stress components from BEHX4 (Material type 2):'
         IF (SEFFMX.EQ.0.0) THEN
            DO 400 I = 1,5
            WRITE(IFILE,'(A,1P,E14.6)') MODE(I),STRM2(ILOADX,I)
  400       CONTINUE
         ELSE
            WRITE(IFILE,'(A,1P,E14.6)') MODE(6),STRM2(ILOADX,6)
         ENDIF
      ENDIF
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/FV01/LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
      REAL LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
      COMMON/FV13/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV14/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV20/ALPHA2(10),TEMCUR(10),DENSTY(10)
      REAL ALPHA2,TEMCUR,DENSTY
      COMMON/FV23/THICK(30),ITHICK
      REAL THICK
      COMMON/IV01/MATTYP(30)
      INTEGER MATTYP
      COMMON/IV02/LAYTYP(90,50),ILAYTYP,JLAYTYP
      INTEGER LAYTYP
      COMMON/IV03/NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      INTEGER NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      COMMON/FV24/ANGLE(30)
      REAL ANGLE
      COMMON/FV25/PX(20)
      REAL PX
      COMMON/FV31/LOCBUK(20),LOCBUKA(20),LOCBUKF(20)
      REAL LOCBUK,LOCBUKA,LOCBUKF
      COMMON/FV34/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV37/STRM1(20,6 ),JSTRM1 ,STRM1A(20,6 ),STRM1F(20,6 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV40/STRM2(20,6 ),STRM2A(20,6 ),STRM2F(20,6 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/IV10/MHIGHG,MLOWL,MHIGHL
      INTEGER MHIGHG,MLOWL,MHIGHL
      COMMON/FV09/ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
```

GENOPT user written code

Created entirely by GENTEXT

```
      REAL ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      COMMON/FV26/PRESS(20),PX0(20),PRESS0(20)
      REAL PRESS,PX0,PRESS0
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
```

Table 12, p.17 of 19

Created entirely by GENTEXT

```
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 the truss-core sandwich module
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 the truss-core sandwich module
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV01/LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
      REAL LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
      COMMON/FV13/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV14/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV20/ALPHA2(10),TEMCUR(10),DENSTY(10)
      REAL ALPHA2,TEMCUR,DENSTY
      COMMON/FV23/THICK(30),ITHICK
      REAL THICK
      COMMON/IV01/MATTYP(30)
      INTEGER MATTYP
      COMMON/IV02/LAYTYP(90,50),ILAYTYP,JLAYTYP
      INTEGER LAYTYP
      COMMON/IV03/NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      INTEGER NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      COMMON/FV24/ANGLE(30)
      REAL ANGLE
      COMMON/FV25/PX(20)
      REAL PX
      COMMON/FV31/LOCBUK(20),LOCBUKA(20),LOCBUKF(20)
      REAL LOCBUK,LOCBUKA,LOCBUKF
      COMMON/FV34/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV37/STRM1(20,6 ),JSTRM1 ,STRM1A(20,6 ),STRM1F(20,6 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV40/STRM2(20,6 ),STRM2A(20,6 ),STRM2F(20,6 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/IV10/MHIGHG,MLOWL,MHIGHL
      INTEGER MHIGHG,MLOWL,MHIGHL
      COMMON/FV09/ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      REAL ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      COMMON/FV26/PRESS(20),PX0(20),PRESS0(20)
      REAL PRESS,PX0,PRESS0
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
      COMMON/NLAYRX/NLAYER(50)
      COMMON/GEOM1X/RADACU,RADOBT,PHI,ALPHA,AREA1,AREA2
```

Table 12, p.18 of 19

The objective is computed.

Created by GENTEXT

GENOPT user added this

108

```
      COMMON/GEOM2X/RP1(50),RP2(50),ZP1(50),ZP2(50),RPC(50),ZPC(50)
C
      WGTALL = 0.
C
      DO 300 ISEG = 1,NSEGS
C
C23456789012345678901234567890123456789012345678901234567890123456789012
      ARING1 = 0.
      ARING2 = 0.
      SEGLNG =SQRT((RP2(ISEG) -RP1(ISEG))**2 +(ZP2(ISEG) -ZP1(ISEG))**2)
      IF (ISEG.EQ.2.OR.ISEG.EQ.10.OR.ISEG.EQ.16.OR.ISEG.EQ.20) THEN
         SEGLNG = RADOBT*2.*PHI
         ARING2 = AREA2
      ENDIF
      IF (ISEG.EQ.5.OR.ISEG.EQ.9.OR.ISEG.EQ.14.OR.ISEG.EQ.21) THEN
         SEGLNG = RADACU*2.*ALPHA
         ARING1 = AREA1
      ENDIF
C
      NLAY = NLAYER(ISEG)
      WTHICK = 0.
      DO 200 ILAYER = 1,NLAY
        LTYPE = LAYTYP(ILAYER,ISEG)
        M = MATTYP(LTYPE)
        WTHICK = WTHICK + THICK(LTYPE)*DENSTY(M)
  200 CONTINUE
C
      WGTALL = WGTALL + WTHICK*SEGLNG +(ARING1+ARING2)*DNOODL
C
  300 CONTINUE
C
      WEIGHT = WGTALL/PITCH
C
      IF (IMODX.EQ.0) THEN
         WRITE(IFILE,'(/,A,1P,E14.6,/,A,/,1P6E12.4)')
     1 ' Objective =weight per surface area from OBJECT =',WEIGHT,
     1 ' PHI,ALPHA,AREA1,AREA2,RADACU,RADOBT=',
     1   PHI,ALPHA,AREA1,AREA2,RADACU,RADOBT
      ENDIF
C
      OBJGEN =WEIGHT
C
      RETURN
      END
C
C
C
```

Table 12, p.19 of 19

GENOPT user written code

End of BEHAVIOR.TRUSSCOMP file

Table 13a bosdec.trusscomp (13 pages)

An abridged version of the file, bosdec.trusscomp, which generates valid input files for BIGBOSOR4 for "huge torus" models of the axially compressed cylindrical shell with a composite truss-core sandwich wall construction. Two types of "huge torus" models are constructed: 1. a "huge torus" model for local buckling (INDX = 1), and 2. a "huge torus" model for general buckling (INDX = 2).

```
C=DECK      BOSDEC
C
C  PURPOSE IS TO SET UP BIGBOSOR4 INPUT FILE FOR "trusscomp"
C
      SUBROUTINE BOSDEC(INDX,IFIL14,ILOADX,INDIC)
C  Insert labelled common blocks: trusscomp.COM
      COMMON/FV01/LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
      REAL LENGTH,FACLEN,RADIUS,PITCH,BCROWN,HEIGHT,RACUTE,ROBTUS
      COMMON/FV13/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV14/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV20/ALPHA2(10),TEMCUR(10),DENSTY(10)
      REAL ALPHA2,TEMCUR,DENSTY
      COMMON/FV23/THICK(30),ITHICK
      REAL THICK
      COMMON/IV01/MATTYP(30)
      INTEGER MATTYP
      COMMON/IV02/LAYTYP(90,50),ILAYTYP,JLAYTYP
      INTEGER LAYTYP
      COMMON/IV03/NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      INTEGER NLAYRF,NLAYRC,INTEXT,NSEGS,NMODULG,NMODULL,MLOWG
      COMMON/FV24/ANGLE(30)
      REAL ANGLE
      COMMON/FV25/PX(20)
      REAL PX
      COMMON/FV31/LOCBUK(20),LOCBUKA(20),LOCBUKF(20)
      REAL LOCBUK,LOCBUKA,LOCBUKF
      COMMON/FV34/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV37/STRM1(20,6 ),JSTRM1 ,STRM1A(20,6 ),STRM1F(20,6 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV40/STRM2(20,6 ),STRM2A(20,6 ),STRM2F(20,6 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/IV10/MHIGHG,MLOWL,MHIGHL
      INTEGER MHIGHG,MLOWL,MHIGHL
      COMMON/FV09/ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      REAL ENOODL,DNOODL,EFOUND,EFNOOD,WEIGHT
      COMMON/FV26/PRESS(20),PX0(20),PRESS0(20)
      REAL PRESS,PX0,PRESS0
C  end of weldland.COM        trusscomp
      COMMON/MODULX/MODULL,MODULG
      COMMON/CSKINY/CSKIN1(6,6,295)
      COMMON/BUCKN0/N0B,NMAXB
      COMMON/RBEGX/RBIG0,RBIGL,RBIGG
      COMMON/NLAYRX/NLAYER(50)
      COMMON/STRANX/EPSX,EPSY1,EPSY3,EPSY5
      COMMON/GEOM1X/RADACU,RADOBT,PHI,ALPHA,AREA1,AREA2
      COMMON/GEOM2X/RL1(50),RL2(50),ZL1(50),ZL2(50),RLC(50),ZLC(50)
      DIMENSION RP1(50),RP2(50),ZP1(50),ZP2(50),RPC(50),ZPC(50)
      DIMENSION R1(50),R2(50),Z1(50),Z2(50),RC(50),ZC(50),SROT(50)
      DIMENSION FOUND(50),PRESUR(50),ZVAL(50),ZRING(50),ZNORM(50)
      DIMENSION NSHAPE(50),NNODES(50),NRING(50),IRING(300),NSUR(50)
      DIMENSION DRADIA(50),DAXIAL(50),RGAREA(50),FN10(50),FN20(50)
      DIMENSION IDISP(50,4),JPREV(50,2),JNODPV(50,2)
      DIMENSION NPREV(50),NODPRV(50,2),NGRND(50),NODGRD(50,2)
      DIMENSION NTYPE(50),IPTRNG(50)
      DIMENSION RCURV1(50),RCURV2(50),ZCURV1(50),ZCURV2(50)
      DIMENSION RCURVC(50),ZCURVC(50)
      DIMENSION NLAYR2(10),LAYTP2(90,10)
      COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
      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
      NSEGS = 22
      PI = 3.1415927
      IF (INDX.EQ.0) THEN
         NMODUL = 1
         RBEG = RBIG0
      ENDIF
      IF (INDX.EQ.1) THEN
         NMODUL = NMODULL
         RBEG = RBIGL
      ENDIF
      IF (INDX.EQ.2) THEN
         NMODUL = NMODULG
         RBEG = RBIGG
      ENDIF
C
      IF (INDX.EQ.0.OR.INDX.EQ.1.OR.INDX.EQ.2) THEN
C
         CALL MOVER(0.,0,SROT,1,50)
         CALL MOVER(0.,0,ZRING,1,50)
         CALL MOVER(0.,0,ZNORM,1,50)
         CALL MOVER(0.,0,DRADIA,1,50)
         CALL MOVER(0.,0,DAXIAL,1,50)
         CALL MOVER(0.,0,RGAREA,1,50)
         CALL MOVER(0.,0,PRESUR,1,50)
         CALL MOVER(0.,0,FOUND,1,50)
         CALL MOVER(0.,0,FN10,1,50)
         CALL MOVER(0.,0,FN20,1,50)
         CALL MOVER(0, 0,IPTRNG,1,50)
         CALL MOVER(1, 0,IDISP,1,200)
         CALL MOVER(0, 0,JPREV,1,100)
         CALL MOVER(0, 0,JNODPV,1,100)
         CALL MOVER(0, 0,NPREV,1,50)
         CALL MOVER(0, 0,NODPRV,1,100)
         CALL MOVER(0, 0,NGRND,1,50)
         CALL MOVER(0, 0,NODGRD,1,100)
         FNOOD1 = 0.
         FNOOD2 = 0.
         FNOODL  = 0.
         PRESS1 = PRESS(ILOADX)
C
C  Single truss-core sandwich 22 segment module model...
C
         SWIDTH = 0.5*PITCH - BCROWN
         IF (SWIDTH.LT.0.0) THEN
            WRITE(IFILE4,'(/,A,/,A,/,A,1P,E12.4,/,A,1P,E12.4,/,A)')
     1      ' ***************** RUN ABORT ***********************',
     1      ' 0.5*PITCH -BCROWN is negative. Illegal configuration',
     1      ' Single module width,        PITCH =',PITCH,
     1      ' Width of truss core crown, BCROWN=',BCROWN,
     1      ' ***********************************************'
            CALL ERREX
         ENDIF
         TWOPHI = ATAN(HEIGHT/SWIDTH)
         PHI = TWOPHI/2.
         SPHI = SIN(PHI)
         CPHI = COS(PHI)
         ZVAL(1) = 0.
         DO 5 I = 1,NLAYRC
            LINDEX = LAYTYP(I,1)
            ZVAL(1) = ZVAL(1) + THICK(LINDEX)
    5    CONTINUE
         RADACU = RACUTE +ZVAL(1)
         RADOBT = ROBTUS +ZVAL(1)
         DWIDTH = RADACU/TAN(PHI)
         TWOALP = PI - TWOPHI
         ALPHA  = TWOALP/2.
         CWIDTH = RADOBT/TAN(ALPHA)
         GAMMA  = PI/2. - TWOPHI
         CTWOPH = COS(TWOPHI)
         STWOPH = SIN(TWOPHI)
         CGAMMA = COS(GAMMA)
         SGAMMA = SIN(GAMMA)
         AREA1  = DWIDTH*RADACU -ALPHA*RADACU**2
```

s in Fig. 1

t_{core} in Fig 8

See Figs. 10-13 for ϕ, α, γ

RADACU = R1 in Figs 10 & 8 & 1

RADOBT=R2 in Figs 10 & 8 & 1

d in Figs 10 & 1

c in Figs 10 & 1

Fig. 7 (top)

```
         AREA2  = CWIDTH*RADOBT -PHI*RADOBT**2
C
C        IMODEL = 1 is the elaborate model used for local buckling
         IMODEL = 1
         NSEGSM = 22                                 Fig. 7 (bottom)
         IF (IMODX.EQ.0) MODULL = NMODUL
         MODULS = MODULL
C
         IF (INDX.EQ.2) THEN
C        IMODEL = 2 is the simpler model used for general buckling
            IMODEL = 2
            NSEGSM = 6
C        The number of modules, "MODULG" should be such that the
C        model for general buckling subtends close to 90 degrees
C        of circumference...
            IF (IMODX.EQ.0) MODULG = (PI/1.89)*RADIUS/PITCH
            MODULS = MODULG
            IF (NSEGSM*MODULS.GT.288) THEN
               WRITE(IFILE4,'(/,A,/,A,2I5,/,A,/,A,)')
     1         ' *************** RUN ABORT ****************************',
     1         ' MODULS, MSEGS = NSEGSM*MODULS = ',MODULS, NSEGSM*MODULS,
     1         ' MSEGS = Too many segments. Maximum number is 288.',
     1         ' ****************************************************'
C23456789012345678901234567890123456789012345678901234567890123456789012
               CALL ERREX
            ENDIF
         ENDIF
C
         IF (IMODEL.EQ.1.AND.NMODUL.GT.13) THEN
           WRITE(IFILE4,'(/,A,/,A,I10,/,A,/,A)')
     1   ' *************** RUN ABORT ****************************',
     1   ' NMODUL = ',NMODUL,
     1   ' Too many modules. Maximum number of modules is 13.',
     1   ' ****************************************************'
           CALL ERREX
         ENDIF
C
         IF (INDX.EQ.1.OR.INDX.EQ.2) THEN
           C111MD = CSKIN1(2,2,1) - CSKIN1(1,2,1)**2/CSKIN1(1,1,1)
           C113MD = CSKIN1(2,2,3) - CSKIN1(1,2,3)**2/CSKIN1(1,1,3)
           C115MD = CSKIN1(2,2,5) - CSKIN1(1,2,5)**2/CSKIN1(1,1,5)
           TOTLNG = 2.*PI*RADIUS/PITCH
           ARC2 = 4.*RADOBT*2.*PHI
           ARC5 = 4.*RADACU*2.*ALPHA
           RATIO5 = C115MD/C111MD
           IF (IMODEL.EQ.1) THEN
             RATIO3 = C113MD/C111MD
             EANOOD = ENOODL*(AREA1 + AREA2) <- axial stiffness of "noodle"
           ELSE
             FL4 = DWIDTH - CWIDTH
             FL3 = DWIDTH + CWIDTH    Fig. 6 (bottom)
             EANOOD = ENOODL*(AREA1 + AREA2)
     1                +C115MD*(ARC1 + ARC2 + FL4 - FL3 -2.*DWIDTH)
             AREA = EANOOD/ENOODL
           ENDIF
         ENDIF
C
C   INDX   = 1 means SUBROUTINE BOSDEC generates LOCAL   buckling model
C   INDX   = 2 means SUBROUTINE BOSDEC generates GENERAL buckling model
C   IMODEL = model index used for buckling:
C            IMODEL = 1 means LOCAL   buckling
C            IMODEL = 2 means GENERAL buckling
C
C   MODEL2 = the model used for GENERAL buckling
C   IMODUL = the module number in a multi-module model
C   MODULS = number of modules in the model
C   RBEG   = horizontal radius to the beginning of the first module
C            in the multi-module model, where
C            1. for the LOCAL buckling model:
C               RBEG = RBIGL = RAVE -FLOAT(NMODULL)*PITCH/2.
C               in which RAVE = 100.*LENGTH*FACLEN/PI
C            2. for the GENERAL buckling model:
C               RBEG = RBIGG = RAVE -0.707*RADIUS
C               in which RAVE = 100.*LENGTH/PI
C            where
C            NMODULL = number of modules in the local buckling model
C            PITCH   = the module width, b, shown in Fig. 1
C            LENGTH  = the total length of the cylindrical shell
```

```
C            FACLEN  = fraction of LENGTH used for local buckling model
C            RADIUS  = radius of the cylindrical shell measured to
C                      the innermost face sheet.
C
C   BETA   = angle shown in Fig. 9
C   ISEGST = number of shell segments in all the previous modules
C   IRGST  = number of discrete rings in all the previous modules
C   NLAYRC = number of layers in the truss-core wrap,   Fig.8
C   NLAYRF = number of layers in each added face sheet, Fig.8
C   ZVAL   = distance from the leftmost shell segment surface
C            to the shell segment reference surface as we
C            "travel" along the shell segment
C   NLAYER(i) = number of layers in the ith shell segment
C               in the model for local buckling (IMODEL=1)
C   LAYTYP(i,j) = layer type (layer index) for layer i, segment j
C                 for IMODEL=1 (local buckling model)
C   NLAYR2(i) = number of layers in the ith shell segment
C               in the model for general buckling (IMODEL=2)
C   LAYTP2(i,j) = layer type (layer index) for layer i, segment j
C                 for IMODEL=2 (general buckling model)
C
C   R1(i),R2(i),Z1(i),Z2(i),RC(i),ZC(i) =
C                 (r,z) end points and center of curvature (rc,zc)
C                 for the ith shell segment as if the module were
C                 flat, as is shown in Fig.1 and in Figs. 10 - 13
C   RCURV1(i),RCURV2(i),ZCURV1(i),ZCURV2(i),RCURVC(i),ZCURVC(i) =
C                 (r,z) end points and center of curvature (rc,zc)
C                 for the ith shell segment as if the module were
C                 curved, as is shown in Fig. 2
C   RP1(i),RP2(i),ZP1(i),ZP2(i),RPC(i),ZPC(i) =
C                 (r,z) end points and center of curvature (rc,zc)
C                 for the ith shell segment as if the module were
C                 curved, as is shown in Fig.2, for module number
C                 IMODUL. (See Fig. 9)
C
C   IRING  = global discrete ring number
C   NTYPE  = type of callout used for the location of ring
C            attachment point to the ith shell segment.
C            NOTE: NTYPE(i) = 1 is always used in this project.
C            NTYPE(i) = 1 means that the location of the ring
C            attachment point to the shell segment is identified
C            by its nodal point number within that shell segment.
C            BIGBOSOR4 had to be modified to do this. Until now
C            BIGBOSOR4 only permitted callout locations to be
C            identified by either radial (r) or axial (z) numbers.
C            The modification to BIGBOSOR4 was needed to avoid input
C            errors in BIGBOSOR4 caused by lack of monotonicity
C            of r-values or z-values within a single shell segment.
C
C   IPTRNG = nodal point number of the attachment point of the
C            discrete ring to the ith shell segment
C   ZNORM  = distance from the ring attachment point to the ring
C            centroid, measured normal to the ith shell segment.
C            Fig. 2 shows locations of the centroids of the
C            eight discrete rings used in each module of the
C            local buckling model. A detailed view of the
C            relationship of the ring attachement point and the
C            ring centroid is given at the top of Fig. 7.
C
C   AREA1  = cross section area shown at the top of Fig. 7
C   AREA2  = cross section area shown at the bottom of Fig. 7
C
C            The total cross section area of a "noodle" is
C            AREA1 + AREA2. In the local buckling model there
C            are two discrete rings for each noodle, one
C            attached to curved shell segments such as
C            that shown at the top of Fig. 7 and the other
C            attached to curved shell segments such as
C            that shown at the bottom of Fig. 7. In the general
C            buckling model there is only one discrete ring
C            for each noodle, modeled as displayed in Fig. 6.
C
C   C111MD = axial stiffness of shell segments such as Segment 1
C            in Fig. 1. In the "huge torus" model this becomes
C            the circumferential stiffness, as follows:
C            C111MD = CSKIN1(2,2,1) - CSKIN1(1,2,1)**2/CSKIN1(1,1,1)
C
C   C113MD = axial stiffness of shell segments such as Segment 3
```

```
C            in Fig. 1. In the "huge torus" model this becomes
C            the circumferential stiffness, as follows:
C            C113MD = CSKIN1(2,2,3) - CSKIN1(1,2,3)**2/CSKIN1(1,1,3)
C
C   C115MD = axial stiffness of shell segments such as Segment 5
C            in Fig. 1. In the "huge torus" model this becomes
C            the circumferential stiffness, as follows:
C            C115MD = CSKIN1(2,2,5) - CSKIN1(1,2,5)**2/CSKIN1(1,1,5)
C
C           The "axial" stiffness of each of the 22 shell segments
C           in the elaborate local buckling module model consist of
C           various combinations of C111MD, C113MD, and C115MD. Once
C           the "axial" stiffness of each segment is determined,
C           the "axial" resultant in that segment can be computed
C           with use of the assumption that each segment of the
C           module and each discrete ring carries its share of
C           "axial load in proportion to its "axial" stiffness.
C           This model presumes that the cylindrical shell is
C           subjected to uniform end shortening, and that the
C           prebuckled state of the shell is a membrane state.
C
         DO 125 IMODUL = 1,MODULS
C
         FMODUL = IMODUL - 1
         BETA = FMODUL*PITCH/RADIUS   <-- β in Fig. 9
         SBETA = SIN(BETA)
         CBETA = COS(BETA)
         RSTART = RBEG   <-- RBEG in Fig. 9 & 1
         IF (IMODEL.EQ.1) THEN
            ISEGST = 22*(IMODUL - 1)
            IRGST = 8*(IMODUL - 1)
         ELSE
            ISEGST =  6*(IMODUL - 1)
            IRGST = 4*(IMODUL - 1)
         ENDIF
C
         IF (IMODEL.EQ.1) THEN
C
C       begin elaborate 22-segment model used for local buckling:
C       IMODEL = 1
C
         IF (IMODUL.EQ.1) THEN
            NSHAPE(1) = 2
            SROT(1) = 1.
            NSUR(1) = -1
            NRING(1) = 0
            IF (INDX.EQ.1) FOUND(1) = EFOUND
            IF (INTEXT.EQ.1) PRESUR(1) = PRESS1
            FN10(1) = PRESS1*RADIUS/2.   <-- meridional resultant in
                                             the "huge torus" model
            NNODES(1) = 11
            NLAYER(1) = NLAYRC + NLAYRF
            ZVAL(1) = 0.
            DO 1 I = 1,NLAYRC
               LINDEX = LAYTYP(I,1)
               ZVAL(1) = ZVAL(1) + THICK(LINDEX)
    1       CONTINUE
            ZRING(1) = 0.
            IRING(1+ISEGST) = 0
            NTYPE(1) = 1
            R1ORIG = RSTART +SWIDTH +CWIDTH
            R1(1) = R1ORIG
            Z1(1) = 0.
            R2(1) = RSTART +SWIDTH +BCROWN -CWIDTH
            Z2(1) = 0.
            RC(1) = RSTART
            ZC(1) = -RADIUS
C new "CURVED MODULE" section:
            RAD1 = RADIUS + Z1(1)
            RAD2 = RADIUS + Z2(1)
            BETA1 = (R1(1)-RSTART)/RAD1
            BETA2 = (R2(1)-RSTART)/RAD2
            RCURV1(1) = RSTART +RAD1*SIN(BETA1)
            RCURV2(1) = RSTART +RAD2*SIN(BETA2)
            RCURVC(1) = RC(1)
            ZCURV1(1) = Z1(1) -RAD1*(1. -COS(BETA1))
            ZCURV2(1) = Z2(1) -RAD2*(1. -COS(BETA2))
            ZCURVC(1) = ZC(1)
            RAD0 = RCURV1(1)
```

Table 13a, p.6 of 13

```
            ZED0 = ZCURV1(1)
 C end "CURVED MODULE" section
         ENDIF
 C       End of "IMODUL.EQ.1" condition
 C
         DRAD1 = RCURV1(1) - RAD0
         DZED1 = ZCURV1(1) - ZED0
         DRAD2 = RCURV2(1) - RAD0
         DZED2 = ZCURV2(1) - ZED0
         DRADC = RCURVC(1) - RAD0
         DZEDC = ZCURVC(1) - ZED0
         RP1(1) = RAD0 +RADIUS*SBETA +DRAD1*CBETA +DZED1*SBETA
         ZP1(1) = ZED0 -RADIUS*(1.-CBETA) -DRAD1*SBETA +DZED1*CBETA
         RP2(1) = RAD0 +RADIUS*SBETA +DRAD2*CBETA +DZED2*SBETA
         ZP2(1) = ZED0 -RADIUS*(1.-CBETA) -DRAD2*SBETA +DZED2*CBETA
         RPC(1) = RAD0 +RADIUS*SBETA +DRADC*CBETA +DZEDC*SBETA
         ZPC(1) = ZED0 -RADIUS*(1.-CBETA) -DRADC*SBETA +DZEDC*CBETA
         IF (INDX.EQ.1.AND.IMODUL.EQ.1) THEN
            RL1(1) = RP1(1)
            ZL1(1) = ZP1(1)
            RL2(1) = RP2(1)
            ZL2(1) = ZP2(1)
            RLC(1) = RPC(1)
            ZLC(1) = ZPC(1)
         ENDIF
 C
         IF (IMODUL.EQ.1) THEN
            NSHAPE(2) = 2
```

(many lines skipped to save space) ← See ...GENOPT/case/trusscomp/bosdec.trusscomp

```
 C      end of elaborate 22-segment model used for local buckling,
 C      IMODEL = 1
 C
         ELSE
 C
 C begin simplified model used for general buckling, IMODEL = 2:
 C
         IF (IMODUL.EQ.1) THEN
            NSHAPE(1) = 2
            SROT(1) = 1.
            NSUR(1) = -1
            NRING(1) = 1
            RGAREA(1) = AREA
            IF (INDX.EQ.1) FOUND(1) = EFOUND
            IF (INTEXT.EQ.1) PRESUR(1) = PRESS1
            FN10(1) = PRESS1*RADIUS/2.
            NNODES(1) = 11
            NLAYR2(1) = NLAYRC + NLAYRF
            LAYR21 = NLAYR2(1)
            DO 26 I = 1,LAYR21
               LAYTP2(I,1) = LAYTYP(I,1)
   26       CONTINUE
            ZVAL(1) = 0.
            DO 28 I = 1,NLAYRC
               LINDEX = LAYTYP(I,1)
               ZVAL(1) = ZVAL(1) + THICK(LINDEX)
   28       CONTINUE
            ZRING(1) = 0.
            NTYPE(1) = 1
            IPTRNG(1) = NNODES(1)
            R1(1) = RSTART +SWIDTH
            Z1(1) = 0.
            R2(1) = RSTART +SWIDTH +BCROWN
            Z2(1) = 0.
            RC(1) = RSTART
            ZC(1) = -RADIUS
C new "CURVED MODULE" section:
            RAD1 = RADIUS + Z1(1)
            RAD2 = RADIUS + Z2(1)
            BETA1 = (R1(1)-RSTART)/RAD1
            BETA2 = (R2(1)-RSTART)/RAD2
            RCURV1(1) = RSTART +RAD1*SIN(BETA1)
            RCURV2(1) = RSTART +RAD2*SIN(BETA2)
            RCURVC(1) = RC(1)
            ZCURV1(1) = Z1(1) -RAD1*(1. -COS(BETA1))
            ZCURV2(1) = Z2(1) -RAD2*(1. -COS(BETA2))
            ZCURVC(1) = ZC(1)
```

```
            RAD0 = RCURV1(1)
            ZED0 = ZCURV1(1)
C end "CURVED MODULE" section
         ENDIF
C       End of "IMODUL.EQ.1" condition
C
         IRING(1+ISEGST) = 1 + IRGST
         DRAD1 = RCURV1(1) - RAD0
         DZED1 = ZCURV1(1) - ZED0
         DRAD2 = RCURV2(1) - RAD0
         DZED2 = ZCURV2(1) - ZED0
         DRADC = RCURVC(1) - RAD0
         DZEDC = ZCURVC(1) - ZED0
         RP1(1) = RAD0 +RADIUS*SBETA +DRAD1*CBETA +DZED1*SBETA
         ZP1(1) = ZED0 -RADIUS*(1.-CBETA) -DRAD1*SBETA +DZED1*CBETA
         RP2(1) = RAD0 +RADIUS*SBETA +DRAD2*CBETA +DZED2*SBETA
         ZP2(1) = ZED0 -RADIUS*(1.-CBETA) -DRAD2*SBETA +DZED2*CBETA
         RPC(1) = RAD0 +RADIUS*SBETA +DRADC*CBETA +DZEDC*SBETA
         ZPC(1) = ZED0 -RADIUS*(1.-CBETA) -DRADC*SBETA +DZEDC*CBETA
C
         IF (IMODUL.EQ.1) THEN
            NSHAPE(2) = 2

   (lines skipped to save space)

C
C end of simplified model used for general buckling, IMODEL = 2
C
         ENDIF
C       End of "IF (MODEL.EQ.1)....ELSE ...ENDIF" condition
C
C234567890123456789012345678901234567890123456789012345678901234567890l2
C
C  Next, the "axial" resultant carried by each shell segment
C  and by each discrete ring is computed from the assumption
C  that the cylindrical shell is subjected to uniform end
C  shortening, from the assumption that the prebuckled state
C  is a membrane state, and from the knowledge of the "axial"
C  stiffness of each shell segment and of each discrete ring ("noodle").
C  The "axial" resultants in the cylindrical shell become the
C  hoop (circumferential) resultants in the "huge torus" model
C  of the cylindrical shell. In the BIGBOSOR4 model these hoop
C  resultants are called "FN20".
C
         IF (IMODUL.EQ.1.AND.(INDX.EQ.1.OR.INDX.EQ.2)) THEN
C  Get huge torus hoop resultants carried by each segment
           IF (IMODEL.EQ.1) THEN
             SLANT7 = SQRT((R2(7) - R1(7))**2 +(Z2(7) - Z1(7))**2)
             SLANT4 = SQRT((R2(4) - R1(4))**2 +(Z2(4) - Z1(4))**2)
             DR18 = R2(18) - R1(18)
             DR6  = R2(6)  - R1(6)
             DR3  = R2(3)  - R1(3)
C234567890123456789012345678901234567890123456789012345678901234567890l2
             DENOM=(RATIO5*(ARC2 +ARC5) +2.*DR18 +2.*DR6 +4.*DR3*RATIO3
     1             +RATIO5*4.*(SLANT7 +SLANT4) +4.*EANOOD/C111MD)*TOTLNG
           ELSE
             SLANT  = SQRT((R2(3) - R1(3))**2 +(Z2(3) - Z1(3))**2)
             DR1 = BCROWN
             DR2 = PITCH - BCROWN
             DENOM=(2.*DR1 +2.*DR2
     1             +RATIO5*4.*SLANT +4.*EANOOD/C111MD)*TOTLNG
           ENDIF
C
C   FNX1 = "axial" resultant in Segment 1 of the model shown in Fig. 1
C   FNX5 = "axial" resultant in Segment 5 of the model shown in Fig. 1
C
           FNX1 = PX(ILOADX)/DENOM
           FNX5 = RATIO5*FNX1
C
           IF (IMODEL.EQ.1) THEN
C
C       This branch is for the elaborate 22-segment module model.
C       See Fig. 1 for the arrangement of each of the 22 segments
C       in the elaborate module model used for local buckling.
C
              FNX3 = RATIO3*FNX1
              FNOOD1 = ENOODL*AREA1*FNX1/C111MD
              FNOOD2 = ENOODL*AREA2*FNX1/C111MD
```

Table 13a, p. 8 of 13

FN20 = circumferential resultant in the "huge torus" model

```
             FN20(1) = FNX1
             FN20(2) = FNX5
             FN20(3) = FNX3
             FN20(4) = FNX5
             FN20(5) = FNX5
             FN20(6) = FNX1
             FN20(7) = 2.*FNX5
             FN20(8) = FNX3
             FN20(9) = FNX5
             FN20(10) = FNX5
             FN20(11) = FNX5
             FN20(12) = 2.*FNX5
             FN20(13) = FNX5
             FN20(14) = FNX5
             FN20(15) = FNX1
             FN20(16) = FNX5
             FN20(17) = FNX3
             FN20(18) = FNX1
             FN20(19) = FNX5
             FN20(20) = FNX5
             FN20(21) = FNX5
             FN20(22) = FNX3
C
C  The strain components, EPSX, EPSY1, EPSY3, EPSY5 are used to
C  compute the stresses in each layer of each shell segment. The
C  following definitions apply:
C
C   FNX1   = "axial" resultant in shell segment 1 in Fig. 1
C   C111MD = "axial" stiffness of shell segment 1 in Fig. 1
C   FN10   = "circumferential" resultant in each face sheet.
C   EPSX   = "axial" strain (assumed to be the same for all shell
C            segments)
C   EPSY1  = "circumferential" strain in "segment 1 type"
C            shell segments
C   EPSY3  = "circumferential" strain in "segment 3 type"
C            shell segments
C   EPSY5  = "circumferential" strain in "segment 5 type"
C            shell segments
C
C         Remember, in the "huge torus" model of the cylindrical
C         shell "axial" is actually the circumferential component,
C         and "circumferential" is actually the meridional component.
C
            EPSX = FNX1/C111MD
            EPSY1 = FN10(1) -EPSX*CSKIN1(1,2,1)/CSKIN1(2,2,1)
            EPSY3 = FN10(3) -EPSX*CSKIN1(1,2,3)/CSKIN1(2,2,3)
            EPSY5 = -EPSX*CSKIN1(1,2,5)/CSKIN1(2,2,5)
C
          ELSE
C
C      This branch is for the simplified 6-segment module model.
C      See Fig. 6, top, for the arrangement of each of the 6 segments
C      in the module model for general buckling.
C
            FN20(1) = FNX1
            FN20(2) = FNX1
            FN20(3) = 2.*FNX5
            FN20(4) = 2.*FNX5
            FN20(5) = FNX1
            FN20(6) = FNX1
            FNOODL = EANOOD*FNX1/C111MD
          ENDIF
        ENDIF
C
C2345678901234567890123456789012345678901234567890123456789012
C
C  Next, we generate a valid input data file, *.ALL, for BIGBOSOR4
C
C  Global input before segment data...
        IF (IMODUL.EQ.1) THEN
          MSEGS = MODULS*NSEGSM
          IF (INDX.EQ.0.OR.INDX.EQ.1) WRITE(IFIL14,'(A)')
     1    ' local buckling from single module model (INDIC=4)'
          IF (INDX.EQ.2) WRITE(IFIL14,'(A)')
     1    ' general buckling from multiple module model (INDIC=4)'
          WRITE(IFIL14,'(A,I3,A)')
     1    ' 4, 1, 0, 0, ',MSEGS,' $ INDIC,NPRT,ISTRESS,IPRE,NSEG'
        ENDIF
```

← circumferential in the "huge" torus model

```
C
C  Segment data...
         MAXTYP = 0
         MAXMAT = 0
C
C  First, provide the input for each of the NSEGSM shell segments
C  in the next module...
C
         DO 100 ISEG = 1,NSEGSM
         I = ISEG
         WRITE(IFIL14,'(A,4I6)')' H    $ Segment number ',I,I,I,I
         WRITE(IFIL14,'(I4,A,I3,A)') NNODES(ISEG),', 3, ',NSHAPE(ISEG),
     1 ' $ NMESH,NTYPEH,NSHAPE'
         WRITE(IFIL14,'(1P,4E14.6,A)') RP1(I),ZP1(I),RP2(I),ZP2(I),
     1 ' $ R1,Z1,R2,Z2'
         IF (NSHAPE(ISEG).EQ.2) THEN
          WRITE(IFIL14,'(1P,3E14.6,A)') RPC(ISEG),ZPC(ISEG),SROT(ISEG),
     1  ' $ RC,ZC,SROT'
         ENDIF
C23456789012345678901234567890123456789012345678901234567890123456789012
         WRITE(IFIL14,'(A,1P,E14.6,A)')' 0, 3, ',ZVAL(ISEG),
     1 ' $ IMP,NTYPEZ,ZVAL'
         WRITE(IFIL14,'(A)')' N  $ do not print r(s), etc.'
         IF (IMODEL.EQ.1) THEN
            NRLOAD = 8*MODULS
         ELSE
            NRLOAD = 4*MODULS
         ENDIF
         IF (NRING(ISEG).EQ.0) THEN
           WRITE(IFIL14,'(I5,1PE14.6,A,2I4,A)') NRING(ISEG),FOUND(ISEG),
     1    ', 2, ',NRLOAD,NTYPE(ISEG),' $ NRINGS,K,NSTRES,NRLOAD,NTYPE'
         ELSE
           WRITE(IFIL14,'(3I5,A)') NRING(ISEG),NTYPE(ISEG),
     1     IPTRNG(ISEG),', 2 $ NRINGS,NTYPE,IPOINT,NTYPER'
           RAREA = RGAREA(ISEG)
C
C    You may want to modify the values for discrete ring bending
C    moments of inertia, FMOMTS, FMOMTN, FMOMSN, and the torsional
C    stiffness, GJ. I just assumed these formulas without much thought.
C
           FMOMTS = RAREA**2/8.
           FMOMTN = RAREA**2/8.
           FMOMSN = 0.
           ZMERID = 0.
           GJ     = FMOMTS*ENOODL/2.6
           RGWGT  = RAREA*DNOODL
           WRITE(IFIL14,'(1P,3E14.6,A)') ENOODL,RAREA,FMOMTS,
     1 ' $ ERING,AREA,Is'
           WRITE(IFIL14,'(1P,3E14.6,A)') FMOMTN,FMOMSN,ZNORM(ISEG),
     1 ' $ In,Isn,ZNORM'
           WRITE(IFIL14,'(1P,3E14.6,A)') ZMERID,GJ,RGWGT,
     1 ' $ ZMERID,GJ,RWGT'
           WRITE(IFIL14,'(1P,E14.6,A,2I4,A)') FOUND(ISEG),
     1    ', 2, ',NRLOAD,NTYPE(ISEG),' $ K,NSTRES,NRLOAD,NTYPE'
         ENDIF
C
         IF (NTYPE(ISEG).EQ.1)
     1   WRITE(IFIL14,'(A,I4,A)')' 1, ',NNODES(ISEG),
     1 ' $ IPOINT(1),IPOINT(2)'
         IF (NTYPE(ISEG).EQ.2)
     1   WRITE(IFIL14,'(1P,2E14.6,A)') ZP1(ISEG),ZP2(ISEG),
     1 ' $ Z1(ISEG),Z2(ISEG)'
         IF (NTYPE(ISEG).EQ.3)
     1   WRITE(IFIL14,'(1P,2E14.6,A)') RP1(ISEG),RP2(ISEG),
     1 ' $ R1(ISEG),R2(ISEG)'
         WRITE(IFIL14,'(1P,4E14.6,A)') FN10(I),FN10(I),FN20(I),FN20(I),
     1 ' $ FN10,FN10,FN20,FN20'
C
C23456789012345678901234567890123456789012345678901234567890123456789012
         IF (ISEG.EQ.1.AND.IMODUL.EQ.1) THEN
C
C  Note that in the BIGBOSOR4 INDIC = 4 branch the prebuckling forces
C  in ALL the rings in the multi-segment model are read in with the
C  data for the first shell segment.
C
            WRITE(IFIL14,'((4I5))') (I,I=1,NRLOAD)
C
            IF (IMODEL.EQ.1) THEN
```

```
C
C   Elaborate 22-segment local buckling module model:
C   FNOOD1 and FNOOD2 are the prebuckling forces in the discrete rings
C   ("noodles") with cross section areas, AREA1 and AREA2, respectively.
C   (Fig. 7)
C
              DO 90 KMODUL = 1,MODULS
                WRITE(IFIL14,'(1P4E14.6,A)') FNOOD2,FNOOD1,FNOOD1,FNOOD2,
     1      ' $ FNOOD2,FNOOD1,FNOOD1,FNOOD2'
                WRITE(IFIL14,'(1P4E14.6,A)') FNOOD1,FNOOD2,FNOOD2,FNOOD1,
     1      ' $ FNOOD1,FNOOD2,FNOOD2,FNOOD1'
   90         CONTINUE
            ELSE
C
C   Simplified 6-segment general buckling module model:
C   FNOODL is the prebuckling force in the discrete ring ("noodle")
C   with cross section area, AREA1 + AREA2. (Fig. 6)
C
              DO 92 KMODUL = 1,MODULS
                WRITE(IFIL14,'(1P4E14.6,A)') FNOODL,FNOODL,FNOODL,FNOODL,
     1      ' $ FNOODL,FNOODL,FNOODL,FNOODL'
   92         CONTINUE
            ENDIF
          ENDIF
C       End of (ISEG.EQ.1.AND.IMODUL.EQ.1) condition
C23456789012345678901234567890123456789012345678901234567890123456789012
          WRITE(IFIL14,'(A)')' N  $ do not print prestresses'
C
          IF (IMODEL.EQ.1) THEN
             NLAY = NLAYER(ISEG)
          ELSE
             NLAY = NLAYR2(ISEG)
          ENDIF
          WRITE(IFIL14,'(A,I3,A)')' 4, ',NLAY,' $ NWALL,NLAYER'
          MAXMT1 = MAXMAT
          MAXTP1 = MAXTYP
          DO 95 ILAYER = 1,NLAY
             IF (IMODEL.EQ.1) THEN
                LTYPE = LAYTYP(ILAYER,ISEG)
             ELSE
                LTYPE = LAYTP2(ILAYER,ISEG)
             ENDIF
             WRITE(IFIL14,'(I3,A)') LTYPE,' $ layer index'
             IF (LTYPE.GT.MAXTYP) THEN
                WRITE(IFIL14,'(A)')' Y  $ is this a new layer type?'
                IF (ANGLE(LTYPE).GT.90.0.OR.ANGLE(LTYPE).LT.-90.0) THEN
                  WRITE(IFILE4,'(/,A,/,A,/,A,/,A)')
     1          ' ****************** RUN ABORT **********************',
     1          ' Layup angle out of range. Layup angle must be in the',
     1          ' range, -90.0 .LE. (layup angle) .LE. +90.0 degrees.',
     1          ' ****************************************************'
                  CALL ERREX
                ENDIF
                IF (ANGLE(LTYPE).GT.0.0) ANGLE2 = -(90.0 -ANGLE(LTYPE))
                IF (ANGLE(LTYPE).LT.0.0) ANGLE2 = 90.0 + ANGLE(LTYPE)
                IF (ANGLE(LTYPE).EQ.0.0) ANGLE2 = 90.0
                IF (ANGLE(LTYPE).EQ.90.0) ANGLE2 = 0.0
                WRITE(IFIL14,'(1P2E14.6,I3,A)') THICK(LTYPE),
     1          ANGLE2,MATTYP(LTYPE),' $ thickness,angle,material'
                MAXTYP = MAX(MAXTYP,LTYPE)
                MAXMAT = MAX(MAXMAT,MATTYP(LTYPE))
             ELSE
                WRITE(IFIL14,'(A)')' N  $ is this a new layer type?'
             ENDIF
   95     CONTINUE
C
          IF (MAXTP1.LT.MAXTYP.AND.MAXMAT.LE.MAXMT1) THEN
             WRITE(IFIL14,'(A)')' N  $ Is this material new?'
          ENDIF
          IF (MAXMAT.GT.MAXMT1) THEN
             MAXMTP = MAXMT1 + 1
             DO 97 IMATL = MAXMTP,MAXMAT
                WRITE(IFIL14,'(A)')' Y  $ Is this material new?'
                WRITE(IFIL14,'(1P4E14.6,A)') EMOD1(IMATL),EMOD2(IMATL),
     1          G12(IMATL),NU(IMATL),' $ E1,E2,G12,NU'
                WRITE(IFIL14,'(1P4E14.6,A)') ALPHA1(IMATL),ALPHA2(IMATL),
     1          TEMCUR(IMATL),DENSTY(IMATL),' $ A1,A2,TEMPTUR,DENS'
                IF (IMATL.EQ.1) THEN
```

```
                WRITE(IFIL14,'(1P3E14.6,A)') STRM1A(ILOADX,1),
     1          STRM1A(ILOADX,2),STRM1A(ILOADX,3),' $ S(1),S(2),S(3)'
                WRITE(IFIL14,'(1P2E14.6,A)') STRM1A(ILOADX,4),
     1          STRM1A(ILOADX,5),' $ S(4),S(5)'
                GO TO 96
              ENDIF
              IF (IMATL.EQ.2) THEN
                WRITE(IFIL14,'(1P3E14.6,A)') STRM2A(ILOADX,1),
     1          STRM2A(ILOADX,2),STRM2A(ILOADX,3),' $ S(1),S(2),S(3)'
                WRITE(IFIL14,'(1P2E14.6,A)') STRM2A(ILOADX,4),
     1          STRM2A(ILOADX,5),' $ S(4),S(5)'
                GO TO 96
              ENDIF
              WRITE(IFILE4,'(/,A,/,A,/,A,/,A)')
     1       ' ****************** RUN ABORT ***********************',
     1       ' Material type out of range. Only 2 material types are',
     1       ' permitted in this model.',
     1       ' ***************************************************'
              CALL ERREX
   96       CONTINUE
   97       CONTINUE
C23456789012345678901234567890123456789012345678901234567890123456789012
        ENDIF
        WRITE(IFIL14,'(A)')' 0  $ no additional smeared stiffeners'
        WRITE(IFIL14,'(A)')' Y  $ do you want output for all nodes?'
        WRITE(IFIL14,'(A)')' N  $ do you want to print out Cij?'
        WRITE(IFIL14,'(A)')' N  $ do you want to print out loads?'
C
  100   CONTINUE
C
  125   CONTINUE
C
        WRITE(IFIL14,'(A)')' H  $ GLOBAL DATA BEGINS...'
        WRITE(IFIL14,'(A)')' 0  $ NLAST'
        WRITE(IFIL14,'(A)')' N  $ any expanded plots?'
        NMINB = N0B
        INCRB = 100
        NVEC  = 1
        WRITE(IFIL14,'(5I6,A)') N0B,NMINB,NMAXB,INCRB,NVEC,
     1 '  $ N0B,NMINB,NMAXB,INCRB,NVEC'
        WRITE(IFIL14,'(A)')' H  $ CONSTRAINT CONDITIONS FOLLOW....'
        WRITE(IFIL14,'(I6,A)') MSEGS,'  $ how many segments?'
C
C   Next, generate the BIGBOSOR4 input for all the constraint
C   conditions (connections to ground and segment junctions)
C   in the multi-module model.
C
        DO 400 IMODUL = 1,MODULS
C
        IF (IMODEL.EQ.1) THEN
C
C Do the input for the elaborate 22-segment local buckling
C model in this branch.
C
C Begin IMODEL = 1 logic for connections to ground, junctions
C
        NGRND(1) = 0
        IF (IMODUL.EQ.1) THEN
          NGRND(1) = 1
          NODGRD(1,1) = 1
          IF (INDX.EQ.1) IDISP(1,1) = 1
          IF (INDX.EQ.2) IDISP(1,1) = 0
          IDISP(1,2) = 0
        ELSE
          NPREV(1) = 1
          NODPRV(1,1) = 1
          JPREV(1,1) = 8 + NSEGSM*(IMODUL - 2)
          JNODPV(1,1) = NNODES(8)
        ENDIF
C
        NPREV(2) = 1
        NODPRV(2,1) = 1
        JPREV(2,1) = 1 + NSEGSM*(IMODUL - 1)
        JNODPV(2,1) = NNODES(1)
C
   (lines skipped to save space)
C
C End   IMODEL = 1 logic for connections to ground, junctions
```

```
C
        ELSE
C
C Do the input for the simpler 6-segment local buckling
C model in this branch.
C
C Begin IMODEL = 2 logic for connections to ground, junctions
C
         NGRND(1) = 0
         IF (IMODUL.EQ.1) THEN
           NGRND(1) = 1
           NODGRD(1,1) = 1
C          IDISP(1,1) = 0
C          IDISP(1,2) = 0
           IDISP(1,3) = 0
           IDISP(1,4) = 0
         ELSE
           NPREV(1) = 1
           NODPRV(1,1) = 1
           JPREV(1,1) = 2 + NSEGSM*(IMODUL - 2)
           JNODPV(1,1) = NNODES(2)
         ENDIF
C
   (lines skipped to save space)

         JPREV(6,1) = 3 + NSEGSM*(IMODUL - 1)
         JNODPV(6,1) = NNODES(3)
         NODPRV(6,2) = NNODES(6)
         JPREV(6,2) = 4 + NSEGSM*(IMODUL - 1)
         JNODPV(6,2) = NNODES(4)
C
         NPREV(4) = 1
         NODPRV(4,1) = 1
         JPREV(4,1) = 2 + NSEGSM*(IMODUL - 1)
         JNODPV(4,1) = NNODES(2)
C
C End IMODEL = 2 logic for connections to ground, junctions
C
         ENDIF
C
C End of IMODEL = 1 and IMODEL = 2 logic for ground,junctions
C
         DO 300 I = 1,NSEGSM
C
         WRITE(IFIL14,'(A,4I6)')
     1 ' H  $ CONSTRAINT CONDITIONS FOR SEGMENT ',I,I,I,I
         WRITE(IFIL14,'(A)')' 0  $ number of poles'
         WRITE(IFIL14,'(I3,A)') NGRND(I),' $ connect to ground'
         IF (NGRND(I).GT.0) THEN
            NGRNDI = NGRND(I)
            DO 150 J = 1,NGRNDI
               WRITE(IFIL14,'(I3,A)') NODGRD(I,J),' $ node to ground'
               WRITE(IFIL14,'(4I3,A)')
     1          IDISP(I,1),IDISP(I,2),IDISP(I,3),IDISP(I,4),
     1         ' $ IUSTAR,IVSTAR,IWSTAR,ICHI'
               WRITE(IFIL14,'(A)')' 0.,   0.   $ D1,D2'
               WRITE(IFIL14,'(A)')' Y  $ is constraint same for buck.?'
  150       CONTINUE
         ENDIF
C234567890123456789012345678901234567890123456789012345678901234567890012
C
         IF (I.LE.1.AND.IMODUL.EQ.1) THEN
             WRITE(IFIL14,'(A)')' N  $ joined to previous segments?'
         ELSE
             WRITE(IFIL14,'(A)')' Y  $ joined to previous segments?'
             WRITE(IFIL14,'(I3,A)') NPREV(I),' $ connects to prev.segs'
             NPREVI = NPREV(I)
             DO 200 J = 1,NPREVI
               WRITE(IFIL14,'(I3,A)') NODPRV(I,J),' $ node current seg'
               WRITE(IFIL14,'(I3,A)') JPREV(I,J),' $ prev.segment no.'
               WRITE(IFIL14,'(I3,A)') JNODPV(I,J),' $ node in prev.seg.'
               WRITE(IFIL14,'(A)')' 1, 1, 1, 1 $ IU,IV,IW,ICHI'
               WRITE(IFIL14,'(A)')' 0., 0.   $ D1,D2'
               WRITE(IFIL14,'(A)')' Y  $ is constraint same for buck.?'
C234567890123456789012345678901234567890123456789012345678901234567890012
  200        CONTINUE
         ENDIF
C
```

```
  300     CONTINUE
C
  400     CONTINUE
C
          WRITE(IFIL14,'(A)')' N  $ are rigid body motions possible?'
C
          DO 450 ISEG = 1,MSEGS
             WRITE(IFIL14,'(A)')' Y  $ do you want to list seg. output?'
  450     CONTINUE
          WRITE(IFIL14,'(A)')' Y  $ do you want to list ring forces?'
C
       ENDIF
C    End of "INDX.EQ.0.OR.INDX.EQ.1.OR.INDX.EQ.2" condition
C
       RETURN
       END
```

End of bosdec.trusscomp

"bosdec" = "bosor deck"

Table 13b from bosdec.trusscomp (3 pages)

```
From bosdec.trusscomp: Connections to ground and some junctions
C   Next, generate the BIGBOSOR4 input for all the constraint
C   conditions (connections to ground and segment junctions)
C   in the multi-module model.
C
C   The connections to ground (NGRND(i) > 0), where
C   i = the shell segment number, are governed by the
C   array, IDISP(i,j), where j is an index for which displacement
C   component is constrained or not constrained.
C
C   IDISP(i,j) = 1 means that the jth displacement component is
C                  constrained to be zero.
C   IDISP(i,j) = 0 means that the jth displacement component is
C                  free
C
C   In BIGBOSOR4 [10] there are 4 displacement components:
C
C   IDISP(i,1) refers to USTAR, the radial (horizontal) displacement
C   IDISP(i,2) refers to VSTAR, the circumferential displacement
C   IDISP(i,3) refers to WSTAR, the axial (vertical) displacement
C   IDISP(i,4) refers to CHI, the meridional rotation
C
C   The entire array, IDISP(i,j), i = 1,NSEGS, j = 1,4,
C   is initially set equal to unity when SUBROUTINE BOSDEC is
C   called from SUBROUTINE BEHX1 (local buckling:   IMODEL = 1) and
C   called from SUBROUTINE BEHX2 (general buckling: IMODEL = 2). In
C   the following code IDISP(i,j) is reset to zero in certain of
C   the shell segments:
C
C   For LOCAL   buckling (IMODEL = 1) in Segments 1, 8, 15, and 21
C   For GENERAL buckling (IMODEL = 2) in Segments 1 and 2 and 5
C
C   The following code from bosdec.trusscomp is abridged so that
C   only the shell segments with constraints to ground are included.
C
        DO 400 IMODUL = 1,MODULS
C
        IF (IMODEL.EQ.1) THEN
C
C Do the input for the elaborate 22-segment local buckling
C model in this branch.
C
C Begin IMODEL = 1 logic for connections to ground, junctions
C
        NGRND(1) = 0
        IF (IMODUL.EQ.1) THEN
          NGRND(1) = 1
          NODGRD(1,1) = 1
          IDISP(1,2) = 0
        ELSE
          NPREV(1) = 1
          NODPRV(1,1) = 1
          JPREV(1,1) = 8 + NSEGSM*(IMODUL - 2)
          JNODPV(1,1) = NNODES(8)
        ENDIF
C
  (lines skipped to save space)
C
        NGRND(8) = 0
        IF (IMODUL.EQ.MODULS) THEN
          NGRND(8) = 1
          NODGRD(8,1) = NNODES(8)
          IDISP(8,3) = 0
          IDISP(8,4) = 0
        ENDIF
        NPREV(8) = 1
        NODPRV(8,1) = 1
        JPREV(8,1) = 6 + NSEGSM*(IMODUL - 1)
        JNODPV(8,1) = NNODES(6)
C
  (lines skipped to save space)
C
        NGRND(15) = 0
        IF (IMODUL.EQ.1) THEN
          NGRND(15) = 1
          NODGRD(15,1) = NNODES(15)
          IDISP(15,2) = 0
```

```
            NPREV(15) = 1
          ELSE
            NPREV(15) = 2
            NODPRV(15,2) = NNODES(15)
            JPREV(15,2) = 21 +NSEGSM*(IMODUL - 2)
            JNODPV(15,2) = NNODES(21)
          ENDIF
          NODPRV(15,1) = 1
          JPREV(15,1) = 14 + NSEGSM*(IMODUL - 1)
          JNODPV(15,1) = NNODES(14)
C
  (lines skipped to save space)
C
          NGRND(21) = 0
          IF (IMODUL.EQ.MODULS) THEN
            NGRND(21) = 1
            NODGRD(21,1) = NNODES(21)
            IDISP(21,3) = 0
            IDISP(21,4) = 0
          ENDIF
          NPREV(21) = 1
          NODPRV(21,1) = 1
          JPREV(21,1) = 12 + NSEGSM*(IMODUL - 1)
          JNODPV(21,1) = NNODES(12)
C
  (lines skipped to save space)
C
C End   IMODEL = 1 logic for connections to ground, junctions
C
          ELSE
C
C Do the input for the simpler 6-segment local buckling
C model in this branch.
C
C Begin IMODEL = 2 logic for connections to ground, junctions
C
          NGRND(1) = 0
          IF (IMODUL.EQ.1) THEN
            NGRND(1) = 1
            NODGRD(1,1) = 1
            IDISP(1,3) = 0
            IDISP(1,4) = 0
          ELSE
            NPREV(1) = 1
            NODPRV(1,1) = 1
            JPREV(1,1) = 2 + NSEGSM*(IMODUL - 2)
            JNODPV(1,1) = NNODES(2)
          ENDIF
C
          NGRND(2) = 0
          IF (IMODUL.EQ.MODULS) THEN
            NGRND(2) = 1
            NODGRD(2,1) = NNODES(2)
            IDISP(2,1) = 0
            IDISP(2,4) = 0
          ENDIF
          NPREV(2) = 1
          NODPRV(2,1) = 1
          JPREV(2,1) = 1 + NSEGSM*(IMODUL - 1)
          JNODPV(2,1) = NNODES(1)
C
          NGRND(5) = 0
          IF (IMODUL.EQ.1) THEN
            NGRND(5) = 1
            NODGRD(5,1) = NNODES(5)
            IDISP(5,3) = 0
            IDISP(5,4) = 0
            NPREV(5) = 1
          ELSE
            NPREV(5) = 2
            NODPRV(5,2) = NNODES(5)
            JPREV(5,2) = 4 +NSEGSM*(IMODUL - 2)
            JNODPV(5,2) = NNODES(4)
          ENDIF
          NODPRV(5,1) = 1
          JPREV(5,1) = 3 + NSEGSM*(IMODUL - 1)
          JNODPV(5,1) = NNODES(3)
C
```

Table 13b (p. 3 of 3)

```
  (lines skipped to save space)
C
C End IMODEL = 2 logic for connections to ground, junctions
C
         ENDIF
C
C End of IMODEL = 1 and IMODEL = 2 logic for ground,junctions
```

Table 14 input for "BEGIN" test. BEG (4 pages)

```
    n           $ Do you want a tutorial session and tutorial output?
     96         $ length of the cylindrical shell: LENGTH
 0.1000000      $ fraction of LENGTH for local buckling: FACLEN
  48.00000      $ radius of the cylindrical shell: RADIUS
  5.192400      $ circumferential width of a single module: PITCH
  2.339900      $ circumferential width of the trapezoid crown: BCROWN
  1.417200      $ height of the truss-core sandwich: HEIGHT
 0.1000000      $ local radius from base to side of trapezoidal tool: RACUTE
 0.1000000      $ local radius from side to crown of trapezoidal tool: ROBTUS
 0.1851110E+08 $ axial modulus of a corner "noodle": ENOODL
 0.5700000E-01 $ weight density of the "noodle" material: DNOODL
       0        $ elastic foam "Winkler" foundation stiffness: EFOUND
       0        $ elastic "noodle" Winkler foundation modulus: EFNOOD
       1        $ Number IEMOD1  of rows in the array  EMOD1: IEMOD1
 0.1851110E+08 $ elastic modulus in the fiber direction: EMOD1( 1)
  1640000.      $ elastic modulus transverse to fibers: EMOD2( 1)
  870600.0      $ in-plane shear modulus: G12( 1)
  870600.0      $ out-of-plane x-z shear modulus: G13( 1)
  870600.0      $ out-of-plain y-z shear modulus: G23( 1)
 0.2660000E-01 $ minor (small) Poisson ratio: NU( 1)
 0.2500000E-06 $ coef. of thermal expansion along the fibers: ALPHA1( 1)
 0.1620000E-04 $ coef. of thermal expansion transverse to fibers: ALPHA2( 1)
     240        $ curing temperature difference: TEMCUR( 1) <-- see p.16
 0.5700000E-01 $ weight density of material: DENSTY( 1)
      7         $ Number ITHICK  of rows in the array  THICK: ITHICK
    0.0052      $ layer type thickness: THICK( 1)
    0.0052      $ layer type thickness: THICK( 2)
    0.0052      $ layer type thickness: THICK( 3)
    0.0052      $ layer type thickness: THICK( 4)
    0.0052      $ layer type thickness: THICK( 5)
    0.0156      $ layer type thickness: THICK( 6)
    0.0156      $ layer type thickness: THICK( 7)
      45        $ layer type layup angle: ANGLE( 1)
     -45        $ layer type layup angle: ANGLE( 2)
      90        $ layer type layup angle: ANGLE( 3)
      45        $ layer type layup angle: ANGLE( 4)
     -45        $ layer type layup angle: ANGLE( 5)
       0        $ layer type layup angle: ANGLE( 6)
      90        $ layer type layup angle: ANGLE( 7)
       1        $ layer type material type: MATTYP( 1)
       1        $ layer type material type: MATTYP( 2)
       1        $ layer type material type: MATTYP( 3)
       1        $ layer type material type: MATTYP( 4)
       1        $ layer type material type: MATTYP( 5)
       1        $ layer type material type: MATTYP( 6)
       1        $ layer type material type: MATTYP( 7)
      22        $ Number JLAYTYP of columns in the array, LAYTYP: JLAYTYP
      12        $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
       1        $ layer type: LAYTYP( 1, 1)
       2        $ layer type: LAYTYP( 2, 1)
       3        $ layer type: LAYTYP( 3, 1)
       2        $ layer type: LAYTYP( 4, 1)
       1        $ layer type: LAYTYP( 5, 1)
       4        $ layer type: LAYTYP( 6, 1)
       5        $ layer type: LAYTYP( 7, 1)
       6        $ layer type: LAYTYP( 8, 1)
       7        $ layer type: LAYTYP( 9, 1)
       6        $ layer type: LAYTYP(10, 1)
       5        $ layer type: LAYTYP(11, 1)
       4        $ layer type: LAYTYP(12, 1)
       5        $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
       1        $ layer type: LAYTYP( 1, 2)
       2        $ layer type: LAYTYP( 2, 2)
       3        $ layer type: LAYTYP( 3, 2)
       2        $ layer type: LAYTYP( 4, 2)
       1        $ layer type: LAYTYP( 5, 2)
       7        $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
       4        $ layer type: LAYTYP( 1, 3)
       5        $ layer type: LAYTYP( 2, 3)
       6        $ layer type: LAYTYP( 3, 3)
       7        $ layer type: LAYTYP( 4, 3)
       6        $ layer type: LAYTYP( 5, 3)
       5        $ layer type: LAYTYP( 6, 3)
       4        $ layer type: LAYTYP( 7, 3)
       5        $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
       1        $ layer type: LAYTYP( 1, 4)
       2        $ layer type: LAYTYP( 2, 4)
       3        $ layer type: LAYTYP( 3, 4)
```

```
  2      $ layer type: LAYTYP( 4, 4)
  1      $ layer type: LAYTYP( 5, 4)
  5      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
  2      $ layer type: LAYTYP( 1, 5)
  1      $ layer type: LAYTYP( 2, 5)
  3      $ layer type: LAYTYP( 3, 5)
  1      $ layer type: LAYTYP( 4, 5)
  2      $ layer type: LAYTYP( 5, 5)
 12      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
  1      $ layer type: LAYTYP( 1, 6)
  2      $ layer type: LAYTYP( 2, 6)
  3      $ layer type: LAYTYP( 3, 6)
  2      $ layer type: LAYTYP( 4, 6)
  1      $ layer type: LAYTYP( 5, 6)
  4      $ layer type: LAYTYP( 6, 6)
  5      $ layer type: LAYTYP( 7, 6)
  6      $ layer type: LAYTYP( 8, 6)
  7      $ layer type: LAYTYP( 9, 6)
  6      $ layer type: LAYTYP(10, 6)
  5      $ layer type: LAYTYP(11, 6)
  4      $ layer type: LAYTYP(12, 6)
 10      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
  1      $ layer type: LAYTYP( 1, 7)
  2      $ layer type: LAYTYP( 2, 7)
  3      $ layer type: LAYTYP( 3, 7)
  2      $ layer type: LAYTYP( 4, 7)
  1      $ layer type: LAYTYP( 5, 7)
  2      $ layer type: LAYTYP( 6, 7)
  1      $ layer type: LAYTYP( 7, 7)
  3      $ layer type: LAYTYP( 8, 7)
  1      $ layer type: LAYTYP( 9, 7)
  2      $ layer type: LAYTYP(10, 7)
  7      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
  4      $ layer type: LAYTYP( 1, 8)
  5      $ layer type: LAYTYP( 2, 8)
  6      $ layer type: LAYTYP( 3, 8)
  7      $ layer type: LAYTYP( 4, 8)
  6      $ layer type: LAYTYP( 5, 8)
  5      $ layer type: LAYTYP( 6, 8)
  4      $ layer type: LAYTYP( 7, 8)
  5      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
  1      $ layer type: LAYTYP( 1, 9)
  2      $ layer type: LAYTYP( 2, 9)
  3      $ layer type: LAYTYP( 3, 9)
  2      $ layer type: LAYTYP( 4, 9)
  1      $ layer type: LAYTYP( 5, 9)
  5      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
  2      $ layer type: LAYTYP( 1,10)
  1      $ layer type: LAYTYP( 2,10)
  3      $ layer type: LAYTYP( 3,10)
  1      $ layer type: LAYTYP( 4,10)
  2      $ layer type: LAYTYP( 5,10)
  5      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
  2      $ layer type: LAYTYP( 1,11)
  1      $ layer type: LAYTYP( 2,11)
  3      $ layer type: LAYTYP( 3,11)
  1      $ layer type: LAYTYP( 4,11)
  2      $ layer type: LAYTYP( 5,11)
 10      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
  1      $ layer type: LAYTYP( 1,12)
  2      $ layer type: LAYTYP( 2,12)
  3      $ layer type: LAYTYP( 3,12)
  2      $ layer type: LAYTYP( 4,12)
  1      $ layer type: LAYTYP( 5,12)
  2      $ layer type: LAYTYP( 6,12)
  1      $ layer type: LAYTYP( 7,12)
  3      $ layer type: LAYTYP( 8,12)
  1      $ layer type: LAYTYP( 9,12)
  2      $ layer type: LAYTYP(10,12)
  5      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
  2      $ layer type: LAYTYP( 1,13)
  1      $ layer type: LAYTYP( 2,13)
  3      $ layer type: LAYTYP( 3,13)
  1      $ layer type: LAYTYP( 4,13)
  2      $ layer type: LAYTYP( 5,13)
  5      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
  1      $ layer type: LAYTYP( 1,14)
  2      $ layer type: LAYTYP( 2,14)
```

Table 14 test.BEG (p. 3 of 4)

```
   3      $ layer type: LAYTYP( 3,14)
   2      $ layer type: LAYTYP( 4,14)
   1      $ layer type: LAYTYP( 5,14)
  12      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
   1      $ layer type: LAYTYP( 1,15)
   2      $ layer type: LAYTYP( 2,15)
   3      $ layer type: LAYTYP( 3,15)
   2      $ layer type: LAYTYP( 4,15)
   1      $ layer type: LAYTYP( 5,15)
   5      $ layer type: LAYTYP( 6,15)
   4      $ layer type: LAYTYP( 7,15)
   6      $ layer type: LAYTYP( 8,15)
   7      $ layer type: LAYTYP( 9,15)
   6      $ layer type: LAYTYP(10,15)
   4      $ layer type: LAYTYP(11,15)
   5      $ layer type: LAYTYP(12,15)
   5      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
   2      $ layer type: LAYTYP( 1,16)
   1      $ layer type: LAYTYP( 2,16)
   3      $ layer type: LAYTYP( 3,16)
   1      $ layer type: LAYTYP( 4,16)
   2      $ layer type: LAYTYP( 5,16)
   7      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
   4      $ layer type: LAYTYP( 1,17)
   5      $ layer type: LAYTYP( 2,17)
   6      $ layer type: LAYTYP( 3,17)
   7      $ layer type: LAYTYP( 4,17)
   6      $ layer type: LAYTYP( 5,17)
   5      $ layer type: LAYTYP( 6,17)
   4      $ layer type: LAYTYP( 7,17)
  12      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
   4      $ layer type: LAYTYP( 1,18)
   5      $ layer type: LAYTYP( 2,18)
   6      $ layer type: LAYTYP( 3,18)
   7      $ layer type: LAYTYP( 4,18)
   6      $ layer type: LAYTYP( 5,18)
   5      $ layer type: LAYTYP( 6,18)
   4      $ layer type: LAYTYP( 7,18)
   2      $ layer type: LAYTYP( 8,18)
   1      $ layer type: LAYTYP( 9,18)
   3      $ layer type: LAYTYP(10,18)
   1      $ layer type: LAYTYP(11,18)
   2      $ layer type: LAYTYP(12,18)
   5      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
   1      $ layer type: LAYTYP( 1,19)
   2      $ layer type: LAYTYP( 2,19)
   3      $ layer type: LAYTYP( 3,19)
   2      $ layer type: LAYTYP( 4,19)
   1      $ layer type: LAYTYP( 5,19)
   5      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
   1      $ layer type: LAYTYP( 1,20)
   2      $ layer type: LAYTYP( 2,20)
   3      $ layer type: LAYTYP( 3,20)
   2      $ layer type: LAYTYP( 4,20)
   1      $ layer type: LAYTYP( 5,20)
   5      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
   2      $ layer type: LAYTYP( 1,21)
   1      $ layer type: LAYTYP( 2,21)
   3      $ layer type: LAYTYP( 3,21)
   1      $ layer type: LAYTYP( 4,21)
   2      $ layer type: LAYTYP( 5,21)
   7      $ Number ILAYTYP of rows in this column of LAYTYP: ILAYTYP
   4      $ layer type: LAYTYP( 1,22)
   5      $ layer type: LAYTYP( 2,22)
   6      $ layer type: LAYTYP( 3,22)
   7      $ layer type: LAYTYP( 4,22)
   6      $ layer type: LAYTYP( 5,22)
   5      $ layer type: LAYTYP( 6,22)
   4      $ layer type: LAYTYP( 7,22)
   7      $ number of layers in each face sheet: NLAYRF
   5      $ number of layers around trapezoidal tool: NLAYRC
   0      $ 0=pressure on top face sheet; 1=bottom face sheet: INTEXT
  22      $ number of segments in module: NSEGS
  13      $ number of modules for general buckling: NMODULG
   1      $ number of modules for local buckling: NMODULL
   1      $ low end of the M-range for general buckling: MLOWG
   5      $ high end of the M-range for general buckling: MHIGHG
   1      $ low end of the range of local buckling waves: MLOWL
```