**Table 1 Glossary of the variables for the generic case, "balloon". These variables and their definitions, roles, etc. are established by the GENOPT user during the interactive session, GENTEXT. This table is part of the balloon.DEF file generated automatically by GENTEXT.**

| ARRAY ? | NUMBER OF (ROWS,COLS) | ROLE | PROMPT NUMBER (balloon.PRO) | VARIABLE NAME | DEFINITION OF VARIABLE |
|---|---|---|---|---|---|
| n | ( 0, 0) | 2 | 10 | LENGTH | = length of the cylindrical shell |
| n | ( 0, 0) | 2 | 15 | RADIUS | = inner radius of the cylindrical balloon |
| n | ( 0, 0) | 2 | 20 | NMODUL | = number of modules over 90 degrees |
| n | ( 0, 0) | 2 | 30 | IEMOD1 | = material number in EMOD1(IEMOD1) |
| y | ( 10, 0) | 2 | 35 | EMOD1 | = elastic modulus, meridional direction |
| y | ( 10, 0) | 2 | 40 | EMOD2 | = elastic modulus, circumferential direction |
| y | ( 10, 0) | 2 | 45 | G12 | = in-plane shear modulus |
| y | ( 10, 0) | 2 | 50 | G13 | = out-of-plane (s,z) shear modulus |
| y | ( 10, 0) | 2 | 55 | G23 | = out-of-plane (y,z) shear modulus |
| y | ( 10, 0) | 2 | 60 | NU | = Poisson ratio |
| y | ( 10, 0) | 2 | 65 | ALPHA1 | = meridional coef. thermal expansion |
| y | ( 10, 0) | 2 | 70 | ALPHA2 | = circumf.coef.thermal expansion |
| y | ( 10, 0) | 2 | 75 | TEMPER | = delta-T from fabrication temperature |
| y | ( 10, 0) | 2 | 80 | DENSTY | = weight density of material |
| n | ( 0, 0) | 1 | 90 | HEIGHT | = height from inner to outer membranes |
| n | ( 0, 0) | 1 | 95 | RINNER | = radius of curvature of inner membrane |
| n | ( 0, 0) | 1 | 100 | ROUTER | = radius of curvature of outer membrane |
| n | ( 0, 0) | 1 | 105 | TINNER | = thickness of the inner curved membrane |
| n | ( 0, 0) | 1 | 110 | TOUTER | = thickness of the outer curved membrane |
| n | ( 0, 0) | 1 | 115 | TFINNR | = thickness of inner truss-core segment |
| n | ( 0, 0) | 1 | 120 | TFOUTR | = thickness of the outer truss segment |
| n | ( 0, 0) | 1 | 125 | TFWEBS | = thickness of each truss-core web |
| n | ( 0, 0) | 2 | 135 | NCASES | = Number of load cases (number of environments) |
| y | ( 20, 0) | 3 | 140 | PINNER | = pressure inside the inner membrane |
| y | ( 20, 0) | 3 | 145 | PMIDDL | = pressure between inner and outer membranes |
| y | ( 20, 0) | 3 | 150 | POUTER | = pressure outside the outer membrane |
| y | ( 20, 0) | 4 | 160 | GENBUK | = general buckling load factor |
| y | ( 20, 0) | 5 | 165 | GENBUKA | = allowable for general buckling load factor |
| y | ( 20, 0) | 6 | 170 | GENBUKF | = general buckling factor of safety |
| n | ( 0, 0) | 2 | 175 | JSTRM1 | = stress component number in STRM1(NCASES,JSTRM1) |
| y | ( 20, 5) | 4 | 180 | STRM1 | = stress component in material 1 |
| y | ( 20, 5) | 5 | 185 | STRM1A | = allowable stress in material 1 |
| y | ( 20, 5) | 6 | 190 | STRM1F | = factor of safety for stress in material 1 |
| y | ( 20, 5) | 4 | 195 | STRM2 | = stress component in material 2 |
| y | ( 20, 5) | 5 | 200 | STRM2A | = allowable for stress in material 2 |
| y | ( 20, 5) | 6 | 205 | STRM2F | = factor of safety for stress in material 2 |
| y | ( 20, 5) | 4 | 210 | STRM3 | = stress component in material 3 |
| y | ( 20, 5) | 5 | 215 | STRM3A | = allowable for stress in material 3 |
| y | ( 20, 5) | 6 | 220 | STRM3F | = factor of safety for stress in material 3 |
| n | ( 0, 0) | 7 | 230 | WEIGHT | = weight/length of the balloon |

**Table 2 The prompting file, balloon.PRO, generated automatically by the GENTEXT processor. This file contains the one-line prompts and "help" paragraphs that are created by the GENOPT user and seen by the end user.**

```
=======================================================================
   5.0
       This GENOPT case is for a cylindrical balloon
       the wall of which is a double-walled sandwich made of cloth.
       The case was brought to the attention of the author,
       David Bushnell, by Mike Mayo (650-354-5463) on September 21,
       1010. In this application, GENOPT works with BIGBOSOR4.
       The cylindrical "balloon" has inner pressure
       equal to PINNER, outer pressure equal to POUTER, and
       pressure inside the double-walled sandwich equal to PMIDDL.
       PINNER is the lowest pressure, PMIDDL is the highest
       pressure, and POUTER is higher than PINNER but lower than
       PMIDDL. PMIDDL must be high enough to provide enough
       tension in the membrane segments of the "balloon" to
       prevent buckling under the difference, POUTER - PINNER.
       Details of the model and results are presented in Ref.[1]:
       [1] Bushnell, David, "Use of GENOPT and BIGBOSOR4 to obtain
       optimum designs of a double-walled inflatable cylindrical
       vacuum chamber", unpublished report dated November, 2010.
       Although the BIGBOSOR4 computer program is intended for
       use with axisymmetric shell structures with "finite"
       bending stiffness, the results obtained from this study of
       a balloon that consists of "shell" segments that act like
       membranes with essentially zero bending stiffness seem to
       be reasonable. It is emphasized that the results presented
       in [1] should be verified via models of the optimized
       designs from one or more general-purpose finite element
       codes such as STAGS or ABAQUS or NASTRAN.

  10.1 length of the cylindrical shell: LENGTH
  10.2
       Use a value of about 6000 inches. It should not
       matter what value you use because buckling (or collapse)
       with N = zero circumferential waves around the circumference
       of the huge torus is expected to be critical as of this
       writing. N = 1 is used instead of N = 0 in order to
       avoid rigid body "buckling" possible with N = 0.

  15.1 inner radius of the cylindrical balloon: RADIUS
  15.2
```

This is the radius to the points on the inner membranes
where these "shell" segments are connected to each other.
See Fig. x of [1].

20.1 number of modules over 90 degrees: NMODUL
20.2
This is the number of triangular "trusses" with two points
on the inner membrane and one point on the outer membrane
over a 90-degree sector of the circumference of the
cylindrical balloon. See Figs. 2 and 5 of [1]. For the
configuration in which the webs are radial rather than
slanted, the number of modules is equal to the number
of radial webs over 90 degrees of the circumference
of the cylindrical balloon. See Figs. 1 and 4 of [1].

25.0
Next you will be asked to provide material properties.
Three different materials are allowed:
1. The material of the outer and inner curved membranes.
2. The material of the outer and inner "truss" members
that run in the circumferential direction.
3. The material out of which the "truss" (slanted, Fig.2)
or radial (Fig. 1) webs are fabricated.
The material is orthotropic with the following properties:
EMOD1 = modulus in the meridional direction, that is,
in the direction along the arc of each shell segment in
the plane of the cross section of the complex wall
of the balloon.
EMOD2 = modulus in the circumferential direction of the
huge torus, that is, the modulus along the axis of the
prismatic shell.
G12 = in-plane shear modulus, that is, in the plane of
the wall of a "shell" segment
G13 = out-of plane shear modulus (not used, input required)
G23 = out-of-plane shear mdoulus (not used, input required)
NU = Poisson ratio
ALPHA1 = coefficient of thermal expansion in the meridional
direction
ALPHA2 = coefficient of thermal expansion in the
circumferential direction (prismatic axial direction)
TEMPER = temperature difference from the temperature at
which the balloon was fabricated (not used, input required)
DENSTY = weight density of the material
(Aluminum = 0.1 lb/in^3)

30.1 Number IEMOD1  of rows in the array  EMOD1: IEMOD1
35.1 elastic modulus, meridional direction: EMOD1

40.1 elastic modulus, circumferential direction: EMOD2
45.1 in-plane shear modulus: G12
50.1 out-of-plane (s,z) shear modulus: G13
55.1 out-of-plane (y,z) shear modulus: G23
60.1 Poisson ratio: NU
65.1 meridional coef. thermal expansion: ALPHA1
70.1 circumf.coef.thermal expansion: ALPHA2
75.1 delta-T from fabrication temperature: TEMPER
80.1 weight density of material: DENSTY

85.0
Next, you will be asked to supply the decision variable candidates. These are as follows:
1. HEIGHT = radial difference between the inner radius, RADIUS, and the outer radius where the various segments of the "balloon" are joined together.
2. RINNER = radius of curvature of the inner curved membrane, the one that "bulges" inward.
3. ROUTER = radius of curvature of the outer curved membrane, the one that "bulges" outward
4. TINNER = thickness of the inner curved membrane
5. TOUTER = thickness of the outer curved membrane
6. TFINNR = thickness of outer triangular truss segment
7. TFOUTR = thickness of inner triagular truss segment
8, TFWEBS = thickness of the webs

90.1 height from inner to outer membranes: HEIGHT
90.2
This is the difference from inner to outer radii at the points where the inner segments are joined to eachother and the outer segments are joined to eachother, that is, the height between inner and outer walls of the "balloon" not including the inward "bulging" of the inner wall and the outward "bulging" of the outer wall.

95.1 radius of curvature of inner membrane: RINNER
100.1 radius of curvature of outer membrane: ROUTER
105.1 thickness of the inner curved membrane: TINNER
110.1 thickness of the outer curved membrane: TOUTER
115.1 thickness of inner truss-core segment: TFINNR
115.2
The three straight segments that form each module of the truss core have different thicknesses as follows:
1. The outer truss-core member that is oriented in the circumferential direction has thickness, TFOUTR.
2. The inner truss-core member that is oriented in the circumferential direction has thickness, TFINNR.
3. The two truss-core webs each have thickness, TFWEBS

120.1 thickness of the outer truss segment: TFOUTR
120.2
The three straight segments that form each module of the truss core have different thicknesses as follows:
1. The outer truss-core member that is oriented in the circumferential direction has thickness, TFOUTR.
2. The inner truss-core member that is oriented in the circumferential direction has thickness, TFINNR.
3. The two truss-core webs each have thickness, TFWEBS

125.1 thickness of each truss-core web: TFWEBS
125.2
The three straight segments that form each module of the truss core have different thicknesses as follows:
1. The outer truss-core member that is oriented in the circumferential direction has thickness, TFOUTR.
2. The inner truss-core member that is oriented in the circumferential direction has thickness, TFINNR.
3. The two truss-core webs each have thickness, TFWEBS

130.0
Next, you will be asked to provide three pressures, PINNER, PMIDDL, and POUTER, which are different from each other and which are uniform over the entire structure.
1. PINNER = pressure inside the inner membrane. This is the lowest of the three pressures.
2. PMIDDL = pressure between the inner membrane and outer membrane. This is the highest of the three pressures.
3. POUTER = pressure outside the outer membrane. This pressure is higher than PINNER and lower than PMIDDL.
Use positive numbers for PINNER, PMIDDL, and POUTER.

135.1 Number NCASES of load cases (environments): NCASES
140.1 pressure inside the inner membrane: PINNER
145.1 pressure between inner and outer membranes: PMIDDL
150.1 pressure outside the outer membrane: POUTER

155.0
Next, you will be asked to provide the "behaviors" that might affect the evolution of the design during optimization cycles. The "behaviors" included here are:
1. general buckling: GENBUK, GENBUKA, GENBUKF
GENBUK = general buckling load factor
GENBUKA= general buckling allowable

GENBUKF= general buckling factor of safety
NOTE: The "GENBUK" mode shape may actually represent
local buckling, not general buckling. In this generic
"balloon" case only the lowest buckling eigenvalue
is computed, whether it correspond to a general
buckling mode shape or whether it correspond to a
local buckling mode shape. Whichever buckling mode
happens to be represented by "GENBUK" will correspond
to the lowest eigenvalue. The other type of buckling
(general buckling if the lowest eigenvalue corresponds
to local buckling and local buckling if the lowest
eigenvalue corresponds to general buckling) will be
higher than the eigenvalue used to generate the
buckling constraint condition.
2. stresses: STRMi(j,k), STRMiA(j,k), STRMiF(j,k)
in which "i" is the material number, "j" is the load case
number, and "k" is the stress component.
STRMi(j,k) is the maximum stress.
STRMiA(j,k) is the stress allowable
STRMiF(j,k) is the stress factor of safety.
There are five stress components:
STRMi(j,1) = maximum tensile stress in the meridional direction
STRMi(j,2) = maximum compressive stress in the meridional direction
STRMi(j,3) = maxiamum tensile stress in the circumfer. direction
STRMi(j,4) = maximum compressive stress in the circumf.direction
STRMi(j,5) = maximum in-plane shear stress.

160.0 general buckling load factor: GENBUK
165.1 allowable for general buckling load factor: GENBUKA
165.2
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 in order to obtain
the buckling load.

170.1 general buckling factor of safety: GENBUKF
170.2
For this problem, use 1.0.

175.1 Number JSTRM1 of columns in the array, STRM1: JSTRM1
180.0 stress component in material 1: STRM1
180.2
For an orthotropic material there are 5 stress components
for which stress constraints may be generated:
1. maximum tensile stess in the meridional direction
2. maximum compressive stress in the meridional direction
3. maximum tensile stress in the circumferential direction
4. maximum compressive stress in the circumferential direction

```
      5. maximum in-plane shear stress

 185.1 allowable stress in material 1: STRM1A
 185.2
      For an orthotropic material there are 5 stress components
      for which stress constraints are generated:
      1. maximum tensile stress in the meridional direction:
      STRM1A(i,1), in which "i" is the load set number
      2. maximum compressive stress in the meridional direction:
      STRM1A(i,2), in which "i" is the load set number
      3. maximum tensile stress in the circumferential direction:
      STRM1A(i,3), in which "i" is the load set number
      4. maximum compressive stress in the circumferential direction:
      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

 190.1 factor of safety for stress in material 1: STRM1F
 190.2
      In this application use a factor of safety of 1.0

 195.0 stress component in material 2: STRM2
 200.1 allowable for stress in material 2: STRM2A
 205.1 factor of safety for stress in material 2: STRM2F
 210.0 stress component in material 3: STRM3
 215.1 allowable for stress in material 3: STRM3A
 220.1 factor of safety for stress in material 3: STRM3F

 225.0
      Next, you will be asked to provide an objective.
      In this case the objective is the weight/(axial length)
      of the balloon.

 230.0 weight/length of the balloon: WEIGHT
 999.0 DUMMY ENTRY TO MARK END OF FILE
===================================================================
```

**Table 3 Input file, balloon.INP, for the GENTEXT processor. This file is generated during the interactive GENTEXT session conducted by the GENOPT user. This is where the GENOPT user provides the variable names, one-line definitions, and "help" paragraphs that appear in the balloon.PRO file (Table 2) and that are seen by the end user.**

```
==========================================================================
      5  $ starting prompt index in the file balloon.PRO
      5  $ increment for prompt index
      0  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
 This GENOPT case is for a cylindrical balloon
 y         $ Are there more lines in the "help" paragraph?
 the wall of which is a double-walled sandwich made of cloth.
 y         $ Are there more lines in the "help" paragraph?
 The case was brought to the attention of the author,
 y         $ Are there more lines in the "help" paragraph?
 David Bushnell, by Mike Mayo (650-354-5463) on September 21,
 y         $ Are there more lines in the "help" paragraph?
 1010. In this application, GENOPT works with BIGBOSOR4.
 y         $ Are there more lines in the "help" paragraph?
 The cylindrical "balloon" has inner pressure
 y         $ Are there more lines in the "help" paragraph?
 equal to PINNER, outer pressure equal to POUTER, and
 y         $ Are there more lines in the "help" paragraph?
 pressure inside the double-walled sandwich equal to PMIDDL.
 y         $ Are there more lines in the "help" paragraph?
 PINNER is the lowest pressure, PMIDDL is the highest
 y         $ Are there more lines in the "help" paragraph?
 pressure, and POUTER is higher than PINNER but lower than
 y         $ Are there more lines in the "help" paragraph?
 PMIDDL. PMIDDL must be high enough to provide enough
 y         $ Are there more lines in the "help" paragraph?
 tension in the membrane segments of the "balloon" to
 y         $ Are there more lines in the "help" paragraph?
 prevent buckling under the difference, POUTER - PINNER.
 y         $ Are there more lines in the "help" paragraph?
 Details of the model and results are presented in Ref.[1]:
 y         $ Are there more lines in the "help" paragraph?
 [1] Bushnell, David, "Use of GENOPT and BIGBOSOR4 to obtain
 y         $ Are there more lines in the "help" paragraph?
 optimum designs of a double-walled inflatable cylindrical
 y         $ Are there more lines in the "help" paragraph?
 vacuum chamber", unpublished report dated November, 2010.
```

```
y         $ Are there more lines in the "help" paragraph?
Although the BIGBOSOR4 computer program is intended for
y         $ Are there more lines in the "help" paragraph?
use with axisymmetric shell structures with "finite"
y         $ Are there more lines in the "help" paragraph?
bending stiffness, the results obtained from this study of
y         $ Are there more lines in the "help" paragraph?
a balloon that consists of "shell" segments that act like
y         $ Are there more lines in the "help" paragraph?
membranes with essentially zero bending stiffness seem to
y         $ Are there more lines in the "help" paragraph?
be reasonable. It is emphasized that the results presented
y         $ Are there more lines in the "help" paragraph?
in [1] should be verified via models of the optimized
y         $ Are there more lines in the "help" paragraph?
designs from one or more general-purpose finite element
y         $ Are there more lines in the "help" paragraph?
codes such as STAGS or ABAQUS or NASTRAN.
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
n         $ Is the variable  LENGTH  an array?
length of the cylindrical shell
y         $ Do you want to include a "help" paragraph?
Use a value of about 6000 inches. It should not
y         $ Any more lines in the "help" paragraph?
matter what value you use because buckling (or collapse)
y         $ Any more lines in the "help" paragraph?
with N = zero circumferential waves around the circumference
y         $ Any more lines in the "help" paragraph?
of the huge torus is expected to be critical as of this
y         $ Any more lines in the "help" paragraph?
writing. N = 1 is used instead of N = 0 in order to
y         $ Are there more lines in the "help" paragraph?
avoid rigid body "buckling" possible with N = 0.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $10
      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?
inner radius of the cylindrical balloon
y         $ Do you want to include a "help" paragraph?
This is the radius to the points on the inner membranes
```

```
y         $ Any more lines in the "help" paragraph?
where these "shell" segments are connected to each other.
y         $ Any more lines in the "help" paragraph?
See Fig. x of [1].
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $15
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
NMODUL    $ 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  NMODUL  an array?
number of modules over 90 degrees
y         $ Do you want to include a "help" paragraph?
This is the number of triangular "trusses" with two points
y         $ Any more lines in the "help" paragraph?
on the inner membrane and one point on the outer membrane
y         $ Any more lines in the "help" paragraph?
over a 90-degree sector of the circumference of the
y         $ Any more lines in the "help" paragraph?
cylindrical balloon. See Figs. 2 and 5 of [1]. For the
y         $ Any more lines in the "help" paragraph?
configuration in which the webs are radial rather than
y         $ Any more lines in the "help" paragraph?
slanted, the number of modules is equal to the number
y         $ Any more lines in the "help" paragraph?
of radial webs over 90 degrees of the circumference
y         $ Any more lines in the "help" paragraph?
of the cylindrical balloon. See Figs. 1 and 4 of [1].
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $20
      0  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
Next you will be asked to provide material properties.
y         $ Are there more lines in the "help" paragraph?
Three different materials are allowed:
y         $ Are there more lines in the "help" paragraph?
1. The material of the outer and inner curved membranes.
y         $ Are there more lines in the "help" paragraph?
2. The material of the outer and inner "truss" members
y         $ Any more lines in the "help" paragraph?
that run in the circumferential direction.
y         $ Are there more lines in the "help" paragraph?
3. The material out of which the "truss" (slanted, Fig.2)
y         $ Any more lines in the "help" paragraph?
or radial (Fig. 1) webs are fabricated.
y         $ Are there more lines in the "help" paragraph?
The material is orthotropic with the following properties:
y         $ Are there more lines in the "help" paragraph?
EMOD1 = modulus in the meridional direction, that is,
```

```
y         $ Any more lines in the "help" paragraph?
in the direction along the arc of each shell segment in
y         $ Any more lines in the "help" paragraph?
the plane of the cross section of the complex wall
y         $ Any more lines in the "help" paragraph?
of the balloon.
y         $ Are there more lines in the "help" paragraph?
EMOD2 = modulus in the circumferential direction of the
y         $ Are there more lines in the "help" paragraph?
huge torus, that is, the modulus along the axis of the
y         $ Are there more lines in the "help" paragraph?
prismatic shell.
y         $ Are there more lines in the "help" paragraph?
G12 = in-plane shear modulus, that is, in the plane of
y         $ Any more lines in the "help" paragraph?
the wall of a "shell" segment
y         $ Are there more lines in the "help" paragraph?
G13 = out-of plane shear modulus (not used, input required)
y         $ Are there more lines in the "help" paragraph?
G23 = out-of-plane shear mdoulus (not used, input required)
y         $ Are there more lines in the "help" paragraph?
NU = Poisson ratio
y         $ Are there more lines in the "help" paragraph?
ALPHA1 = coefficient of thermal expansion in the meridional
y         $ Are there more lines in the "help" paragraph?
direction
y         $ Are there more lines in the "help" paragraph?
ALPHA2 = coefficient of thermal expansion in the
y         $ Are there more lines in the "help" paragraph?
circumferential direction (prismatic axial direction)
y         $ Are there more lines in the "help" paragraph?
TEMPER = temperature difference from the temperature at
y         $ Are there more lines in the "help" paragraph?
which the balloon was fabricated (not used, input required)
y         $ Are there more lines in the "help" paragraph?
DENSTY = weight density of the material
y         $ Are there more lines in the "help" paragraph?
(Aluminum = 0.1 lb/in^3)
n         $ Are there more lines in the "help" paragraph?
      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 number
     10  $ Max. allowable number of rows NROWS in the array, EMOD1
```

```
elastic modulus, meridional direction
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $35
      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, circumferential direction
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $40
      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   ?    $45
      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 (s,z) shear modulus
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $50
      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-plane (y,z) shear modulus
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $55
      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 ?
Poisson ratio
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $60
```

```
      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 ?
meridional coef. thermal expansion
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $65
      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 ?
circumf.coef.thermal expansion
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $70
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
TEMPER    $ 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  TEMPER  an array?
n         $ Do you want to establish new dimensions for TEMPER ?
delta-T from fabrication temperature
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $75
      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
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $80
      0  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
Next, you will be asked to supply the decision variable
y         $ Are there more lines in the "help" paragraph?
candidates. These are as follows:
y         $ Are there more lines in the "help" paragraph?
1. HEIGHT = radial difference between the inner radius, RADIUS,
y         $ Are there more lines in the "help" paragraph?
and the outer radius where the various segments
y         $ Are there more lines in the "help" paragraph?
of the "balloon" are joined together.
y         $ Are there more lines in the "help" paragraph?
2. RINNER = radius of curvature of the inner curved membrane,
```

```
y         $ Are there more lines in the "help" paragraph?
the one that "bulges" inward.
y         $ Are there more lines in the "help" paragraph?
3. ROUTER = radius of curvature of the outer curved membrane,
y         $ Are there more lines in the "help" paragraph?
the one that "bulges" outward
y         $ Are there more lines in the "help" paragraph?
4. TINNER = thickness of the inner curved membrane
y         $ Are there more lines in the "help" paragraph?
5. TOUTER = thickness of the outer curved membrane
y         $ Are there more lines in the "help" paragraph?
6. TFINNR = thickness of outer triangular truss segment
y         $ Are there more lines in the "help" paragraph?
7. TFOUTR = thickness of inner triagular truss segment
y         $ Are there more lines in the "help" paragraph?
8, TFWEBS = thickness of the webs
n         $ Are there more lines in the "help" paragraph?
      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 from inner to outer membranes
y         $ Do you want to include a "help" paragraph?
This is the difference from inner to outer radii at the
y         $ Any more lines in the "help" paragraph?
points where the inner segments are joined to eachother
y         $ Any more lines in the "help" paragraph?
and the outer segments are joined to eachother, that is,
y         $ Any more lines in the "help" paragraph?
the height between inner and outer walls of the "balloon"
y         $ Any more lines in the "help" paragraph?
not including the inward "bulging" of the inner wall and
y         $ Any more lines in the "help" paragraph?
the outward "bulging" of the outer wall.
n         $ Any more lines in the "help" paragraph?
y         $ Any more variables for role types  1  or  2   ?    $90
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
RINNER    $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
n         $ Is the variable  RINNER  an array?
radius of curvature of inner membrane
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
ROUTER    $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
n         $ Is the variable  ROUTER  an array?
radius of curvature of outer membrane
```

```
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
TINNER    $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
n          $ Is the variable  TINNER  an array?
thickness of the inner curved membrane
n          $ Do you want to include a "help" paragraph?
y          $ Any more variables for role types  1  or  2   ?   $105
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
TOUTER    $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
n          $ Is the variable  TOUTER  an array?
thickness of the outer curved membrane
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
TFINNR    $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
n          $ Is the variable  TFINNR  an array?
thickness of inner truss-core segment
y          $ Do you want to include a "help" paragraph?
The three straight segments that form each module of the
y          $ Any more lines in the "help" paragraph?
truss core have different thicknesses as follows:
y          $ Any more lines in the "help" paragraph?
1. The outer truss-core member that is oriented in the
y          $ Any more lines in the "help" paragraph?
circumferential direction has thickness, TFOUTR.
y          $ Any more lines in the "help" paragraph?
2. The inner truss-core member that is oriented in the
y          $ Any more lines in the "help" paragraph?
circumferential direction has thickness, TFINNR.
y          $ Any more lines in the "help" paragraph?
3. The two truss-core webs each have thickness, TFWEBS
n          $ Any more lines in the "help" paragraph?
y          $ Any more variables for role types  1  or  2   ?   $115
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
TFOUTR    $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
n          $ Is the variable  TFWEBS  an array?
thickness of the outer truss segment
y          $ Do you want to include a "help" paragraph?
The three straight segments that form each module of the
y          $ Any more lines in the "help" paragraph?
truss core have different thicknesses as follows:
y          $ Any more lines in the "help" paragraph?
1. The outer truss-core member that is oriented in the
```

```
y         $ Any more lines in the "help" paragraph?
circumferential direction has thickness, TFOUTR.
y         $ Any more lines in the "help" paragraph?
2. The inner truss-core member that is oriented in the
y         $ Any more lines in the "help" paragraph?
circumferential direction has thickness, TFINNR.
y         $ Any more lines in the "help" paragraph?
3. The two truss-core webs each have thickness, TFWEBS
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
TFWEBS    $ Name of a variable in the users program (defined below)
      1  $ Role of the variable in the users program
n         $ Is the variable  TFOUTR  an array?
thickness of each truss-core web
y         $ Do you want to include a "help" paragraph?
The three straight segments that form each module of the
y         $ Any more lines in the "help" paragraph?
truss core have different thicknesses as follows:
y         $ Any more lines in the "help" paragraph?
1. The outer truss-core member that is oriented in the
y         $ Any more lines in the "help" paragraph?
circumferential direction has thickness, TFOUTR.
y         $ Any more lines in the "help" paragraph?
2. The inner truss-core member that is oriented in the
y         $ Any more lines in the "help" paragraph?
circumferential direction has thickness, TFINNR.
y         $ Any more lines in the "help" paragraph?
3. The two truss-core webs each have thickness, TFWEBS
n         $ Any more lines in the "help" paragraph?
n         $ Any more variables for role types  1  or  2   ?    $
      0  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
Next, you will be asked to provide three pressures,
y         $ Are there more lines in the "help" paragraph?
PINNER, PMIDDL, and POUTER, which are different from
y         $ Are there more lines in the "help" paragraph?
each other and which are uniform over the entire structure.
y         $ Are there more lines in the "help" paragraph?
1. PINNER = pressure inside the inner membrane. This is
y         $ Are there more lines in the "help" paragraph?
the lowest of the three pressures.
y         $ Are there more lines in the "help" paragraph?
2. PMIDDL = pressure between the inner membrane and outer
y         $ Are there more lines in the "help" paragraph?
membrane. This is the highest of the three
y         $ Are there more lines in the "help" paragraph?
pressures.
y         $ Are there more lines in the "help" paragraph?
```

```
3. POUTER = pressure outside the outer membrane. This
y         $ Are there more lines in the "help" paragraph?
pressure is higher than PINNER and lower than
y         $ Are there more lines in the "help" paragraph?
PMIDDL.
y         $ Any more lines in the "help" paragraph?
Use positive numbers for PINNER, PMIDDL, and POUTER.
n         $ Are there more lines in the "help" paragraph?
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
PINNER    $ Name of a variable in the users program (defined below)
      3  $ Role of the variable in the users program
pressure inside the inner membrane
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role type  3 ?            $140
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
PMIDDL    $ Name of a variable in the users program (defined below)
      3  $ Role of the variable in the users program
pressure between inner and outer membranes
n         $ Do you want to include a "help" paragraph?
y         $ Any more variables for role type  3 ?            $145
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
POUTER    $ Name of a variable in the users program (defined below)
      3  $ Role of the variable in the users program
pressure outside the outer membrane
n         $ Do you want to include a "help" paragraph?
n         $ Any more variables for role type  3 ?            $
      0  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
Next, you will be asked to provide the "behaviors" that
y         $ Are there more lines in the "help" paragraph?
might affect the evolution of the design during optimization
y         $ Are there more lines in the "help" paragraph?
cycles. The "behaviors" included here are:
y         $ Are there more lines in the "help" paragraph?
1. general buckling: GENBUK, GENBUKA, GENBUKF
y         $ Are there more lines in the "help" paragraph?
GENBUK = general buckling load factor
y         $ Are there more lines in the "help" paragraph?
GENBUKA= general buckling allowable
y         $ Are there more lines in the "help" paragraph?
GENBUKF= general buckling factor of safety
y         $ Any more lines in the "help" paragraph?
NOTE: The "GENBUK" mode shape may actually represent
y         $ Any more lines in the "help" paragraph?
local buckling, not general buckling. In this generic
y         $ Any more lines in the "help" paragraph?
"balloon" case only the lowest buckling eigenvalue
y         $ Any more lines in the "help" paragraph?
is computed, whether it correspond to a general
```

```
y         $ Any more lines in the "help" paragraph?
buckling mode shape or whether it correspond to a
y         $ Any more lines in the "help" paragraph?
local buckling mode shape. Whichever buckling mode
y         $ Any more lines in the "help" paragraph?
happens to be represented by "GENBUK" will correspond
y         $ Any more lines in the "help" paragraph?
to the lowest eigenvalue. The other type of buckling
y         $ Any more lines in the "help" paragraph?
(general buckling if the lowest eigenvalue corresponds
y         $ Any more lines in the "help" paragraph?
to local buckling and local buckling if the lowest
y         $ Any more lines in the "help" paragraph?
eigenvalue corresponds to general buckling) will be
y         $ Any more lines in the "help" paragraph?
higher than the eigenvalue used to generate the
y         $ Any more lines in the "help" paragraph?
buckling constraint condition.
y         $ Are there more lines in the "help" paragraph?
2. stresses: STRMi(j,k), STRMiA(j,k), STRMiF(j,k)
y         $ Are there more lines in the "help" paragraph?
in which "i" is the material number, "j" is the load case
y         $ Are there more lines in the "help" paragraph?
number, and "k" is the stress component.
y         $ Are there more lines in the "help" paragraph?
STRMi(j,k) is the maximum stress.
y         $ Are there more lines in the "help" paragraph?
STRMiA(j,k) is the stress allowable
y         $ Are there more lines in the "help" paragraph?
STRMiF(j,k) is the stress factor of safety.
y         $ Are there more lines in the "help" paragraph?
There are five stress components:
y         $ Are there more lines in the "help" paragraph?
STRMi(j,1) = maximum tensile stress in the meridional direction
y         $ Are there more lines in the "help" paragraph?
STRMi(j,2) = maximum compressive stress in the meridional direction
y         $ Are there more lines in the "help" paragraph?
STRMi(j,3) = maxiamum tensile stress in the circumfer. direction
y         $ Are there more lines in the "help" paragraph?
STRMi(j,4) = maximum compressive stress in the circumf.direction
y         $ Are there more lines in the "help" paragraph?
STRMi(j,5) = maximum in-plane shear stress.
n         $ Are there more lines in the "help" paragraph?
      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 in order to obtain
y         $ Any more lines in the "help" paragraph?
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?
For this problem, use 1.0.
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 ?             $170
      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
stress component number
      5  $ 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 an orthotropic material there are 5 stress components
y         $ Any more lines in the "help" paragraph?
for which stress constraints may be generated:
y         $ Any more lines in the "help" paragraph?
1. maximum tensile stess in the meridional direction
y         $ Any more lines in the "help" paragraph?
2. maximum compressive stress in the meridional direction
y         $ Any more lines in the "help" paragraph?
3. maximum tensile stress in the circumferential direction
y         $ Any more lines in the "help" paragraph?
4. maximum compressive stress in the circumferential direction
y         $ Any more lines in the "help" paragraph?
5. maximum in-plane shear stress
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 an orthotropic material there are 5 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 stress in the meridional 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 meridional 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 in the circumferential direction:
y         $ Any more lines in the "help" paragraph?
STRM1A(i,3), in which "i" is the load set number
y         $ Any more lines in the "help" paragraph?
4. maximum compressive stress in the circumferential direction:
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
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 application use a factor of safety of 1.0
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 ?           $190
      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
y         $ Any more variables for role type  4 ?             $205
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
STRM3     $ 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 STRM3 ?
stress component in material 3
n         $ Do you want to include a "help" paragraph?
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
STRM3A    $ Name of a variable in the users program (defined below)
      5  $ Role of the variable in the users program
allowable for stress in material 3
n         $ Do you want to include a "help" paragraph?
      1  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
STRM3F    $ 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 3
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 ?             $
      0  $ Type of prompt: 0="help" paragraph, 1=one-line prompt
Next, you will be asked to provide an objective.
y         $ Are there more lines in the "help" paragraph?
In this case the objective is the weight/(axial length)
y         $ Are there more lines in the "help" paragraph?
of the balloon.
n         $ Are there more lines in the "help" paragraph?
      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/length of the balloon
n         $ Do you want to include a "help" paragraph?
======================================================================
```

# Table 4 A detailed list of a typical GENOPT/BIGBOSOR4 run stream

```
==========================================================================

balloon.runstream

October 23 – November 3, 2010

Please read the file, /home/progs/genopt/doc/getting.started.
Please read the paper, /home/progs/genopt/case/balloon/balloon.paper.pdf:

[1] Bushnell, David, "Use of GENOPT and BIGBOSOR4 to obtain optimum
designs of double-walled inflatable cylindrical vacuum chambers",
unpublished report, November, 2010.

Figure numbers and table numbers referenced below are from this report,
called "Ref.[1]" or simply "[1]" in the following text.

NOTE: It is assumed here that the home directory is "/home/progs".
      Commands typed by the GENOPT user and the end user are in
      bold face.


************************* PART 1 ********************************
TASKS PERFORMED BY THE GENOPT USER

First, the GENOPT user must establish the program system by means
of which optimum designs in the generic class called "balloon"
can be found.

------------------ begin GENOPT user's activities ------------------

COMMAND                  MEANING OF COMMAND

cd /home/progs/genoptcase     (go to "genoptcase")

genoptlog (activate the GENOPT set of commands)

gentext         (provide input for GENTEXT: balloon.INP .
                 If you already have a complete file, balloon.INP,
                 then:
                  cp /home/progs/genopt/case/balloon/balloon.INP
                     /home/progs/genoptcase/balloon.INP
                 balloon.INP = Table 3 in the appendix of [1])
```

```
insert    (add other variables you may have forgotten
           during the initial execution of GENTEXT.
           Remember first to save any valuable new
           coding that may exist in behavior.new and/or
           struct.new, as these two files are destroyed
           by re-execution of GENTEXT.)
```

See Table 1 in [1] for a glossary of variables created by the GENOPT user. Table 1 appears as part of the file, balloon.DEF, in /home/progs/genoptcase. This file is created automatically by GENTEXT.

See Table 2 in [1] for a complete list of the file, balloon.PRO, in /home/progs/genoptcase. This file is created automatically by GENTEXT. It is the file with the input data prompting phrases and "help" paragraphs that will be seen by the end user.

Next:

```
1. "flesh out" the skeletal SUBROUTINES BEHX1, BEHX2, BEHX3
   BEHX4, and  OBJECT. See the file:
   /home/progs/genopt/case/balloon/behavior.balloon
   and, if such a file exists in your archive, type:
   cp /home/progs/genopt/case/balloon/behavior.balloon
      /home/progs/genoptcase/behavior.balloon
   behavior.balloon = Table 5 in the appendix of [1].

2. add the three statements, CALL OPNGEN, CALL RWDGEN, and
   call CLSGEN at the appropriate places in the skeletal
   SUBROUTINE STRUCT, See the file:
   /home/progs/genopt/case/balloon/struct.balloon
   and, if such a file exists in your archive, type:
   cp /home/progs/genopt/case/balloon/struct.balloon
      /home/progs/genoptcase/struct.balloon
   struct.balloon = Table 6 in the appendix of [1].

3. Create SUBROUTINE BOSDEC. In this case there are two versions:

   a. bosdec.balloon  = balloon with "truss" (slanted) webs
      (Fig.2 of [1]). See the file:
      /home/progs/genopt/case/balloon/bosdec.balloon
      and, if such a file exists in your archive, type:
      cp /home/progs/genopt/case/balloon/bosdec.balloon
         /home/progs/bosdec/sources/bosdec.balloon).
      bosdec.balloon = Table 7 in the appendix of [1].

   b. bosdec.balloon2 = balloon with radially oriented webs
```

```
    (Fig. 1 of [1]). See the file:
    /home/progs/genopt/case/balloon/bosdec.balloon2
    and, if such a file exists in your archive, type:
    cp /home/progs/genopt/case/balloon/bosdec.balloon2
       /home/progs/bosdec/sources/bosdec.balloon2
    bosdec.balloon2 = no table included in [1] for this file.
                      It is very like bosdec.balloon (Table 7)
```

COMMAND                    MEANING OF COMMAND

**cd /home/progs/bosdec/sources** (go to directory where "bosdec" exists)
**cp bosdec.balloon bosdec.src**
or
**cp bosdec.balloon2 bosdec.src**
**cd /home/progs/genoptcase**     (return to "genoptcase" directory)

and, if appropriate:

**cp behavior.balloon behavior.new**   (store "fleshed out" "behavior")
**cp struct.balloon struct.new**       (store "fleshed out" "struct")

**genprograms**    (compiles the GENOPT program system for the generic case called "balloon")

Next:

1. Correct any FORTRAN errors in SUBROUTINE BOSDEC (bosdec.src)
2. Correct any FORTRAN errors in SUBROUTINES BEHX1, BEHX2, BEHX3, BEHX4,  OBJECT (behavior.new)
3. Store the updated FORTRAN code as follows:
   cp /home/progs/bosdec/sources/bosdec.src
      /home/progs/bosdec/sources/bosdec.balloon or bosdec.balloon2
   whichever version of SUBROUTINE BOSDEC you are working on.
   cp /home/progs/genoptcase/behavior.new
      /home/progs/genoptcase/behavior.balloon
   Make sure before you do these "cp" that the versions of bosdec.src and behavior.new are really the ones you want to use to update bosdec.balloon (or bosdec.balloon2) and behavior.balloon. You don't want erroneously to overwrite your archive files that you have worked so hard on!

COMMAND                    MEANING OF COMMAND

**genprograms**     (re-compiles the GENOPT program system for the generic case called "balloon")

Next, correct any logical errors or other mistakes in SUBROUTINE BOSDEC or in SUBROUTINES BEHX1, BEHX2, BEHX3, BEHX4, OBJECT and again save the updated FORTRAN code as above.

**genprograms** (re-compiles the GENOPT program system for the generic case called "balloon")

Keep correcting bosdec.src, behavior.new and recompiling via the "genprograms" command until you are satisfied that everything is correct. You will usually have to execute BEGIN, DECIDE, MAINSETUP, OPTIMIZE as described next in order to complete your modifications of bosdec.src and behavior.new. Each time through the "modification loop" make sure to save archive versions of bosdec.src and behavior.new (here called "bosdec.balloon" and "behavior.balloon").

------------- end of GENOPT user's activities ------------------

************************* **PART 2** *******************************
TASKS PERFORMED BY **THE END USER**

---------------- begin end user's activities -------------------

Next, establish a specific name for a case that fits within the generic class called "balloon". In this example, we use the specific name, "try4". Then execute the GENOPT processors, BEGIN, DECIDE, MAINSETUP, OPTIMIZE, as listed next. Let us assume that this run stream is based on the balloon model in which the webs form a truss-like configuration (bosdec.balloon, Figs. 2 or 5 in [1] and Table 7 in the appendix of [1]).

COMMAND MEANING OF COMMAND

**begin** (provide the starting design, etc. Input file for BEGIN = try4.BEG. If you already have a file, say, /home/progs/genopt/case/balloon/try41.starting.beg, then:
```
cp home/progs/genopt/case/balloon/try41.starting.beg try4.BEG
```
try4.BEG = Table 8 in the appendix of [1])

**decide** (provide decision variables, etc. Input file for DECIDE = try4.DEC. If you already have a file, say, /home/progs/genopt/case/balloon/try41.starting.dec, then:
```
cp home/progs/genopt/case/balloon/try41.starting.dec try4.DEC
```
try4.DEC = Table 9 in the appendix of [1])

```
mainsetup (provide analysis type, strategy, etc. Input file for
           MAINSETUP = try4.OPT. If you already have a file, say,
           /home/progs/genopt/case/balloon/try41.OPT,
           then:
           cp home/progs/genopt/case/balloon/try41.OPT try4.OPT
           try4.OPT = Table 10 in the appendix of [1].

           IMPORTANT NOTE: At first, always use ITYPE=2, not ITYPE=1 .
           That is, analyze a fixed design rather than do optimization.
           After you carefully check to see that everything in the "fixed"
           design execution is working properly, then you can do
           optimization as described below.)

optimize  (launch the OPTIMIZE run either to obtain results for
           a fixed design or to do optimization or to do a
           design sensitivity analysis. Let us assume that in
           this case we analyze a fixed design [ITYPE = 2
           in the try4.OPT file].)
```

Inspect the try4.OPM file. (No table for this file is included in [1]. It would be analogous to Table 11 in the appendix of [1].)

Suppose you used OPTIMIZE to obtain results for a fixed design, that is, the starting design that you specified in the try4.BEG file (Table 8 in] the appendix of [1]).

Next, you want to get plots of the buckling modes and the pre-buckling deformations from direct executions of the version of BIGBOSOR4 that is independent of the GENOPT system. Do the following:

First, get a plot of the "general" buckling mode: ("general" is in quotes because the lowest buckling load factor might well correspond to local buckling, not general buckling. The lowest eigenvalue could correspond either to local or to general buckling.)

```
COMMAND                  MEANING OF COMMAND

cd /home/progs/work6     (go to a working space, "work6")
bigbosor4log   (activate the BIGBOSOR4 set of commands)
cp /home/progs/genoptcase/try4.BEHX1 try4.ALL(get BIGBOSOR4 input file)
bigbosorall    (execute BIGBOSOR4: input file = try2.ALL .
                NOTE: valid input files for bigbosor4 always have the
                suffix, ".ALL")
```

Inspect the try2.OUT file. Search for the string, "EIGENVALUE(", including the trailing left parenthesis.

Next, you want to plot the buckling mode. Do the following:

COMMAND MEANING OF COMMAND

**bosorplot** (choose what to plot. Use "x" in response to prompt if you want plot on your screen. Use "p" in reponse to prompt if you want plot to be in the file called "metafile.ps".)

**gv metafile.ps** (get a plot on your screen via the "ghost view" utility, if "ghost view" is available on your workstation. Fig. 6a in [1] is an edited version of what appears on your screen.)

**cleanup** (Clean up the files generated by BIGBOSOR4)

NOTE: THE FOLLOWING LITTLE SECTION IS NOW OBSOLETE BECAUSE THERE IS NO LONGER ANY *.BEHX2 FILE GENERATED FOR THE GENERIC CASE, "balloon". Figures 9 and 10 in [1] were generated by temporarily changing the FORTRAN coding in SUBROUTINE BEHX2 in order to generate plots of the pre-buckled state of the balloon at Load Step 1 (Fig.9) and at Load Step 2 (Fig. 10).

--------------- BEGINNING OF THE LITTLE OBSOLETE SECTION --------------

Next, you want to plot the pre-buckling deformations for Load Step No. 1 and for Load Step No. 2. Do the following:

**cp /home/progs/genoptcase/try4.BEHX2 try4.ALL** (get BIGBOSOR4 input file)
**bigbosorall** (execute BIGBOSOR4: input file = try4.ALL)

Inspect the try4.OUT file. Search for the string, "LOAD STEP", and look to see how many Newton iterations were required for convergence.

Next, you want to plot the pre-buckled state for Load Step 1 or 2. Do the following:

COMMAND MEANING OF COMMAND

**bosorplot** (choose what to plot. Use "x" in response to prompt if you want plot on your screen. Use "p" in reponse to prompt if you want plot to be in the file called "metafile.ps".)

**gv metafile.ps**  (get a plot on your screen via the "ghost view"
                 utility, if "ghost view" is available on your
                 workstation. Fig. 9 in [1] is for Load Step 1)

**bosorplot** (choose what to plot. Use "x" in response to
           prompt if you want plot on your screen. Use "p"
           in reponse to prompt if you want plot to be
           in the file called "metafile.ps".)

**gv metafile.ps**  (get a plot on your screen via the "ghost view"
                 utility, if "ghost view" is available on your
                 workstation. Fig. 10 in [1] is for Load Step 2)

**cleanup**         (Clean up the files generated by BIGBOSOR4)

---------------  END OF THE LITTLE OBSOLETE SECTION -------------------

**cd /home/progs/genoptcase**     (return to genoptcase)

Next, suppose you want to optimize the specific case, "try4".
Before you do any optimization make as certain as possible that
your versions of behavior.new and bosdec.src are correct and
have been compiled by execution of the "genprograms" command.
.
First, you edit the try4.OPT file so that it appears as follows:

```
  n      $ Do you want a tutorial session and tutorial output?
     0   $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
     0   $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
     1   $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
     5   $ How many design iterations in this run (3 to 25)?
  n      $ Take "shortcuts" for perturbed designs (Y or N)?
     2   $ Choose 1 or 2 or 3 or 4 or 5 for IDESIGN
     1   $ Choose 1 or 2 or 3 or 4 or 5 for move limits, IMOVE
  y      $ Do you want default (RATIO=10) for initial move limit jump?
  y      $ Do you want the default perturbation (dx/x = 0.05)?
  n      $ Do you want to have dx/x modified by GENOPT?
  n      $ Do you want to reset total iterations to zero (Type H)?
     1   $ Choose IAUTOF= 1 or 2 or 3 or 4 or 5 or 6 to change X(i)
```

Then, type the following commands:

COMMAND                    MEANING OF COMMAND

**mainsetup** (use the try4.OPT file just listed as input)
**superopt** (launch the "global" optimizer. Use 5 OPTIMIZEs per AUTOCHANGE)

SUPEROPT will require about 20 hours (for a model with 15 modules such as that shown in Fig. 1 of [1]) if SUPEROPT runs successfully to completion (about 470 design iterations).

------------BEGINNING OF A SMALL DIGRESSION ----------------------

Inspect the try4.OPP file to see if SUPEROPT finished successfully. If it did, then there should be about 470 design iterations. Look near the end of try4.OPP to inspect the best FEASIBLE design and the best ALMOST FEASIBLE design.

Sometimes the SUPEOPT run aborts before there are about 470 design iterations. When that happens look at the end of the *.OPM file. You will probably find a message such as occurred on October 19, 2010 in the try4.OPM file:

```
 ************* ABORT **********
 0.5 x FLOUTR is greater than ROUTER
 0.5 x FLOUTR =  7.1845E+00; ROUTER =  7.1000E+00; IMODX= 1
 Put a higher lower bound on ROUTER.
 The run is now aborting.
 ********************************
```

You can either follow the directions given in the "ABORT" diagnostic by appropriately editing the try4.DEC file (input for DECIDE), executing DECIDE and then re-launching SUPEROPT.

Or you can accept the optimum design that SUPEROPT has produced up to the point when the SUPEROPT run aborted by looking near the end of the try4.OPP file and accepting either the "BEST FEASIBLE" design or the "ALMOST FEASIBLE" design, whichever you prefer.

NOTE: With "BEST FEASIBLE" designs all design margins must be greater than -0.01. With "ALMOST FEASIBLE" designs all design margins must be greater than -0.05.

If SUPEROPT quits before 470 design iterations, and you do not see an "ABORT" message at the end of the *.OPM file, then look at the end of the *.OUT file. You may find output such as the following at the end of the *.OUT file:

--- beginning of excerpt from the *.OUT file of a run that terminated early ----

```
 PRESSURE MULTIPLIER, P=   1.000000E-01, TEMPERATURE MULTIPLIER,TEMP =   1.000000E-01
 ANGULAR VELOCITY, OMEGA=   0.000000E+00

 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   0. START FACTORING AND SOLVING
 Factoring done for iteration   0; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  0
 ITERATION NO.  0 MAXIMUM DISPLACEMENT=  2.4119E+01
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   1. START FACTORING AND SOLVING
 Factoring done for iteration   1; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  1
 ITERATION NO.  1 MAXIMUM DISPLACEMENT=  1.9859E+01
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   2. START FACTORING AND SOLVING
 Factoring done for iteration   2; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  2
 ITERATION NO.  2 MAXIMUM DISPLACEMENT=  4.8352E+01
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   3. START FACTORING AND SOLVING
 Factoring done for iteration   3; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  3
 ITERATION NO.  3 MAXIMUM DISPLACEMENT=  5.8661E+01
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   4. START FACTORING AND SOLVING
 Factoring done for iteration   4; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  4
 ITERATION NO.  4 MAXIMUM DISPLACEMENT=  3.3983E+02
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   5. START FACTORING AND SOLVING
 Factoring done for iteration   5; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  5
 ITERATION NO.  5 MAXIMUM DISPLACEMENT=  4.1441E+03
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   6. START FACTORING AND SOLVING
 Factoring done for iteration   6; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  6
 ITERATION NO.  6 MAXIMUM DISPLACEMENT=  1.6590E+06
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   7. START FACTORING AND SOLVING
 Factoring done for iteration   7; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  7
 ITERATION NO.  7 MAXIMUM DISPLACEMENT=  9.7225E+10
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   8. START FACTORING AND SOLVING
 Factoring done for iteration   8; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  8
 ITERATION NO.  8 MAXIMUM DISPLACEMENT=  2.5642E+16
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   9. START FACTORING AND SOLVING
 Factoring done for iteration   9; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  9
 ITERATION NO.  9 MAXIMUM DISPLACEMENT=  8.0107E+21
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.  10. START FACTORING AND SOLVING
 Factoring done for iteration  10; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO. 10
 ITERATION NO. 10 MAXIMUM DISPLACEMENT=  7.5466E+30
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.  11. START FACTORING AND SOLVING

 ********** ALMOST SINGULAR STIFFNESS MATRIX **********
 Maximum diagonal of factored matrix at iteration 0=  2.8868E+11
 Maximum diagonal of factored matrix, current iter.=  3.3820E+27
 Newton iterations now aborting.
```

```
 ********** ALMOST SINGULAR STIFFNESS MATRIX **********
 Factoring done for iteration  11; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO. 11
 ITERATION NO. 11 MAXIMUM DISPLACEMENT=  2.2389E+37

 INITIAL LOADS TOO HIGH FOR THIS STRUCTURE. REDUCE THEM AND RERUN.

SHELL COLLAPSES AXISYMMETRICALLY AT P=0.1
```

--- end of excerpt from the *.OUT file of a run that terminated early ---

If you find this kind of output in *.OUT, then probably you should increase either the lower bound of RINNER or increase the lower bound of ROUTER, whichever of RINNER or ROUTER is closest to its lower bound. If RINNER and/or ROUTER are not decision variables then increase the value of the one of them that is not a decision variable or the values of the both of them that are not decision variables.

NOTE: RINNER should be greater than RADIUS x (pi/2)/(2 x NMODUL).
ROUTER should be greater than (RADIUS+HEIGHT) x (pi/2)/(2 x NMODUL).

---------------- END OF A SMALL DIGRESSION ----------------------

Next, obtain a plot of the objective versus design iterations.
Execute CHOOSEPLOT with the file, try4.CPL, as input:

```
try4.CPL file:
  n      $ Do you want a tutorial session and tutorial output?
  n      $ Any design variables to be plotted v. iterations (Y or N)?
  n      $ Any design margins to be plotted v. iterations (Y or N)?
  n      $ Do you want to get more plots before your next "SUPEROPT"?
```

| COMMAND | MEANING OF COMMAND |
|---|---|
| **chooseplot** | (get plot file, objective v. design iterations) |
| **diplot** | (this command produces the file, try4.5.ps) |
| **gv try4.5.ps** | (your screen shows the plot from try4.5.ps try4.5.ps = Fig. 6b in [1]) |

Next, you want to run OPTMIZE with the following input file, try4.OPT, that is, you want to do an analysis of the fixed, optimized design::

```
  n      $ Do you want a tutorial session and tutorial output?
     0   $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
     2   $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
     2   $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
```

```
    5   $ How many design iterations in this run (3 to 25)?
  n     $ Take "shortcuts" for perturbed designs (Y or N)?
    2   $ Choose 1 or 2 or 3 or 4 or 5 for IDESIGN
    1   $ Choose 1 or 2 or 3 or 4 or 5 for move limits, IMOVE
  y     $ Do you want default (RATIO=10) for initial move limit jump?
  y     $ Do you want the default perturbation (dx/x = 0.05)?
  n     $ Do you want to have dx/x modified by GENOPT?
  n     $ Do you want to reset total iterations to zero (Type H)?
    1   $ Choose IAUTOF= 1 or 2 or 3 or 4 or 5 or 6 to change X(i)
```

COMMAND                  MEANING OF COMMAND

**mainsetup** (use the try4.OPT file just listed as input: fixed design)
**optimize** (get the try4.OPM file for the fixed, optimized design)

Inspect the try4.OPM file. try4.OPM = Table 11 in the appendix of [1].

Next, save the optimum design listed near the top of Table 11 of [1].

Next, we want to save the optimized design. If you do not have an appropriate file called try4.CHG, do the following:

**change** (answer the prompts interactively)

If you do have an appropriate file, do the following:

cp /home/progs/genopt/case/balloon/try41.superopt1.chg try4.CHG
try41.superopt1.chg = Table 12 in the appendix of [1]

**change** (use the try4.CHG file as input)
**optimize** (analyze the fixed, optimized, design)

Next, we wish to do more optimization. Notice in the try4.DEC file listed as Table 9 in [1] that the variables, RINNER and ROUTER, are not decision variables. These were not used as decision variables during the first optimization in order to avoid prebuckling axisymmetric collapse, which causes an early termination of the SUPEROPT execution analogous to the type described earlier when SUPEROPT was first executed. We wish now to introduce RINNER and ROUTER as additional decision variables while keeping HEIGHT and the various wall thicknesses as decision variables. Also, we may wish to change the upper or lower bounds of one or more of the decision variables, depending on the optimum design determined after the first SUPEROPT execution. For example, if a thickness is at or near its lower bound, we will probably want to lower that lower bound in the

*.DEC file (input for DECIDE).

Do the following:

```
COMMAND                  MEANING OF COMMAND

decide          (provide decision variables, etc. Input file for
                 DECIDE = try4.DEC. If you already have a file, say,
                 /home/progs/genopt/case/balloon/try41.dec2,
                 then:
                 cp /home/progs/genopt/case/balloon/try41.dec2 try4.DEC
                 try4.DEC = Table 13 in the appendix of [1])

mainsetup       (use try4.OPT with NPRINT = 0 and ITYPE = 1 as input)
superopt        (launch the "global" optimizer. Use 5 OPTIMIZEs per
                 AUTOCHANGE)
```

SUPEROPT will require about 20 hours (for a model with 15 modules) if it runs successfully to completion (about 470 design iterations).

Inspect the try4.OPP file to see if SUPEROPT finished successfully. If it did, then there should be about 470 design iterations. Look near the end of try4.OPP to inspect the best FEASIBLE design and the best ALMOST FEASIBLE design.

Next, obtain a plot of the objective versus design iterations. Execute CHOOSEPLOT with the file, try4.CPL, as input:

```
try4.CPL
  n      $ Do you want a tutorial session and tutorial output?
  n      $ Any design variables to be plotted v. iterations (Y or N)?
  n      $ Any design margins to be plotted v. iterations (Y or N)?
  n      $ Do you want to get more plots before your next "SUPEROPT"?
```

```
COMMAND                  MEANING OF COMMAND

chooseplot      (get plot file, objective v. design iterations)
diplot          (this command produces the file, try4.5.ps)
gv try4.5.ps    (your screen shows the plot from try4.5.ps)
                try4.5.ps = Fig. 7 in [1]
```

Next, you want to run OPTMIZE with the following input file, try4.OPT:

```
  n      $ Do you want a tutorial session and tutorial output?
     0   $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
     2   $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
     2   $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
```

```
    5   $ How many design iterations in this run (3 to 25)?
 n      $ Take "shortcuts" for perturbed designs (Y or N)?
    2   $ Choose 1 or 2 or 3 or 4 or 5 for IDESIGN
    1   $ Choose 1 or 2 or 3 or 4 or 5 for move limits, IMOVE
 y      $ Do you want default (RATIO=10) for initial move limit jump?
 y      $ Do you want the default perturbation (dx/x = 0.05)?
 n      $ Do you want to have dx/x modified by GENOPT?
 n      $ Do you want to reset total iterations to zero (Type H)?
    1   $ Choose IAUTOF= 1 or 2 or 3 or 4 or 5 or 6 to change X(i)
```

| COMMAND | MEANING OF COMMAND |
|---|---|
| **mainsetup** | (use the try4.OPT file just listed as input: fixed design) |
| **optimize** | (get the try4.OPM file for the fixed, optimized design) |

Inspect the try4.OPM file. try4.OPM = Table 14 in the appendix of [1].

Next, save the optimum design listed near the top of Table 14 of [1].

Next, we want to save the optimized design. If you do not have an appropriate file called try4.CHG, do the following:

**change** (answer the prompts interactively)

If you do have an appropriate file, do the following:

cp /home/progs/genopt/case/balloon/try41.superopt2.chg try4.CHG
  try41.superopt2.chg = Table 15 in the appendix of [1].

**change** (use the try4.CHG file as input)
**optimize** (analyze the fixed, optimized, design)

Next, we wish to do still more optimization, mainly because the variable, HEIGHT, in the previous optimization reached its lower bound: 50 inches. Hence, we decrease the lower bound from 50 to 40 inches. Do the following:

| COMMAND | MEANING OF COMMAND |
|---|---|
| **decide** | (provide decision variables, etc. Input file for DECIDE = try4.DEC. If you already have a file, say, /home/progs/genopt/case/balloon/try41.dec3, then: cp /home/progs/genopt/case/balloon/try41.dec3 try4.DEC try4.DEC = Table 16 in in the appendix of [1]) |

**mainsetup** (use try4.OPT with NPRINT = 0 and ITYPE = 1 as input)
**superopt** (launch the "global" optimizer. Use 5 OPTIMIZEs per AUTOCHANGE)

SUPEROPT will require about 20 hours (for a model with 15 modules) if it runs successfully to completion (about 470 design iterations).

Inspect the try4.OPP file to see if SUPEROPT finished successfully. If it did, then there should be about 470 design iterations. Look near the end of try4.OPP to inspect the best FEASIBLE design and the best ALMOST FEASIBLE design. It turns out in this case that the best design is the same as the optimum design obtained after the 2nd execution of SUPEROPT (Table 14 in the appendix of [1]).

Next, obtain a plot of the objective versus design iterations. Execute CHOOSEPLOT with the file, try4.CPL, as input:

```
try4.CPL
  n      $ Do you want a tutorial session and tutorial output?
  n      $ Any design variables to be plotted v. iterations (Y or N)?
  n      $ Any design margins to be plotted v. iterations (Y or N)?
  n      $ Do you want to get more plots before your next "SUPEROPT"?
```

COMMAND MEANING OF COMMAND

**chooseplot** (get plot file, objective v. design iterations)
**diplot** (this command produces the file, try4.5.ps)
**gv try4.5.ps** (your screen shows the plot from try4.5.ps try4.5.ps = Fig. 8 in [1])

Next, you want to get a plot of the buckling mode of the optimized design from a direct execution of the version of BIGBOSOR4 that is independent of the GENOPT system.
Do the following:

COMMAND MEANING OF COMMAND

**cd /home/progs/work6** (go to a working space, "work6")
**bigbosor4log** (activate the BIGBOSOR4 set of commands)
**cp /home/progs/genoptcase/try4.BEHX1 try4.ALL** (get BIGBOSOR4 input file)
**bigbosorall** (execute BIGBOSOR4: input file = try4.ALL)

Inspect the try4.OUT file. Search for the string, "EIGENVALUE(", including the trailing left parenthesis.

Next, you want to plot the buckling mode. Do the following:

COMMAND MEANING OF COMMAND

**bosorplot** (choose what to plot. Use "x" in response to prompt if you want plot on your screen. Use "p" in reponse to prompt if you want plot to be in the file called "metafile.ps".)

**gv metafile.ps** (get a plot on your screen via the "ghost view" utility, if "ghost view" is available on your workstation. metafile.ps, with some editing, produces Fig. 11 in [1].)

**cleanup** (Clean up the files generated by BIGBOSOR4)

The input file for BIGBOSOR4 generated by SUBROUTINE BOSDEC is, in the specific case, “try4”, called “try4.BEHX1”. This file does not have the usual complete annotation for each datum that is typical of ordinary BIGBOSOR4 input files called “*.ALL”. One of the important things that the command, “cleanup”, does is to produce a properly annotated input file, “*.ALL”, for BIGOBOSOR4. Before the “cleanup” command just listed the input files, try4.BEHX1 = try4.ALL, have the following lines (corresponding to the initial input data followed by the input data only for the first shell segment in this example):

```
general buckling,  12-module model (INDIC=1) ixprism
  6.000000E+03 $ AXIALL = axial length of cyl.
    1  1, 0,  64 $ INDIC,NPRT,ISTRESS,NSEG
 H   $ Segment number      1     1     1     1
  31, 3,   1 $ NMESH,NTYPEH,NSHAPE
  3.819719E+01  1.877272E+02  5.050148E+01  1.877272E+02 $ R1,Z1,R2,Z2
 0, 3,   2.807000E-02 $ IMP,NTYPEZ,ZVAL
 N  $ do not print r(s), etc.
    0  0.000000E+00   0   3 $ NRINGS,K,LINTYP,IDISAB
    3    2    0    0    1  $ NLTYPE,NPSTAT,NLOAD(1),NLOAD(2),NLOAD(3)
  0.000000E+00  0.000000E+00  $ PN(1),PN(2)
    1    1   31  $ NTYPE,IPOINT(1),IPOINT(2)
    2    1    1    0    0  $ NTSTAT,NTGRAD,NLOAD(1),NLOAD(2),NLOAD(3)
  0.000000E+00  0.000000E+00  $ T1(1),T1(2)
    1    1   31  $ NTYPE,IPOINT(1),IPOINT(2)
    3    2    0    0    1  $ NLTYPE,NPSTAT,NLOAD(1),NLOAD(2),NLOAD(3)
  0.000000E+00  0.000000E+00  $ PN(1),PN(2)
    1    1   31  $ NTYPE,IPOINT(1),IPOINT(2)
```

```
   2    1    1    0    0  $ NTSTAT,NTGRAD,NLOAD(1),NLOAD(2),NLOAD(3)
 -1.114082E+02 -1.114082E+02  $ T1(1),T1(2)
   1    1   31  $ NTYPE,IPOINT(1),IPOINT(2)
  4  1 $ NWALL,NLAYER
  1 $ layer index
 Y  $ is this a new layer type?
  5.614000E-02  0.000000E+00  2 $ thickness,angle,material
 Y  $ Is this material new?
  4.351000E+05  4.351000E+05  1.673460E+05  3.000000E-01 $ E1,E2,G12,NU
  1.000000E-10  1.000000E-04  0.000000E+00  1.000000E-01 $
A1,A2,TEMPTUR,DENS
  1.000000E+04  1.000000E+04  1.000000E+04 $ S(1),S(2),S(3)
  1.000000E+04  1.000000E+04 $ S(4),S(5)
 0  $ no additional smeared stiffeners
 Y  $ do you want output for all nodes?
 N  $ do you want to print out Cij?
 N  $ do you want to print out loads?
(many, many more lines are omitted here in order to save space)
```

After the "cleanup" command the same input data are arranged in the completely annotated form, as follows:

```
general buckling,  12-module model (INDIC=1) ixprism
   6000.000     $ AXIALL= length of the prismatic shell
        1       $ INDIC = analysis type indicator
        1       $ NPRT = output options (1=minimum, 2=medium, 3=maximum)
        0       $ ISTRES= output control (0=resultants, 1=sigma, 2=epsilon)
       64       $ NSEG  = number of shell segments (less than 295)
      H         $
      H         $ SEGMENT NUMBER    1    1    1    1    1    1    1    1
      H         $ NODAL POINT DISTRIBUTION FOLLOWS...
       31       $ NMESH = number of node points (5 = min.; 98 = max.)(  1)
        3       $ NTYPEH= control integer (1 or 3) for nodal point spacing
      H         $ REFERENCE SURFACE GEOMETRY FOLLOWS...
        1       $ NSHAPE= indicator (1,2 or 4) for geometry of meridian
   38.19719     $ R1     = radius at beginning of segment (see p. 66)
   187.7272     $ Z1     = global axial coordinate at beginning of segment
   50.50148     $ R2     = radius at end of segment
   187.7272     $ Z2     = global axial coordinate at end of segment
      H         $ IMPERFECTION SHAPE FOLLOWS...
        0       $ IMP   = indicator for imperfection (0=none, 1=some)
      H         $ REFERENCE SURFACE LOCATION RELATIVE TO WALL
        3       $ NTYPEZ= control (1 or 3) for reference surface location
  0.2807000E-01 $ ZVAL  = distance from leftmost surf. to reference surf.
      N         $ Do you want to print out r(s), r'(s), etc. for this segment?
      H         $ DISCRETE RING INPUT FOLLOWS...
        0       $ NRINGS= number (max=20) of discrete rings in this segment
   0.000000     $ K=elastic foundation modulus (e.g. lb/in**3)in this seg.
```

```
    H         $ LINE LOAD INPUT FOLLOWS...
       0      $ LINTYP= indicator (0, 1, 2 or 3) for type of line loads
    H         $ DISTRIBUTED LOAD INPUT FOLLOWS...
       3      $ IDISAB= indicator (0, 1, 2 or 3) for load set A and B
    H         $ SURFACE LOAD INPUT FOR LOAD SET "A" FOLLOWS
       3      $ NLTYPE=control (0,1,2,3) for type of surface loading
       2      $ NPSTAT= number of meridional callouts for surface loading
       0      $ NLOAD(1)=indicator for meridional traction (0=none, 1=some)
       0      $ NLOAD(2)=indicator for circumferential traction
       1      $ NLOAD(3)=indicator for normal pressure     (0=none, 1=some)
 0.000000     $ PN(i)   = normal pressure (p.74) at ith callout, PN(  1)
 0.000000     $ PN(i)   = normal pressure (p.74) at ith callout, PN(  2)
       1      $ NTYPE = control for meaning of loading callout (2=z, 3=r)
       1      $ IPOINT(I)=segment nodal point of the Ith callout(  1)
      31      $ IPOINT(I)=segment nodal point of the Ith callout(  2)
       2      $ NTSTAT= number of meridional callouts for temperature
       1      $ NTGRAD=control for type of thermal gradient thru thickness
       1      $ NLOAD(1)=indicator for temperature coef. T1 (0=none, 1=some)
       0      $ NLOAD(2)=indicator for temperature coef. T2 (0=none, 1=some)
       0      $ NLOAD(3)=indicator for temperature coef. T3 (0=none, 1=some)
 0.000000     $ T1    = temperature factor at Ith meridional callout, T1(  1)
 0.000000     $ T1    = temperature factor at Ith meridional callout, T1(  2)
       1      $ NTYPE = control for meaning of callout (2=z, 3=r)
       1      $ IPOINT(I)=segment nodal point of the Ith callout(  1)
      31      $ IPOINT(I)=segment nodal point of the Ith callout(  2)
    H         $ SURFACE LOAD INPUT FOR LOAD SET "B" FOLLOWS
       3      $ NLTYPE=control (0,1,2,3) for type of surface loading
       2      $ NPSTAT= number of meridional callouts for surface loading
       0      $ NLOAD(1)=indicator for meridional traction (0=none, 1=some)
       0      $ NLOAD(2)=indicator for circumferential traction
       1      $ NLOAD(3)=indicator for normal pressure     (0=none, 1=some)
 0.000000     $ PN(i)   = normal pressure (p.74) at ith callout, PN(  1)
 0.000000     $ PN(i)   = normal pressure (p.74) at ith callout, PN(  2)
       1      $ NTYPE = control for meaning of loading callout (2=z, 3=r)
       1      $ IPOINT(I)=segment nodal point of the Ith callout(  1)
      31      $ IPOINT(I)=segment nodal point of the Ith callout(  2)
       2      $ NTSTAT= number of meridional callouts for temperature
       1      $ NTGRAD=control for type of thermal gradient thru thickness
       1      $ NLOAD(1)=indicator for temperature coef. T1 (0=none, 1=some)
       0      $ NLOAD(2)=indicator for temperature coef. T2 (0=none, 1=some)
       0      $ NLOAD(3)=indicator for temperature coef. T3 (0=none, 1=some)
-111.4082     $ T1    = temperature factor at Ith meridional callout, T1(  1)
-111.4082     $ T1    = temperature factor at Ith meridional callout, T1(  2)
       1      $ NTYPE = control for meaning of callout (2=z, 3=r)
       1      $ IPOINT(I)=segment nodal point of the Ith callout(  1)
      31      $ IPOINT(I)=segment nodal point of the Ith callout(  2)
    H         $ SHELL WALL CONSTRUCTION FOLLOWS...
       4      $ NWALL=index (1, 2, 4, 5, 6, 7, 8, 9, 10) for wall construction
       1      $ number of layers in the wall
       1      $ layer index (1,2,...), for layer no.(  1)
    Y         $ Is this a new layer type?
0.5614000E-01 $ thickness for layer index no.(  1)
 0.000000     $ winding angle (deg.) for layer index no.(  1)
```

```
        2      $ material index (1,2,...) for layer index no.(  1)
      Y        $ Next material type... Is material new for material type(  2)
   435100.0    $ modulus in the fiber direction, E1(  2)
   435100.0    $ modulus transverse to fibers,   E2(  2)
   167346.0    $ in-plane shear modulus,          G(  2)
  0.3000000    $ small Poisson's ratio,          NU(  2)
  0.1000000E-09 $ thermal expansion along fibers, A1(  2)
  0.1000000E-03 $ transverse thermal expansion,   A2(  2)
   0.000000    $ residual stress temperature (positive),TEMPTUR(  2)
  0.1000000    $ mass density (e.g. lb-sec**2/in. Aluminum=.00025), DENS(  2)
   10000.00    $ maximum tensile stress along fibers, matl(  2)
   10000.00    $ max compressive stress along fibers, matl(  2)
   10000.00    $ max tensile stress normal to fibers, matl(  2)
   10000.00    $ max compress stress normal to fibers,matl(  2)
   10000.00    $ maximum  shear  stress  in  material type(  2)
        0      $ NRS  = control (0 or 1) for addition of smeared stiffeners
      Y        $ Do you want output for all the nodal points in Segment(  1)
      N        $ Do you want to print out the C(i,j) at meridional stations?
      N        $ Do you want to print out distributed loads along meridian?
      H        $
(many, many more lines are omitted here in order to save space)
```

It is easier to find a specific input datum when the BIGBOSOR4 input file is in the properly annotated form as just listed.

Next, we want to get more than one buckling eigenvector. In this particular case the lowest eigenvalue happens to correspond to general buckling (Fig. 11 of [1]), and we want next to see what some local buckling modes look like. Therefore, we do the following:

Edit the "cleaned up" (properly annotated) file, try4.ALL, as follows: Search for the string, "NVEC". (NOTE: “NVEC” does not appear in the properly annotated list above because it occurs in a later section of the try4.ALL file not included there in order to save space.)

Change NVEC from 1 to 10. Then do the following:

COMMAND                    MEANING OF COMMAND

**bigbosorall**    (execute BIGBOSOR4: input file=try4.ALL)

Inspect the try4.OUT file. Search for the string, "EIGENVALUE(", including the trailing left parenthesis. Just above, you will see 10 eigenvalues, all clustered very close together. The first eigenvalue in this particular case  corresponds to global buckling as displayed in Fig. 11 of [1]. Eigenvalues 2 - 9 correspond to various combinations of local buckling and general buckling, as shown, for examples, in Figs. 13 and 14 of [1]. The 10th eigenvalue in this particular case corresponds to almost pure local buckling, as shown in Fig. 12 of [1].

Next, you want to plot some of the "higher" buckling modes. Do the following:

| COMMAND | MEANING OF COMMAND |
|---|---|
| **bosorplot** | (choose the 2nd eigenvalue. Use "x" in response to prompt if you want plot on your screen. Use "p" in reponse to prompt if you want plot to be in the file called "metafile.ps".) |
| **gv metafile.ps** | (get a plot on your screen via the "ghost view" utility, if "ghost view" is available on your workstation. metafile.ps, with some editing, produces Fig. 13 in [1].) |
| **bosorplot** | (choose the 3rd eigenvalue. Use "x" in response to prompt if you want plot on your screen. Use "p" in reponse to prompt if you want plot to be in the file called "metafile.ps".) |
| **gv metafile.ps** | (get a plot on your screen via the "ghost view" utility, if "ghost view" is available on your workstation. metafile.ps, with some editing, produces Fig. 14 in [1].) |
| **bosorplot** | (choose the 10th eigenvalue. Use "x" in response to prompt if you want plot on your screen. Use "p" in reponse to prompt if you want plot to be in the file called "metafile.ps".) |
| **gv metafile.ps** | (get a plot on your screen via the "ghost view" utility, if "ghost view" is available on your workstation. metafile.ps, with some editing, produces Fig. 12 in [1].) |
| **cleanup** | (Clean up the files generated by BIGBOSOR4) |
| **cd /home/progs/genoptcase** | (return to genoptcase) |

Next, use GENOPT/BIGBOSOR4 to do "design sensitivity" analyses (ITYPE = 3 in the *.OPT file).

Suppose you want to obtain a plot of the buckling margin as a function of the radius, ROUTER, of the local outward "bulges" in the outer membrane.
Ordinarily, the optimum value of ROUTER should fall approximately in the middle of the range of ROUTER explored in this "design sensitivity" analysis (ITYPE=3 in the try4.OPT file). The try4.OPT file should be as follows in this particular case:

```
    n       $ Do you want a tutorial session and tutorial output?
       0    $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
       2    $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
       3    $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
       3    $ Choose a design variable (1, 2, 3, ...), IBVAR
  8.95000   $ Starting value of the design parameter, VARBEG
 12.95000   $ Ending value of the design parameter, VAREND
    n       $ Do you want to use the default for the number of steps?
      11    $ Number of steps from VARBEG to VAREND. NSTEPS
```

COMMAND MEANING OF COMMAND

**mainsetup** (use the try4.OPT file just listed as input)
**optimize** (get the try4.OPM file for the range of ROUTER)

Next, obtain a plot of the buckling and stress margins versus ROUTER:

**chooseplot** (choose what to plot versus the range of ROUTER.
The input file in this case, try4.CPL, follows:)

```
    n       $ Do you want a tutorial session and tutorial output?
    y       $ Any design margins to be plotted v. iterations (Y or N)?
       1    $ Choose a margin to be plotted v. iterations (1,2,3,..)
    y       $ Any more margins to be plotted (Y or N) ?
       2    $ Choose a margin to be plotted v. iterations (1,2,3,..)
    y       $ Any more margins to be plotted (Y or N) ?
       4    $ Choose a margin to be plotted v. iterations (1,2,3,..)
    y       $ Any more margins to be plotted (Y or N) ?
       7    $ Choose a margin to be plotted v. iterations (1,2,3,..)
    y       $ Any more margins to be plotted (Y or N) ?
       9    $ Choose a margin to be plotted v. iterations (1,2,3,..)
    y       $ Any more margins to be plotted (Y or N) ?
      12    $ Choose a margin to be plotted v. iterations (1,2,3,..)
    y       $ Any more margins to be plotted (Y or N) ?
      14    $ Choose a margin to be plotted v. iterations (1,2,3,..)
    n       $ Any more margins to be plotted (Y or N) ?
       1    $ Give maximum value (positive) to be included in plot frame.
```

COMMAND                    MEANING OF COMMAND

**diplot**          (diplot creates try4.3.ps and try4.5.ps. The design margins chosen in CHOOSEPLOT" are in the file, try4.3.ps and the objective versus ROUTER is in the file, try4.5.ps.)

**gv try4.3.ps**    (this command shows the plot of the design margins versus ROUTER for 8.95 < ROUTER < 12.95. The plot is displayed in Fig. 17 in [1].)

Do an analogous "design sensitivity" study (ITYPE = 3 in the *.OPT file) with use of RINNER as the changing decision variable. The input to MAINSETUP is as follows:

```
    n       $ Do you want a tutorial session and tutorial output?
       0    $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
       2    $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
       3    $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
       2    $ Choose a design variable (1, 2, 3, ...), IBVAR
 6.280400   $ Starting value of the design parameter, VARBEG
 7.000000   $ Ending value of the design parameter, VAREND
    n       $ Do you want to use the default for the number of steps?
      11    $ Number of steps from VARBEG to VAREND. NSTEPS
```

The relevant plot is Fig. 16 in [1].

Do an analogous "design sensitivity" study (ITYPE = 3 in the *.OPT file) with use of HEIGHT as the changing decision variable. The input to MAINSETUP is as follows:

```
    n       $ Do you want a tutorial session and tutorial output?
       0    $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
       2    $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
       3    $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
       1    $ Choose a design variable (1, 2, 3, ...), IBVAR
 40.00000   $ Starting value of the design parameter, VARBEG
 60.00000   $ Ending value of the design parameter, VAREND
    n       $ Do you want to use the default for the number of steps?
      11    $ Number of steps from VARBEG to VAREND. NSTEPS
```

The relevant plot is Fig. 15 in [1].

Up to this point we have used POUTER = 5.0 psi and a factor of safety of 3.0 for buckling. Although the "real" case has POUTER = 15.0 psi and factor of safety for buckling of 1.0, we have used the lower POUTER and the higher factor of safety in order to avoid pre-buckling collapse during optimization cycles. Hence, we must next check the optimized design with the use of POUTER = 15.0 psi and factor of safety for buckling of 1.0. Do the following:

COMMAND MEANING OF COMMAND

**cleanspec** (clean up the files with the specific name, try4)

Edit the try4.BEG file (Table 8 in the appendix of [1]), making the following changes:

The following lines in the try4.BEG file:

```
  5.00000   $ pressure outside the outer membrane: POUTER(  1)
 1.000000   $ allowable for general buckling load factor: GENBUKA(  1)
 3.000000   $ general buckling factor of safety: GENBUKF(  1)
```

should be changed to the following lines in the new try4.BEG file:

```
 15.00000   $ pressure outside the outer membrane: POUTER(  1)
 1.000000   $ allowable for general buckling load factor: GENBUKA(  1)
 1.000000   $ general buckling factor of safety: GENBUKF(  1)
```

Then, do the following:

COMMAND MEANING OF COMMAND

**begin** (input is the new try4.BEG file)

**change** (input is the file, try41.superopt2.chg:
cp /home/progs/genopt/case/balloon/try41.superopt2.chg try4.CHG
Table 15 in the appendix of [1])

**decide** (input is the file, try41.dec2:
cp /home/progs/genopt/case/balloon/try41.dec2 try4.DEC
Table 13 in the appendix of [1])

**mainsetup** (input is the file, try41.OPT:
cp /home/progs/genopt/case/balloon/try41.OPT try4.OPT
Table 10 in [1] with NPRINT=2 and ITYPE = 2 for the analysis of the fixed, optimized design)

**optimize** (run OPTIMIZE and inspect the try4.OPM file.
try4.OPM = try41.optimum.POUTER15.opm = Table 17 in the appendix of [1])

Next, you want to get a plot of the buckling mode of the optimized design from a direct execution of the version of BIGBOSOR4 that is independent of the GENOPT system.
Do the following:

```
COMMAND                        MEANING OF COMMAND

cd /home/progs/work6    (go to a working space, "work6")
bigbosor4log    (activate the BIGBOSOR4 set of commands)
cp /home/progs/genoptcase/try4.BEHX1 try4.ALL   (get BIGBOSOR4 input file)
bigbosorall     (execute BIGBOSOR4: input file = try4.ALL)
```

Inspect the try4.OUT file. Search for the string, "EIGENVALUE(", including the trailing left parenthesis.

Next, you want to plot the buckling mode. Do the following:

```
COMMAND                        MEANING OF COMMAND

bosorplot        (choose what to plot. Use "x" in response to
                  prompt if you want plot on your screen. Use "p"
                  in reponse to prompt if you want plot to be
                  in the file called "metafile.ps".)

gv metafile.ps   (get a plot on your screen via the "ghost view"
                  utility, if "ghost view" is available on your
                  workstation. metafile.ps, with some editing,
                  produces results essentially the same as
                  Fig. 11 in [1].)

cleanup          (Clean up the files generated by BIGBOSOR4)

cd /home/progs/genoptcase       (return to genoptcase)
```

**WHAT IF A SUPEROPT RUN ABORTS BEFORE REACHING ABOUT 470 DESIGN ITERATIONS?**

NOTE: THE FOLLOWING DESCRIPTION APPLIES TO THE STRATEGY IN EXISTENCE BEFORE "**Try no. 3**" WAS ADDED (See **Item 9** in **Section 8** of [1]).

The following output from a *.OPM file shows what is listed in the *.OPM file during processing that is at first normal, but then "double" failure to achieve nonlinear pre-buckling convergence leads to an unrecoverable situation and therefore the SUPEROPT run aborts. Advice is given to the user about how to proceed.

**Output from a *.OPM file before at the moment of an unrecoverable error**

---

Part of the try4.OPM file during optimization cycles

Ths excerpt from the try4.OPM file generated for optimization (ITYPE = 1 in the try4.OPT file) shows output when unrecoverable pre-buckling nonlinear collapse occurs.

```
==========================================================
 STRUCTURAL ANALYSIS FOR DESIGN ITERATION NO.   3:

  STRUCTURAL ANALYSIS WITH UNPERTURBED DECISION VARIABLES
 VAR. DEC. ESCAPE LINK. LINKED  LINKING   LOWER     CURRENT    UPPER      DEFINITION
  NO. VAR.  VAR.  VAR.    TO   CONSTANT   BOUND      VALUE     BOUND
   1   Y     N     N      0    0.00E+00  2.00E+01 7.6412E+01  1.20E+02  height from inner to outer
membranes: HEIGHT
   2   N     N     N      0    0.00E+00  0.00E+00 1.6000E+01  0.00E+00  radius of curvature of
inner membrane: RINNER
   3   N     N     N      0    0.00E+00  0.00E+00 3.0000E+01  0.00E+00  radius of curvature of
outer membrane: ROUTER
   4   Y     N     N      0    0.00E+00  3.00E-02 9.4838E-02  3.00E-01  thickness of the inner
curved membrane: TINNER
   5   Y     N     N      0    0.00E+00  3.00E-02 1.7850E-01  3.00E-01  thickness of the outer
curved membrane: TOUTER
   6   Y     N     N      0    0.00E+00  3.00E-02 1.4529E-01  3.00E-01  thickness of inner truss-
core segment: TFINNR
   7   Y     Y     N      0    0.00E+00  3.00E-02 3.1920E-02  3.00E-01  thickness of the outer
truss segment: TFOUTR
   8   Y     Y     N      0    0.00E+00  3.00E-02 1.8814E-01  3.00E-01  thickness of each truss-
core web: TFWEBS

 Newton iterations required to solve the nonlinear
 axisymmetric pre-buckling equilibrium state for the
 "fixed" loads, PINNER=  0.0000E+00, PMIDDL=  6.0000E+01, DELTAT= -
1.0274E+02
 LOAD STEP   Newton iterations  Maximum displacement
     1            11             4.702700E-01
     2             2             9.389533E-01
     3             2             1.406679E+00
     4             2             1.873601E+00
     5             2             2.339956E+00
     6             2             2.805957E+00
     7             2             3.271797E+00
     8             2             3.737653E+00
     9             1             4.203687E+00
    10             2             4.670050E+00
 WRDCOL=
 IMODX=0 for current design,
 IMODX=1 for perturbed design: IMODX= 0

 Changes in temperature required to create 2 total axial loads:
```

```
 1. Change in temperature required to create the axial thermal
    strain that generates the axial tension due to closing the
    two ends of the pressurized volume (PMIDDL=  6.0000E+01)
    between the inner and outer walls of the balloon in
    Load Step No. 1:                                DELTAT= -1.0274E+02

 2. Change in temperature required to simulate the Poisson
    axial expansion caused by the application of the outer
    pressure, POUTER =  5.0000E+00 in Load Step No. 2:  DELT=  0.0000E+00


GENERAL BUCKLING LOAD FACTORS AND MODES (BEHX1)
       3.0538E+00(     1)
Critical buckling load factor, GENBUK=  3.0538E+00
Critical number of axial half-waves, NWVCRT=    1

Differences in the resultants along the axis of the prismatic
balloon for each segment, J, of the first module:
[N2VAR(J) for the total load] - [N2FIX(J) for the fixed load]=
N2DIFF(J),J=1,6)= -3.0402E+01 -6.1806E+01 -2.0300E+02 -1.1364E+01
                  -4.6538E+01 -3.0465E+01
N2VAR(J) (total load) are the resultants from Load Step No. 2.
N2FIX(J) (fixed load) are the resultants from Load Step No. 1.
NOTE: The stresses used as behavioral constraints are
      computed from N2VAR(J)/thickness(J). These stresses are
      lower than those computed from N2FIX(J)/thickness(J).

Newton iterations required to solve the nonlinear
axisymmetric pre-buckling equilibrium state for the
"fixed" loads (PINNER, PMIDDL, DELTAT):       ITER=  1
Maximum displacement,                         FMAX=  4.6700E+00
Newton iterations required to solve the nonlinear
axisymmetric pre-buckling equilibrium state for the
total loads (PINNER, PMIDDL, DELTAT, POUTER): ITER=  2
Maximum displacement,                         FMAX=  3.4400E+00


***** RESULTS FOR LOAD SET NO.  1  ******

******* NOTE ******* NOTE ******* NOTE ****** NOTE ******
The phrase, "NOT APPLY", for MARGIN VALUE means that that
particular margin value is exactly zero.
*** END NOTE *** END NOTE *** END NOTE *** END NOTE *****


***** RESULTS FOR LOAD SET NO.  1  ******
```

```
MARGINS LESS THAN CONMAX-1 CORRESPONDING TO CURRENT DESIGN

MARGIN CURRENT
NO.     VALUE           DEFINITION
 1    1.792E-02  (GENBUK(1 )/GENBUKA(1 )) / GENBUKF(1 )-1; F.S.=  3.00
 2   -2.373E-02  (STRM1A(1 ,1 )/STRM1(1 ,1 )) / STRM1F(1 ,1 )-1; F.S.=  1.00
 3    3.257E-01  (STRM1A(1 ,3 )/STRM1(1 ,3 )) / STRM1F(1 ,3 )-1; F.S.=  1.00
 4    3.136E-02  (STRM2A(1 ,1 )/STRM2(1 ,1 )) / STRM2F(1 ,1 )-1; F.S.=  1.00
 5    3.552E-01  (STRM2A(1 ,3 )/STRM2(1 ,3 )) / STRM2F(1 ,3 )-1; F.S.=  1.00
 6    2.832E-02  (STRM3A(1 ,1 )/STRM3(1 ,1 )) / STRM3F(1 ,1 )-1; F.S.=  1.00
 7    3.536E-01  (STRM3A(1 ,3 )/STRM3(1 ,3 )) / STRM3F(1 ,3 )-1; F.S.=  1.00

***************************************************
******************** DESIGN OBJECTIVE *******************
*************                             ***********
   CURRENT VALUE OF THE OBJECTIVE FUNCTION:

VAR.   CURRENT
NO.     VALUE           DEFINITION
 1    9.277E+01  weight/length of the balloon: WEIGHT

*************                          ************
******************** DESIGN OBJECTIVE *******************
*****************************************************
WRDCOL=
IMODX=0 for current design,
IMODX=1 for perturbed design: IMODX= 1
WRDCOL=
IMODX=0 for current design,
IMODX=1 for perturbed design: IMODX= 1
Decision variable candidates, HEIGHT,RINNER,ROUTER,TINNER,TOUTER=
 7.641174E+01  1.600000E+01  3.000000E+01  9.483758E-02  1.874235E-01
TFINNR,TFOUTR,TFWEBS=  1.4529E-01  3.1920E-02  1.8814E-01

****** CHANGE FROM 10 TO 1 LOAD STEPS ******
INITIAL LOADS TOO HIGH FOR THIS STRUCT
Changing from 10 to 1 steps: IMODX= 1
********************************
WRDCOL=
IMODX=0 for current design,
IMODX=1 for perturbed design: IMODX= 1
Decision variable candidates, HEIGHT,RINNER,ROUTER,TINNER,TOUTER=
 7.641174E+01  1.600000E+01  3.000000E+01  9.483758E-02  1.784986E-01
TFINNR,TFOUTR,TFWEBS=  1.5255E-01  3.1920E-02  1.8814E-01

****** CHANGE FROM 10 TO 1 LOAD STEPS ******
INITIAL LOADS TOO HIGH FOR THIS STRUCT
Changing from 10 to 1 steps: IMODX= 1
********************************
```

```
 Decision variable candidates, HEIGHT,RINNER,ROUTER,TINNER,TOUTER=
  7.641174E+01  1.600000E+01  3.000000E+01  9.483758E-02  1.784986E-01
 TFINNR,TFOUTR,TFWEBS=  1.5255E-01  3.1920E-02  1.8814E-01

 ************* ABORT **********
 INITIAL LOADS TOO HIGH FOR THIS STRUCT
 This is an unrecoverable error because we have already
 tried and failed to obtain nonlinear pre-buckling convergence
 by changing from a nonlinear solution with 10 load steps to
 a nonlinear solution with 1 load step. That strategy just
 failed. You may well have performed enough design iterations
 to have a good optimum design now. Look near the end of the
 *.OPP file at the "FEASIBLE" and "ALMOST FEASIBLE" designs.
 If you are not satisfied that you have performed enough
 design iterations, then look at the thicknesses of the
 various segments. If any thicknesses seem too small, then
 increase them and also increase the corresponding lower
 bounds of them. Another thing you can try that has worked
 for Bushnell is to look near the end of the *.OPM file for
 the last successfully obtained design. Use the GENOPT
 processor, CHANGE, to reset the values of the decision
 variables to those of the last successfully obtained design
 and then launch a new execution of SUPEROPT, probably
 leaving the lower bounds unchanged, or perhaps also changing
 them if you wish (before launching SUPEROPT, of course).
 The run is now aborting: IMODX= 1
 ********************************
-------------------------------------------------------------------


---------------- end of end user's activities -------------------

=====================================================================
```

**Table 5 The file, behavior.balloon, which is the "fleshed out" version of the skeletal behavior.new that is automatically produced by GENTEXT. It is the duty of the GENOPT user to do the "fleshing out". The code supplied by the GENOPT user is in bold face.**

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

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 "balloon".

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 "balloon" 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.)

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 [2c] Bushnell, D., "Use of GENOPT and a BIGBOSOR4 "huge"
C torus" model to optimize a typical weld land and weld land
C edge stringers in a previously optimized internally
C stiffened cylindrical shell without weld lands",
C unpublished report to NASA Langley Research, May 15, 2009

C [2d] Bushnell, D., "Use of GENOPT and BIGBOSOR4 to obtain
C optimum designs of a cylindrical shell with a composite
C truss-core sandwich wall", unpublished report to NASA
C Langley Research Center, Hampton, VA, June 20, 2009

C [2e] Bushnell, D., "Use of GENOPT and BIGBOSOR4 to obtain
C an optimum design of a deep submergence tank", unpublished
C report to the DOER company, Alameda, CA, June 30, 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, Ref.[2b]

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

C [15] Bushnell, D., the case ..genopt/case/wavycyl, Ref.[2]

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

C [17] Bushnell, D., the case ..genopt/case/weldland, Ref.[2c]

C [18] Bushnell, D., the case ..genopt/case/trusscomp,Ref.[2d]

C [19] Bushnell, D., the case ..genopt/case/submarine,Ref.[2e]

C [20] 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,
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 = balloon)
C        4.  GENOPT.HLP file    (type HELPG)
C==============================================================

C===============================================================
C   TABLE 2   GLOSSARY OF VARIABLES USED IN "balloon"
C==========================================================
C  ARRAY  NUMBER OF         PROMPT
C    ?   (ROWS,COLS)  ROLE  NUMBER   NAME         DEFINITION OF VARIABLE
C                         (balloon.PRO)
C==========================================================
C    n   (    0,   0)    2      10   LENGTH   = length of the cylindrical shell
C    n   (    0,   0)    2      15   RADIUS   = inner radius of the cylindrical b
C    n   (    0,   0)    2      20   NMODUL   = number of modules over 90 degrees
C    n   (    0,   0)    2      30   IEMOD1   = material number in EMOD1(IEMOD1)
C    y   (   10,   0)    2      35   EMOD1    = elastic modulus, meridional direc
C    y   (   10,   0)    2      40   EMOD2    = elastic modulus, circumferential
```

```
C    y   (  10,   0)   2      45   G12      = in-plane shear modulus
C    y   (  10,   0)   2      50   G13      = out-of-plane (s,z) shear modulus
C    y   (  10,   0)   2      55   G23      = out-of-plane (y,z) shear modulus
C    y   (  10,   0)   2      60   NU       = Poisson ratio
C    y   (  10,   0)   2      65   ALPHA1   = meridional coef. thermal expansio
C    y   (  10,   0)   2      70   ALPHA2   = circumf.coef.thermal expansion
C    y   (  10,   0)   2      75   TEMPER   = delta-T from fabrication temperat
C    y   (  10,   0)   2      80   DENSTY   = weight density of material
C    n   (   0,   0)   1      90   HEIGHT   = height from inner to outer membra
C    n   (   0,   0)   1      95   RINNER   = radius of curvature of inner memb
C    n   (   0,   0)   1     100   ROUTER   = radius of curvature of outer memb
C    n   (   0,   0)   1     105   TINNER   = thickness of the inner curved mem
C    n   (   0,   0)   1     110   TOUTER   = thickness of the outer curved mem
C    n   (   0,   0)   1     115   TFINNR   = thickness of inner truss-core seg
C    n   (   0,   0)   1     120   TFOUTR   = thickness of the outer truss segm
C    n   (   0,   0)   1     125   TFWEBS   = thickness of each truss-core web
C    n   (   0,   0)   2     135   NCASES   = Number of load cases (number of e
C    y   (  20,   0)   3     140   PINNER   = pressure inside the inner membran
C    y   (  20,   0)   3     145   PMIDDL   = pressure between inner and outer
C    y   (  20,   0)   3     150   POUTER   = pressure outside the outer membra
C    y   (  20,   0)   4     160   GENBUK   = general buckling load factor
C    y   (  20,   0)   5     165   GENBUKA  = allowable for general buckling lo
C    y   (  20,   0)   6     170   GENBUKF  = general buckling factor of safety
C    n   (   0,   0)   2     175   JSTRM1   = stress component number in STRM1(
C    y   (  20,   5)   4     180   STRM1    = stress component in material 1
C    y   (  20,   5)   5     185   STRM1A   = allowable stress in material 1
C    y   (  20,   5)   6     190   STRM1F   = factor of safety for stress in ma
C    y   (  20,   5)   4     195   STRM2    = stress component in material 2
C    y   (  20,   5)   5     200   STRM2A   = allowable for stress in material
C    y   (  20,   5)   6     205   STRM2F   = factor of safety for stress in ma
C    y   (  20,   5)   4     210   STRM3    = stress component in material 3
C    y   (  20,   5)   5     215   STRM3A   = allowable for stress in material
C    y   (  20,   5)   6     220   STRM3F   = factor of safety for stress in ma
C    n   (   0,   0)   7     230   WEIGHT   = weight/length of the balloon
C
C=DECK      BEHX1
      SUBROUTINE BEHX1
     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/FV03/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)
      REAL ALPHA2,TEMPER,DENSTY
      COMMON/FV21/PINNER(20)
      REAL PINNER
      COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )
      REAL STRM3,STRM3A,STRM3F
      COMMON/IV01/NMODUL
      INTEGER NMODUL
      COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT
      REAL TFINNR,TFOUTR,TFWEBS,WEIGHT
      COMMON/FV22/PMIDDL(20),POUTER(20)
      REAL PMIDDL,POUTER
C
```

```
C  INSERT SUBROUTINE STATEMENTS HERE.
C

      COMMON/ITRYX/ITRY
      DOUBLE PRECISION FTOTX
      COMMON/FPREBX/FMAXST(200),FTOTX(20000)
      COMMON/IFPREB/IFTOTS
      COMMON/SEGS/NSEGB4,M2B4,I5B4(295),I2B4,I2GB4
      COMMON/IFRHX/IFBB4,RHFIX(198),
     1             KKKK,MNUMB,ISWTCH,KNTB4,IFTOT,INDSIG,IFIXB4
      COMMON/FLNFLO/FLINNR,FLOUTR
      COMMON/WRDCLX/WRDCOL
      CHARACTER*45 WRDCOL
      COMMON/ITERS/ITER
      COMMON/ITERS2/ITRSTP(200)
      COMMON/MEMSTR/STRS1F(1,5),STRS2F(1,5),STRS1V(1,5),STRS2V(1,5)
      DIMENSION THK(5)
      COMMON/ERROR/ERR
      COMMON/N1N2FX/N1FIX(100,100),N2FIX(100,100)
      COMMON/N1N2VR/N1VAR(100,100),N2VAR(100,100)
      REAL N1FIX,N2FIX,N1VAR,N2VAR
      COMMON/N2DIFX/N2DIFF(6)
      REAL N2DIFF
      COMMON/FINNER/C44FIN,DELTAT,DELT,NODSEG,MSEGS
      COMMON/NUMSEG/NSEGS
      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,CASA2
C
      PI = 3.1415927
C
      IF (NMODUL.GT.36) THEN
         I=INDEX(CASE,' ')
         NLET = I - 1
         IF (I.EQ.0) NLET = 28
         WRITE(IFILE,'(/,A,/,A,/,A,A,A,/,A)')
```

```
     1' ************* RUN ABORT *****************',
     1' Too many modules. NMODUL must be less than 36',
     1' Reduce NMODUL in the file, ',CASE(1:NLET),'.BEG.',
     1' *****************************************'
        CALL ERREX
      ENDIF
C
      IF (IMODX.EQ.0) ERR = 0.
      IF (IMODX.EQ.1) ERR = 0.01
C
      RAVE = RADIUS/PI
      RBIGG = RAVE
C
C   Obtain nonlinear equilibrium for Load Set B by itself.
C   Use 10 load steps to assure convergence.
C
      INDIC = 0
      WRDCOL = '                                                      '
      IFTOTS = 0
      ITRY = 1
      NSTEPS = 10
      CALL MOVER(0.,0,FTOTX,1,40000)
      CALL BOSDEC(0,24,ILOADX,INDIC)
C
      IF (0.5*FLINNR.GT.RINNER) THEN
         WRITE(IFILE,'(/,A)') ' ************* ABORT **********'
         WRITE(IFILE,'(A)') ' 0.5 x FLINNR is greater than RINNER'
         WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')
     1 ' 0.5 x FLINNR =',0.5*FLINNR,'; RINNER =',RINNER,'; IMODX=',IMODX
         WRITE(IFILE,'(A)') ' Put a higher lower bound on RINNER.'
         WRITE(IFILE,'(A)') ' The run is now aborting.'
         WRITE(IFILE,'(A)') ' ********************************'
         CALL ERREX
      ENDIF
C23456789012345678901234567890123456789012345678901234567890123456789012
C
      IF (0.5*FLOUTR.GT.ROUTER) THEN
         WRITE(IFILE,'(/,A)') ' ************* ABORT **********'
         WRITE(IFILE,'(A)') ' 0.5 x FLOUTR is greater than ROUTER'
         WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')
     1 ' 0.5 x FLOUTR =',0.5*FLOUTR,'; ROUTER =',ROUTER,'; IMODX=',IMODX
         WRITE(IFILE,'(A)') ' Put a higher lower bound on ROUTER.'
         WRITE(IFILE,'(A)') ' The run is now aborting.'
         WRITE(IFILE,'(A)') ' ********************************'
         CALL ERREX
      ENDIF
C
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       pre-buckling state of the balloon.
C         I=INDEX(CASE,' ')
C         IF(I.NE.0) THEN
C            CASA=CASE(:I-1)//'.BEHX2'
C         ELSE
C            CASA=CASE//'.BEHX2'
C         ENDIF
C         OPEN(UNIT=61,FILE=CASA2,STATUS='UNKNOWN')
C         CALL BOSDEC(1,61,ILOADX,INDIC)
C         CLOSE(UNIT=61)
C         WRITE(IFILE,'(/,/,A,A,/,A)')
C     1 ' BIGBOSOR4 input file for:',
C     1 ' pre-buckling state of the balloon',
C     1   CASA2
C      ENDIF
C
      CALL B4READ
      CALL B4MAIN
C
      ILETW = INDEX(WRDCOL,'SHELL COLLAPSES AXISYMMETRICALLY')
      IF (ILETW.NE.0) THEN
         WRITE(IFILE,'(/,A)') ' ************* ABORT **********'
         WRITE(IFILE,'(A,A,/,1P,5E14.6)')
     1 ' Decision variable candidates,',
     1 ' HEIGHT,RINNER,ROUTER,TINNER,TOUTER=',
     1   HEIGHT,RINNER,ROUTER,TINNER,TOUTER
         WRITE(IFILE,'(A,1P,3E12.4)')
     1 ' TFINNR,TFOUTR,TFWEBS=',TFINNR,TFOUTR,TFWEBS
         WRITE(IFILE,'(A)') ' SHELL COLLAPSES AXISYMMETRICALLY'
         WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')
     1 ' 0.5 x FLINNR =',0.5*FLINNR,'; RINNER =',RINNER,'; IMODX=',IMODX
         WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')
     1 ' 0.5 x FLOUTR =',0.5*FLOUTR,'; ROUTER =',ROUTER,'; IMODX=',IMODX
         WRITE(IFILE,'(A)') ' Put a higher lower bound on RINNER or'
         WRITE(IFILE,'(A)') ' put a higher lower bound on ROUTER.'
         WRITE(IFILE,'(A)') ' The run is now aborting.'
         WRITE(IFILE,'(A)') ' ********************************'
         CALL ERREX
      ENDIF
C
      ILETW = INDEX(WRDCOL,'INITIAL LOADS TOO HIGH FOR THIS STRUCT')
      IF (ILETW.NE.0) THEN
         WRITE(IFILE,'(A,A,/,1P,5E14.6)')
     1 ' Decision variable candidates,',
     1 ' HEIGHT,RINNER,ROUTER,TINNER,TOUTER=',
     1   HEIGHT,RINNER,ROUTER,TINNER,TOUTER
```

```
         WRITE(IFILE,'(A,1P,3E12.4)')
     1 ' TFINNR,TFOUTR,TFWEBS=',TFINNR,TFOUTR,TFWEBS
         WRITE(IFILE,'(/,A)')
     1              ' ****** CHANGE FROM 10 TO 1 LOAD STEPS ******'
         WRITE(IFILE,'(A)') ' INITIAL LOADS TOO HIGH FOR THIS STRUCT'
         WRITE(IFILE,'(A,I2)') ' Changing from 10 to 1 steps: IMODX=',
     1    IMODX
         WRITE(IFILE,'(A)') ' ********************************'
         CALL GASP(DUM1,DUM2,-2,DUM3)
         WRDCOL = '                                              '
         INDIC = 0
         IFTOTS = 0
         ITRY = 2
         NSTEPS = 1
         CALL MOVER(0.,0,FTOTX,1,40000)
         CALL BOSDEC(0,24,ILOADX,INDIC)
         CALL B4READ
         CALL B4MAIN
         ITRY = 1
      ENDIF
C
      ILETW = INDEX(WRDCOL,'SHELL COLLAPSES AXISYMMETRICALLY')
      IF (ILETW.NE.0) THEN
         WRITE(IFILE,'(A,A,/,1P,5E14.6)')
     1 ' Decision variable candidates,',
     1 ' HEIGHT,RINNER,ROUTER,TINNER,TOUTER=',
     1   HEIGHT,RINNER,ROUTER,TINNER,TOUTER
         WRITE(IFILE,'(A,1P,3E12.4)')
     1 ' TFINNR,TFOUTR,TFWEBS=',TFINNR,TFOUTR,TFWEBS
         WRITE(IFILE,'(/,A)') ' ************* ABORT **********'
         WRITE(IFILE,'(A)') ' SHELL COLLAPSES AXISYMMETRICALLY'
         WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')
     1 ' 0.5 x FLINNR =',0.5*FLINNR,'; RINNER =',RINNER,'; IMODX=',IMODX
         WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')
     1 ' 0.5 x FLOUTR =',0.5*FLOUTR,'; ROUTER =',ROUTER,'; IMODX=',IMODX
         WRITE(IFILE,'(A)') ' Put a higher lower bound on RINNER or'
         WRITE(IFILE,'(A)') ' put a higher lower bound on ROUTER.'
         WRITE(IFILE,'(A)') ' The run is now aborting.'
         WRITE(IFILE,'(A)') ' ********************************'
         CALL ERREX
      ENDIF
C
      ILETW = INDEX(WRDCOL,'INITIAL LOADS TOO HIGH FOR THIS STRUCT')
      IF (ILETW.NE.0) THEN
         WRITE(IFILE,'(A,A,/,1P,5E14.6)')
     1 ' Decision variable candidates,',
     1 ' HEIGHT,RINNER,ROUTER,TINNER,TOUTER=',
     1   HEIGHT,RINNER,ROUTER,TINNER,TOUTER
```

```
         WRITE(IFILE,'(A,1P,3E12.4)')
     1 ' TFINNR,TFOUTR,TFWEBS=',TFINNR,TFOUTR,TFWEBS
         WRITE(IFILE,'(/,A)')
     1               ' ****** CHANGE FROM 1 TO 50 LOAD STEPS ******'
         WRITE(IFILE,'(A)') ' INITIAL LOADS TOO HIGH FOR THIS STRUCT'
         WRITE(IFILE,'(A,I2)') ' Changing from 1 to 50 steps: IMODX=',
     1    IMODX
         WRITE(IFILE,'(A)') ' ********************************'
         CALL GASP(DUM1,DUM2,-2,DUM3)
         WRDCOL = '                                              '
         INDIC = 0
         IFTOTS = 0
         ITRY = 3
         NSTEPS = 50
         CALL MOVER(0.,0,FTOTX,1,40000)
         CALL BOSDEC(0,24,ILOADX,INDIC)
         CALL B4READ
         CALL B4MAIN
         ITRY = 1
      ENDIF
C
      ILETW = INDEX(WRDCOL,'SHELL COLLAPSES AXISYMMETRICALLY')
      IF (ILETW.NE.0) THEN
         WRITE(IFILE,'(A,A,/,1P,5E14.6)')
     1 ' Decision variable candidates,',
     1 ' HEIGHT,RINNER,ROUTER,TINNER,TOUTER=',
     1   HEIGHT,RINNER,ROUTER,TINNER,TOUTER
         WRITE(IFILE,'(A,1P,3E12.4)')
     1 ' TFINNR,TFOUTR,TFWEBS=',TFINNR,TFOUTR,TFWEBS
         WRITE(IFILE,'(/,A)') ' ************* ABORT **********'
         WRITE(IFILE,'(A)') ' SHELL COLLAPSES AXISYMMETRICALLY'
         WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')
     1 ' 0.5 x FLINNR =',0.5*FLINNR,'; RINNER =',RINNER,'; IMODX=',IMODX
         WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')
     1 ' 0.5 x FLOUTR =',0.5*FLOUTR,'; ROUTER =',ROUTER,'; IMODX=',IMODX
         WRITE(IFILE,'(A)') ' Put a higher lower bound on RINNER or'
         WRITE(IFILE,'(A)') ' put a higher lower bound on ROUTER.'
         WRITE(IFILE,'(A)') ' The run is now aborting.'
         WRITE(IFILE,'(A)') ' ********************************'
         CALL ERREX
      ENDIF
C
      ILETW = INDEX(WRDCOL,'INITIAL LOADS TOO HIGH FOR THIS STRUCT')
      IF (ILETW.NE.0) THEN
         WRITE(IFILE,'(A,A,/,1P,5E14.6)')
     1 ' Decision variable candidates,',
     1 ' HEIGHT,RINNER,ROUTER,TINNER,TOUTER=',
     1   HEIGHT,RINNER,ROUTER,TINNER,TOUTER
```

```
         WRITE(IFILE,'(A,1P,3E12.4)')
     1 ' TFINNR,TFOUTR,TFWEBS=',TFINNR,TFOUTR,TFWEBS
         WRITE(IFILE,'(/,A)') ' ************* ABORT **********'
         WRITE(IFILE,'(A)') ' INITIAL LOADS TOO HIGH FOR THIS STRUCT'
         WRITE(IFILE,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1 ' This is an unrecoverable error because we have already',
     1 ' tried and failed to obtain nonlinear pre-buckling convergence',
     1 ' by changing from a nonlinear solution with 10 load steps to',
     1 ' a nonlinear solution with 1 load step and then changing from',
     1 ' 1 load step to 50 load steps:three tries. That strategy just',
     1 ' failed. You may well have performed enough design iterations',
     1 ' to have a good optimum design now. Look near the end of the',
     1 ' *.OPP file at the "FEASIBLE" and "ALMOST FEASIBLE" designs.'
         WRITE(IFILE,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1 ' If you are not satisfied that you have performed enough',
     1 ' design iterations, then look at the thicknesses of the',
     1 ' various segments. If any thicknesses seem too small, then',
     1 ' increase them and also increase the corresponding lower',
     1 ' bounds of them. Another thing you can try that has worked',
     1 ' for Bushnell is to look near the end of the *.OPM file for',
     1 ' the last successfully obtained design. Use the GENOPT',
     1 ' processor, CHANGE, to reset the values of the decision',
     1 ' variables to those of the last successfully obtained design',
     1 ' and then launch a new execution of SUPEROPT, probably',
     1 ' leaving the lower bounds unchanged, or perhaps also changing',
     1 ' them if you wish (before launching SUPEROPT, of course).'
         WRITE(IFILE,'(A,I2)') ' The run is now aborting: IMODX=',IMODX
         WRITE(IFILE,'(A)') ' ********************************'
         CALL ERREX
      ENDIF
C
      IF (IMODX.EQ.0) THEN
         WRITE(IFILE,'(/,A,/,A,/,A,1P,E12.4,A,1P,E12.4,A,1P,E12.4,/,A)')
     1 ' Newton iterations required to solve the nonlinear',
     1 ' axisymmetric pre-buckling equilibrium state for the',
     1 ' "fixed" loads, PINNER=',PINNER(ILOADX),
     1 ', PMIDDL=',PMIDDL(ILOADX),', DELTAT=',DELTAT,
     1 ' LOAD STEP   Newton iterations  Maximum displacement'
         DO 3 I = 1,NSTEPS
          WRITE(IFILE,'(I6,I15,1P,E25.6)') I,ITRSTP(I),FMAXST(I)
    3    CONTINUE
      ENDIF
C
      WRITE(IFILE,'(A,A,/,A,/,A,I2)') ' WRDCOL=',WRDCOL,
     1' IMODX=0 for current design,',
     1' IMODX=1 for perturbed design: IMODX=',IMODX
C
C23456789012345678901234567890123456789012345678901234567890123456789012
```

```
C     WRITE(IFILE,'(/,A,2I10)')' IFTOT, M2B4 =', IFTOT,M2B4
      IFTOTS = IFTOT
      M22B4 = 2*M2B4
      CALL GASP(FTOTX,M22B4,3,IFTOT)
C     WRITE(IFILE,'(/,A,/,(1P,10E10.2))')
C    1' Nonlinear solution for Load Set B by itself, FTOTX=',
C    1 (FTOTX(I),I=1,M2B4)
      CALL GASP(DUM1,DUM2,-2,DUM3)
C
      INDIC = 1
      NOB = 1
      NMAXB = 1
      WRDCOL = '                                              '
      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      general 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 ' general buckling load',
     1   CASA
      ENDIF
C
      CALL B4READ
      IF (IMODX.EQ.0) THEN
         NOBX = NOB
         NMINBX = NOB
         NMAXBX = NMAXB
         INCRBX = 1
      ELSE
         NOBX = NWAV1
         NMINBX = NWAV1
         NMAXBX = NWAV1
         INCRBX = 1
      ENDIF
      REWIND IFILE9
      CALL STOCM1(IFILE9)
```

```
      CALL STOCM2(IFILE9)
      CALL B4MAIN
      CALL GASP(DUM1,DUM2,-2,DUM3)
      IF (IMODX.EQ.0) THEN
         EIG2 = EIGCRT
         NWAV1= NWVCRT
      ENDIF
C
      ILETW = INDEX(WRDCOL,'SHELL COLLAPSES AXISYMMETRICALLY')
      IF (ILETW.NE.0) THEN
         WRITE(IFILE,'(/,A)') ' ************* ABORT **********'
         WRITE(IFILE,'(A)') ' SHELL COLLAPSES AXISYMMETRICALLY'
         WRITE(IFILE,'(A,I2)') ' The run is now aborting: IMODX=',IMODX
         WRITE(IFILE,'(A)') ' ********************************'
         CALL ERREX
      ENDIF
C
      ILETW = INDEX(WRDCOL,'INITIAL LOADS TOO HIGH FOR THIS STRUCT')
      IF (ILETW.NE.0) THEN
         WRITE(IFILE,'(/,A)') ' ************* ABORT **********'
         WRITE(IFILE,'(A)') ' INITIAL LOADS TOO HIGH FOR THIS STRUCT'
         WRITE(IFILE,'(A,I2)') ' The run is now aborting: IMODX=',IMODX
         WRITE(IFILE,'(A)') ' ********************************'
         CALL ERREX
      ENDIF
C
C23456789012345678901234567890123456789012345678901234567890123456789012
C
C   Compute the membrane stresses
      THK(1) = TFOUTR
      THK(2) = TOUTER
      THK(3) = TFINNR
      THK(4) = TINNER
      THK(5) = TFWEBS
C
      DO 5 J = 1,5
       STRS1F(1,J) = N1FIX(1,J)/THK(J)
       STRS2F(1,J) = N2FIX(1,J)/THK(J)
       STRS1V(1,J) = N1VAR(1,J)/THK(J)
       STRS2V(1,J) = N2VAR(1,J)/THK(J)
    5 CONTINUE
C
      IF (IMODX.EQ.0) THEN
C23456789012345678901234567890123456789012345678901234567890123456789012
        WRITE(IFILE,'(/,A,/,A,/,A,/,A,/,A,1P,E12.4,A,/,A,/,A,1P,E12.4)')
     1' Changes in temperature required to create 2 total axial loads:',
     1'  ',
     1'  1. Change in temperature required to create the axial thermal',
```

```
     1'     strain that generates the axial tension due to closing the',
     1'     two ends of the pressurized volume (PMIDDL=',
     1  PMIDDL(ILOADX),')',
     1'     between the inner and outer walls of the balloon in',
     1'     Load Step No. 1:                                DELTAT=',
     1  DELTAT
        WRITE(IFILE,'(/,A,/,A,/,A,1P,1E12.4,A,1P,E12.4,/,/)')
     1'  2. Change in temperature required to simulate the Poisson',
     1'     axial expansion caused by the application of the outer',
     1'     pressure, POUTER =',
     1  POUTER(ILOADX),' in Load Step No. 2:  DELT=',
     1  DELT
        WRITE(IFILE,'(/,A)')
     1 ' GENERAL BUCKLING LOAD FACTORS AND MODES (BEHX1)'
        DO 10 I = 1,IWAVEB
         WRITE(IFILE,'(A,1P,E12.4,A,I6,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 axial half-waves, NWVCRT=',NWVCRT
C
        DO 30 J = 1,6
        DO 20 I = 1,1
          N2DIFF(J) = N2VAR(I,J) - N2FIX(I,J)
   20   CONTINUE
   30   CONTINUE
        WRITE(IFILE,'(/,A,/,A,/,A,/,A,1P,6E12.4,/,A,/,A,/,A,/,A,/,A)')
     1' Differences in the resultants along the axis of the prismatic',
     1' balloon for each segment, J, of the first module:',
     1' [N2VAR(J) for the total load] - [N2FIX(J) for the fixed load]=',
     1' N2DIFF(J),J=1,6)=',(N2DIFF(J),J=1,6),
     1' N2VAR(J) (total load) are the resultants from Load Step No. 2.',
     1' N2FIX(J) (fixed load) are the resultants from Load Step No. 1.',
     1' NOTE: The stresses used as behavioral constraints are',
     1'       computed from N2VAR(J)/thickness(J). These stresses are',
     1'       lower than those computed from N2FIX(J)/thickness(J).'
C234567890123456789012345678901234567890123456789012345678901234567890I2
C
        IF (NPRINX.GE.2) THEN
         WRITE(IFILE,'(/,A,/,A,A,/,A,A)')
     1 ' PREBUCKLING STRESS RESULTANTS IN THE FIRST MODULE',
     1 '          "fixed" from Load Step No. 1',
     1 ' total from Load Step No. 2',
     1 ' Seg.J  Node I  N1FIX(I,J)   N2FIX(I,J)',
     1 '   N1VAR(I,J)   N2VAR(I,J)'
         DO 60 J = 1,6
```

```
         DO 50 I = 1,1
           WRITE(IFILE,'(I5,I7,1P,4E13.4)')
     1     J,I,N1FIX(I,J),N2FIX(I,J),N1VAR(I,J),N2VAR(I,J)
   50    CONTINUE
   60    CONTINUE
C
C234567890123456789012345678901234567890123456789012345678901234567890
         WRITE(IFILE,'(/,A,A,/,A,A,/,A,A,/,A,A)')
     1 ' PREBUCKLING MEMBRANE STRESSES IN THE FIRST MODULE COMPUTED',
     1 ' FROM',
     1 ' N1FIX/thickness, N2FIX/thickness, N1VAR/thickness,',
     1 ' N2VAR/thickness:',
     1 '          "fixed" from Load Step No. 1',
     1 ' total from Load Step No. 2',
     1 ' Seg.J  Node I STRS1F(I,J)  STRS2F(I,J)',
     1 '  STRS1V(I,J)  STRS2V(I,J)'
         DO 90 J = 1,5
         DO 80 I = 1,1
           WRITE(IFILE,'(I5,I7,1P,4E13.4)')
     1     J,I,STRS1F(I,J),STRS2F(I,J),STRS1V(I,J),STRS2V(I,J)
   80    CONTINUE
   90    CONTINUE
C
         WRITE(IFILE,'(/,A)')
     1 ' Behavior number, General buckling load factor:'
        ENDIF
C
        WRITE(IFILE,
     1  '(/,A,/,A,/,A,I3,/,A,1P,E12.4,/,A,/,A,/,A,I3,/,A,1P,E12.4,/)')
     1' Newton iterations required to solve the nonlinear',
     1' axisymmetric pre-buckling equilibrium state for the',
     1' "fixed" loads (PINNER, PMIDDL, DELTAT):       ITER=',ITRSTP(1),
     1' Maximum displacement,                         FMAX=',FMAXST(1),
     1' Newton iterations required to solve the nonlinear',
     1' axisymmetric pre-buckling equilibrium state for the',
     1' total loads (PINNER, PMIDDL, DELTAT, POUTER): ITER=',ITRSTP(2),
     1' Maximum displacement,                         FMAX=',FMAXST(2)
C
      ENDIF
C
      IF (EIGCRT.LE.0.0) THEN
         WRITE(IFILE,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1 ' ******************** RUN ABORT **********************',
     1 ' Divergence or failure of convergence of nonlinear',
     1 ' pre-buckling solution either at Load Step No. 1 (fixed',
     1 ' loads: PINNER, PMIDDL, DELTAT) or at Load Step No. 2',
     1 ' (total loads: PINNER, PMIDDL, DELTAT, POUTER)',
     1 ' Probably you should increase either RINNER or ROUTER or',
```

```
     1 ' both RINNER and ROUTER.',
     1 ' ******************* RUN ABORT **********************'
          CALL ERREX
      ENDIF
C
      GENBUK(ILOADX) = EIGCRT
C
      RETURN
      END
C
C
C
C=DECK      BEHX2
      SUBROUTINE BEHX2
     1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)
C
C   PURPOSE: OBTAIN stress component in material 1
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     stress component number
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 STRM1
C     JCOL   = stress component number
```

```
C     PHRASE = stress component in material 1
C
C   OUTPUT:
C
C     STRM1(ILOADX,JCOL)
C
      CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV03/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)
      REAL ALPHA2,TEMPER,DENSTY
      COMMON/FV21/PINNER(20)
      REAL PINNER
      COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )
      REAL STRM3,STRM3A,STRM3F
      COMMON/IV01/NMODUL
      INTEGER NMODUL
      COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT
      REAL TFINNR,TFOUTR,TFWEBS,WEIGHT
      COMMON/FV22/PMIDDL(20),POUTER(20)
      REAL PMIDDL,POUTER
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
      COMMON/MEMSTR/STRS1F(1,5),STRS2F(1,5),STRS1V(1,5),STRS2V(1,5)
      COMMON/FINNER/C44FIN,DELTAT,DELT,NODSEG,MSEGS
      COMMON/ERROR/ERR
      COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
      COMMON/LAMSTR/STRRAT(99)
      COMMON/WRDSTR/STRWRD(99)
      CHARACTER*80 STRWRD
      DIMENSION ILET1(15,6),ILET2(15,3)
      common/caseblock/CASE
      CHARACTER*28 CASE
      CHARACTER*35 CASA
      COMMON/STRCMP/STRC1(20,5),STRC2(20,5),STRC3(20,5)
```

```
C
C     IF (JCOL.GT.1) GO TO 500
C
C  NOTE IMPORTANT CHANGE:
C
C  October 7, 2010: Use the MEMBRANE stresses computed in
C  SUBROUTINE BEHX1 because the
C  meridional curvature change, KAPPA1, from BIGBOSOR4 is
C  sometimes much too large in the immediate neighborhoods
C  of the ends of the shell segments, generating maximum
C  stress components that are much too high in this particular
C  case that involves a balloon-like (membrane) structure,
C  This is especially true for the outer and inner curved
C  membranes, that is, segments 2 (outer) and 4 (inner)
C  of each module of the multi-module model. For example,
C  here is some BIGBOSOR4 output for Segment 4 (inner
C  curved membrane) for Load Step No. 1 (applied loads
C  are PINNER, PMIDDL, and DELTAT):
C
C      AXISYMMETRIC PRESTRESS DISTRIBUTION FOR SEGMENT 4
C POINT   EPSILON 1  EPSILON 2   KAPPA 1      KAPPA 2
C           MERID.     CIRCUMF.    MERID.      CIRCUMF.
C           STRAIN     STRAIN     CHANGES IN CURVATURE
C   1    1.070E-01   7.019E-15   5.784E-01   1.795E-08
C   2    1.062E-01  -1.695E-09  -1.015E+01  -4.307E-10
C   3    1.059E-01   1.154E-08   2.981E+00  -8.949E-09
C   4    1.074E-01   1.021E-08  -1.251E+00  -2.625E-09
C   5    1.077E-01   1.483E-08   6.715E-01  -5.319E-09
C   6    1.085E-01   1.647E-08  -2.629E-01  -3.690E-09
C   7    1.090E-01   1.910E-08   1.910E-01  -4.184E-09
C   8    1.096E-01   2.095E-08  -3.076E-02  -3.533E-09
C   9    1.100E-01   2.283E-08   7.706E-02  -3.398E-09
C  10    1.104E-01   2.441E-08   2.394E-02  -2.950E-09
C
C  Note that the meridional change in curvature, KAPPA 1,
C  is very large at nodal points 2, 3, 4, especially at
C  nodal point 2. This very local edge effect gives rise to
C  artificially high local bending meridional strain, which
C  probably does not exist in a balloon (membrane
C  pressure-stabilized "shell" structure). The extreme
C  fiber meridional strain from BIGBOSOR4 is given by:
C  EPS1 = EPSILON1 + THICK*KAPPA1/2.
C  in which EPSILON1 is the reference (middle) surface
C  meridional strain and THICK is the thickness of the
C  shell segment.
C
C  Because of this spurious and extremely high meridional
C  bending strain predicted by BIGBOSOR4 (which has difficulty
```

```
C  predicting accurate bending stresses in membrane-like structures
C  but which works well for shell structures with "finite"
C  bending stiffness), the previous FORTRAN statement:
C     IF (JCOL.GT.1) GO TO 500
C  has been commented out and replaced by the following
C  statement, "GO TO 500". Because of this important change
C  the file, *.BEHX2, is no longer created and you can
C  therefore no longer obtain plots of the pre-buckled states
C  at Load Steps 1 and 2 unless you remove the "C" in column
C  1 of the statement, "IF (JCOL.GT.1) GO TO 500", and insert
C  a "C" in column 1 of the following statement, "GO TO 500".
C  and then re-compile via the GENOPT command, "genprograms".
C
      GO TO 500
C
      IF (IMODX.EQ.0) ERR = 0.
      IF (IMODX.EQ.1) ERR = 0.01
C
      INDIC = 0
      RAVE = RADIUS/PI
      RBIGG = RAVE
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      STRM1(ILOADX,JCOL), JCOL=1,5: stress components in material 1
C
C      NOTE: Also computed in SUBROUTINE BEHX2 are the following:
C      STRM2(ILOADX,JCOL), JCOL=1,5: stress components in material 2
C      STRM3(ILOADX,JCOL), JCOL=1,5: stress components in material 3
C
C      STRM2(ILOADX,JCOL) is available in SUBROUTINE BEHX3 because
C                         it is in a labelled common block.
C      STRM3(ILOADX,JCOL) is available in SUBROUTINE BEHX4 because
C                         it is in a labelled common block.
C
         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 ' stress components in materials 1,2,3',
     1   CASA
      ENDIF
C
      CALL B4READ
C
      CALL B4MAIN
      CALL GASP(DUM1,DUM2,-2,DUM3)
C
C With INDIC = 0,
C BIGBOSOR4 generates stress constraints for laminated composite
C material in the following form (in this case all shell segments
C have only one layer, and the balloon is in tension everywhere.:
C
C****** (ALLOWABLE STRESS)/(ACTUAL STRESS) ********
C 1  3.1045E+00 fiber tension : matl=1 ,  A , seg=50, node=32, layer=1
,z=0.01   ;FS=1.
C 2  1.7557E+00 transv tension: matl=1 ,  A , seg=92, node=33, layer=1
,z=0.01   ;FS=1.
C 3  5.1415E+02 fiber tension : matl=2 ,  A , seg=1 , node=33, layer=1
,z=1.     ;FS=1.
C 4  1.7557E+02 transv tension: matl=2 ,  A , seg=91, node=33, layer=1
,z=1.     ;FS=1.
C 5  4.9549E+00 fiber tension : matl=3 ,  A , seg=11, node=1 , layer=1
,z=-0.01  ;FS=1.
C 6  1.7557E+00 transv tension: matl=3 ,  A , seg=90, node=33, layer=1
,z=0.01   ;FS=1.
C*****************************************************
C
C or, for an isotropic material:
C
C****** (ALLOWABLE STRESS)/(ACTUAL STRESS) ********
C 1  1.5325E+00 effect. stress: matl=1 ,  A , seg=52, node=32, layer=1
,z=0.01   ;FS=1.
C 2  1.0960E+00 effect. stress: matl=2 ,  A , seg=81, node=32, layer=1
,z=0.01   ;FS=1.
C 3  1.9372E+00 effect. stress: matl=3 ,  A , seg=84, node=33, layer=1
,z=0.01   ;FS=1.
C*****************************************************
C
      ICONST = 5*3
C
      DO 10 J = 1,5
         STRC1(ILOADX,J) = 0.
         STRC2(ILOADX,J) = 0.
         STRC3(ILOADX,J) = 0.
   10 CONTINUE
```

```
C
      DO 50 I = 1,ICONST
         IF (STRRAT(I).EQ.0.) THEN
            ICONS2 = ICONST - 1
            GO TO 60
         ENDIF
         IF (ITYPEX.EQ.2.AND.IMODX.EQ.0) THEN
           IF (I.EQ.1) WRITE(IFILE,'(/,A,A)')
     1    ' Maximum stress components in the entire structure',
     1    ' at the last load step (from BIGBOSOR4):'
           WRITE(IFILE,40) I,STRRAT(I),STRWRD(I)(1:64)
         ENDIF
   40 FORMAT(I3,1P,E12.4,1X,A)
   50 CONTINUE
   60 CONTINUE
C
      DO 100 I = 1,ICONS2
C
        ILET1(I,1)  = INDEX(STRWRD(I),'fiber tension')
        ILET1(I,2)  = INDEX(STRWRD(I),'fiber compres')
        ILET1(I,3)  = INDEX(STRWRD(I),'transv tension')
        ILET1(I,4)  = INDEX(STRWRD(I),'transv compres')
        ILET1(I,5)  = INDEX(STRWRD(I),'in-plane shear')
        ILET1(I,6)  = INDEX(STRWRD(I),'effect. stress')
C
        ILET2(I,1)  = INDEX(STRWRD(I),'matl=1')
        ILET2(I,2)  = INDEX(STRWRD(I),'matl=2')
        ILET2(I,3)  = INDEX(STRWRD(I),'matl=3')
  100 CONTINUE
C
      DO 200 I = 1,ICONS2
C23456789012345678901234567890123456789012345678901234567890123456789012
        IF (ILET1(I,1).NE.0) THEN
         IF (ILET2(I,1).NE.0) STRC1(ILOADX,1)=STRM1A(ILOADX,1)/STRRAT(I)
         IF (ILET2(I,2).NE.0) STRC2(ILOADX,1)=STRM2A(ILOADX,1)/STRRAT(I)
         IF (ILET2(I,3).NE.0) STRC3(ILOADX,1)=STRM3A(ILOADX,1)/STRRAT(I)
        ENDIF
C
        IF (ILET1(I,2).NE.0) THEN
         IF (ILET2(I,1).NE.0) STRC1(ILOADX,2)=STRM1A(ILOADX,2)/STRRAT(I)
         IF (ILET2(I,2).NE.0) STRC2(ILOADX,2)=STRM2A(ILOADX,2)/STRRAT(I)
         IF (ILET2(I,3).NE.0) STRC3(ILOADX,2)=STRM3A(ILOADX,2)/STRRAT(I)
        ENDIF
C
        IF (ILET1(I,3).NE.0) THEN
         IF (ILET2(I,1).NE.0) STRC1(ILOADX,3)=STRM1A(ILOADX,3)/STRRAT(I)
         IF (ILET2(I,2).NE.0) STRC2(ILOADX,3)=STRM2A(ILOADX,3)/STRRAT(I)
         IF (ILET2(I,3).NE.0) STRC3(ILOADX,3)=STRM3A(ILOADX,3)/STRRAT(I)
```

```
         ENDIF
C
         IF (ILET1(I,4).NE.0) THEN
          IF (ILET2(I,1).NE.0) STRC1(ILOADX,4)=STRM1A(ILOADX,4)/STRRAT(I)
          IF (ILET2(I,2).NE.0) STRC2(ILOADX,4)=STRM2A(ILOADX,4)/STRRAT(I)
          IF (ILET2(I,3).NE.0) STRC3(ILOADX,4)=STRM3A(ILOADX,4)/STRRAT(I)
         ENDIF
C
         IF (ILET1(I,5).NE.0) THEN
          IF (ILET2(I,1).NE.0) STRC1(ILOADX,5)=STRM1A(ILOADX,5)/STRRAT(I)
          IF (ILET2(I,2).NE.0) STRC2(ILOADX,5)=STRM2A(ILOADX,5)/STRRAT(I)
          IF (ILET2(I,3).NE.0) STRC3(ILOADX,5)=STRM3A(ILOADX,5)/STRRAT(I)
         ENDIF
C
         IF (ILET1(I,6).NE.0) THEN
          IF (ILET2(I,1).NE.0) STRC1(ILOADX,1)=STRM1A(ILOADX,1)/STRRAT(I)
          IF (ILET2(I,2).NE.0) STRC2(ILOADX,1)=STRM2A(ILOADX,1)/STRRAT(I)
          IF (ILET2(I,3).NE.0) STRC3(ILOADX,1)=STRM3A(ILOADX,1)/STRRAT(I)
         ENDIF
C
  200 CONTINUE
C
C23456789012345678901234567890123456789012345678901234567890123456789012
       IF (IMODX.EQ.0) THEN
          WRITE(IFILE,'(/,A,A,/,A,A,/,A)')
     1 ' FIVE STRESS COMPONENTS (including bending) FOR MATL i,',
     1 ' STRCi(ILOADX,J), J=1,5:',
     1 '    fiber tension  fiber compres ',
     1 ' transv tension transv compres in-plane shear',
     1 '  or effect.stress'
          WRITE(IFILE,'(A,/,1P,5E15.4)')
     1 ' Material 1 stress: STRC1(ILOADX,J),J=1,5)=',
     1   (STRC1(ILOADX,J),J=1,5)
          WRITE(IFILE,'(A,/,1P,5E15.4)')
     1 ' Material 2 stress: STRC2(ILOADX,J),J=1,5)=',
     1   (STRC2(ILOADX,J),J=1,5)
          WRITE(IFILE,'(A,/,1P,5E15.4,/)')
     1 ' Material 3 stress: STRC3(ILOADX,J),J=1,5)=',
     1   (STRC3(ILOADX,J),J=1,5)
C23456789012345678901234567890123456789012345678901234567890123456789012
         WRITE(IFILE,'(/,A,/,A,/,A,/,A,/,A,1P,E12.4,A,/,A,/,A,1P,E12.4)')
     1' Changes in temperature required to create 2 total axial loads:',
     1'  ',
     1'  1. Change in temperature required to create the axial thermal',
     1'     strain that generates the axial tension due to closing the',
     1'     two ends of the pressurized volume (PMIDDL=',
     1  PMIDDL(ILOADX),')',
     1'     between the inner and outer walls of the balloon in',
```

```
     1'     Load Step No. 1:                              DELTAT=',
     1  DELTAT
        WRITE(IFILE,'(/,A,/,A,/,A,1P,1E12.4,A,1P,E12.4,/,/)')
     1'  2. Change in temperature required to simulate the Poisson',
     1'     axial expansion caused by the application of the outer',
     1'     pressure, POUTER =',
     1  POUTER(ILOADX),' in Load Step No. 2:  DELT=',
     1  DELT
        WRITE(IFILE,'(A)')
     1' BEHAVIOR OVER J =  stress component number'
      ENDIF
C
  500 CONTINUE
C
C   NOTE: the quantities, STRS1V and STRS2V, are computed
C         in SUBROUTINE BEHX1.
C
      STRC1(ILOADX,1) = MAX(STRS1V(1,2),STRS1V(1,4))
      STRC1(ILOADX,2) = 0.
      STRC1(ILOADX,3) = MAX(STRS2V(1,2),STRS2V(1,4))
      STRC1(ILOADX,4) = 0.
      STRC1(ILOADX,5) = 0.
C
      STRM1(ILOADX,JCOL) = STRC1(ILOADX,JCOL)
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 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     stress component number
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   = stress component number
C     PHRASE = stress component in material 2
C
C   OUTPUT:
C
C     STRM2(ILOADX,JCOL)
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV03/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)
      REAL ALPHA2,TEMPER,DENSTY
      COMMON/FV21/PINNER(20)
      REAL PINNER
      COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )
      REAL STRM3,STRM3A,STRM3F
      COMMON/IV01/NMODUL
      INTEGER NMODUL
      COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT
```

```
      REAL TFINNR,TFOUTR,TFWEBS,WEIGHT
      COMMON/FV22/PMIDDL(20),POUTER(20)
      REAL PMIDDL,POUTER
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
      COMMON/MEMSTR/STRS1F(1,5),STRS2F(1,5),STRS1V(1,5),STRS2V(1,5)
      COMMON/STRCMP/STRC1(20,5),STRC2(20,5),STRC3(20,5)
C
C    Do not need any calculations added here because
C    STRC2(ILOADX,JCOL), JCOL = 1,2,3,4,5
C    are computed in SUBROUTINE BEHX2
C
C  NOTE IMPORTANT CHANGE:
C
C  October 7, 2010: Use the MEMBRANE stresses because the
C  meridional curvature change, KAPPA1, from BIGBOSOR4 is
C  sometimes much too large in the immediate neighborhoods
C  of the ends of the shell segments, generating maximum
C  stress components that are much too high in this particular
C  case that involves a balloon-like (membrane) structure,
C  This is especially true for the outer and inner curved
C  membranes, that is, segments 2 (outer) and 4 (inner)
C  of each module of the multi-module model. For example,
C  here is some BIGBOSOR4 output for Segment 4 (inner
C  curved membrane) for Load Step No. 1:
C
C      AXISYMMETRIC PRESTRESS DISTRIBUTION FOR SEGMENT 4
C POINT   EPSILON 1  EPSILON 2   KAPPA 1      KAPPA 2
C           MERID.     CIRCUMF.    MERID.      CIRCUMF.
C           STRAIN     STRAIN     CHANGES IN CURVATURE
C   1    1.070E-01   7.019E-15   5.784E-01   1.795E-08
C   2    1.062E-01  -1.695E-09  -1.015E+01  -4.307E-10
C   3    1.059E-01   1.154E-08   2.981E+00  -8.949E-09
C   4    1.074E-01   1.021E-08  -1.251E+00  -2.625E-09
C   5    1.077E-01   1.483E-08   6.715E-01  -5.319E-09
C   6    1.085E-01   1.647E-08  -2.629E-01  -3.690E-09
C   7    1.090E-01   1.910E-08   1.910E-01  -4.184E-09
C   8    1.096E-01   2.095E-08  -3.076E-02  -3.533E-09
C   9    1.100E-01   2.283E-08   7.706E-02  -3.398E-09
C  10    1.104E-01   2.441E-08   2.394E-02  -2.950E-09
C
C
C   NOTE: the quantities, STRS1V and STRS2V, are computed
C         in SUBROUTINE BEHX1.
C
      STRC2(ILOADX,1) = MAX(STRS1V(1,1),STRS1V(1,3))
```

```
      STRC2(ILOADX,2) = 0.
      STRC2(ILOADX,3) = MAX(STRS2V(1,1),STRS2V(1,3))
      STRC2(ILOADX,4) = 0.
      STRC2(ILOADX,5) = 0.
C
      STRM2(ILOADX,JCOL) = STRC2(ILOADX,JCOL)
C
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 component in material 3
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     STRM3(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     stress component number
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 STRM3
C      JCOL   = stress component number
C      PHRASE = stress component in material 3
C
C    OUTPUT:
C
C      STRM3(ILOADX,JCOL)
C
        CHARACTER*80 PHRASE
C   INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV03/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)
      REAL ALPHA2,TEMPER,DENSTY
      COMMON/FV21/PINNER(20)
      REAL PINNER
      COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )
      REAL STRM3,STRM3A,STRM3F
      COMMON/IV01/NMODUL
      INTEGER NMODUL
      COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT
      REAL TFINNR,TFOUTR,TFWEBS,WEIGHT
      COMMON/FV22/PMIDDL(20),POUTER(20)
      REAL PMIDDL,POUTER
C
C
C   INSERT SUBROUTINE STATEMENTS HERE.
C
      COMMON/MEMSTR/STRS1F(1,5),STRS2F(1,5),STRS1V(1,5),STRS2V(1,5)
      COMMON/STRCMP/STRC1(20,5),STRC2(20,5),STRC3(20,5)
C
C     Do not need any calculations added here because
C     STRC3(ILOADX,JCOL), JCOL = 1,2,3,4,5
C     are computed in SUBROUTINE BEHX2
C
C   NOTE IMPORTANT CHANGE:
C
C   October 7, 2010: Use the MEMBRANE stresses because the
```

```
C  meridional curvature change, KAPPA1, from BIGBOSOR4 is
C  sometimes much too large in the immediate neighborhoods
C  of the ends of the shell segments, generating maximum
C  stress components that are much too high in this particular
C  case that involves a balloon-like (membrane) structure,
C  This is especially true for the outer and inner curved
C  membranes, that is, segments 2 (outer) and 4 (inner)
C  of each module of the multi-module model. For example,
C  here is some BIGBOSOR4 output for Segment 4 (inner
C  curved membrane) for Load Step No. 1:
C
C      AXISYMMETRIC PRESTRESS DISTRIBUTION FOR SEGMENT 4
C POINT   EPSILON 1  EPSILON 2   KAPPA 1      KAPPA 2
C          MERID.     CIRCUMF.    MERID.      CIRCUMF.
C          STRAIN     STRAIN     CHANGES IN CURVATURE
C   1    1.070E-01   7.019E-15   5.784E-01   1.795E-08
C   2    1.062E-01  -1.695E-09  -1.015E+01  -4.307E-10
C   3    1.059E-01   1.154E-08   2.981E+00  -8.949E-09
C   4    1.074E-01   1.021E-08  -1.251E+00  -2.625E-09
C   5    1.077E-01   1.483E-08   6.715E-01  -5.319E-09
C   6    1.085E-01   1.647E-08  -2.629E-01  -3.690E-09
C   7    1.090E-01   1.910E-08   1.910E-01  -4.184E-09
C   8    1.096E-01   2.095E-08  -3.076E-02  -3.533E-09
C   9    1.100E-01   2.283E-08   7.706E-02  -3.398E-09
C  10    1.104E-01   2.441E-08   2.394E-02  -2.950E-09
C
C
C   NOTE: the quantities, STRS1V and STRS2V, are computed
C         in SUBROUTINE BEHX1.
C
      STRC3(ILOADX,1) = STRS1V(1,5)
      STRC3(ILOADX,2) = 0.
      STRC3(ILOADX,3) = STRS2V(1,5)
      STRC3(ILOADX,4) = 0.
      STRC3(ILOADX,5) = 0.
C
      STRM3(ILOADX,JCOL) = STRC3(ILOADX,JCOL)
C
C
      RETURN
      END
C
C
C
C
C=DECK      USRCON
      SUBROUTINE USRCON(INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,
     1  WORDCX,WORDMX,PCWORD,CPLOTX,ICARX,IFILEX)
```

```
C    PURPOSE: GENERATE USER-WRITTEN
C    INEQUALITY CONSTRAINT CONDITION
C    USING ANY COMBINATION OF PROGRAM
C    VARIABLES.
C    YOU MUST WRITE CODE THAT, USING
C    THE VARIABLES IN THE LABELLED
C    COMMON BLOCKS AS INPUT, ULTIMATELY
C    YIELDS A CONSTRAINT CONDITION,
C    CALLED "CONX" IN THIS ROUTINE.
      DIMENSION WORDCX(*),WORDMX(*),IPOINC(*),CONSTX(*)
      DIMENSION PCWORD(*),CPLOTX(*)
      CHARACTER*80 WORDCX,WORDMX,PCWORD
C   INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV03/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)
      REAL ALPHA2,TEMPER,DENSTY
      COMMON/FV21/PINNER(20)
      REAL PINNER
      COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )
      REAL STRM3,STRM3A,STRM3F
      COMMON/IV01/NMODUL
      INTEGER NMODUL
      COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT
      REAL TFINNR,TFOUTR,TFWEBS,WEIGHT
      COMMON/FV22/PMIDDL(20),POUTER(20)
      REAL PMIDDL,POUTER
C
      CONX = 0.0
C
C   INSERT USER-WRITTEN STATEMENTS
C   HERE. THE CONSTRAINT CONDITION
C   THAT YOU CALCULATE IS CALLED "CONX"
C
      IF (CONX.EQ.0.0) RETURN
      IF (CONX.LT.0.0) THEN
         WRITE(IFILEX,*)' CONX MUST BE GREATER THAN ZERO.'
         CALL EXIT
```

```
      ENDIF
C
C  DO NOT CHANGE THE FOLLOWING STATEMENTS, EXCEPT WORDC
C
      ICARX = ICARX + 1
      INUMTT = INUMTT + 1
      WORDCX(ICARX) = ' USER: PROVIDE THIS.'
      CPLOTX(ICARX) = CONX - 1.
      CALL BLANKX(WORDCX(ICARX),IENDP)
      PCWORD(ICARX)  = WORDCX(ICARX)(1:IENDP)//' -1'
      IF (IMODX.EQ.0.AND.CONX.GT.CONMAX) GO TO 200
      IF (IMODX.EQ.1.AND.IPOINC(INUMTT).EQ.0) GO TO 200
      ICONSX = ICONSX + 1
      IF (IMODX.EQ.0) IPOINC(INUMTT) = 1
      CONSTX(ICONSX) = CONX
      WORDMX(ICONSX)  = WORDCX(ICARX)(1:IENDP)//' -1'
  200 CONTINUE
C  END OF USRCON
C
C
      RETURN
      END
C
C
C
C=DECK      USRLNK
      SUBROUTINE USRLNK(VARI,I,VARIAB)
C Purpose: generate user-written
C linking conditions using any
C combination of decision variables.
C You must write conde that, using
C the variables in the subroutine
C argument VARIAB as input, ultimately
C yield a value for the linked variable
C VARI.
C
C VARI is the Ith entry of the array
C VARIAB.  You have decided that this
C is to be a linked variable with user
C defined linking.  It is linked to
C the decision variables in the array
C VARIAB.
C An example will provide the simplest
C explanation of this:
C Let"s say that the 5th decision
C variable candidate (I=5) is linked
C to the decision variable candidates
C 2 and 7. (You used DECIDE to select
```

```
C these as decision variables.
C In this case VARI is equal to
C VARIAB(I).  You then write your
C linking equation in the form
C VARI=f(VARIAB(2),VARIAB(7)).
C Use the index I in an IF statement if
C you have more than one user-defined
C linked variable.
C
C
      REAL VARI,VARIAB(50)
      INTEGER I
C
C  INSERT USER-WRITTEN DECLARATION
C  STATEMENTS HERE.
C
C  INSERT USER-WRITTEN
C  STATEMENTS HERE.
C
C
C  END OF USRLNK
      RETURN
      END
C=DECK      OBJECT
      SUBROUTINE OBJECT(IFILE,NPRINX,IMODX,OBJGEN,PHRASE)
C   PURPOSE:weight/length of the balloon
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/length of the balloon
C
       CHARACTER*80 PHRASE
C  INSERT ADDITIONAL COMMON BLOCKS:
      COMMON/FV03/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)
      REAL ALPHA2,TEMPER,DENSTY
      COMMON/FV21/PINNER(20)
      REAL PINNER
      COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )
      REAL STRM3,STRM3A,STRM3F
      COMMON/IV01/NMODUL
      INTEGER NMODUL
      COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT
      REAL TFINNR,TFOUTR,TFWEBS,WEIGHT
      COMMON/FV22/PMIDDL(20),POUTER(20)
      REAL PMIDDL,POUTER
C
C
C  INSERT SUBROUTINE STATEMENTS HERE.
C
C Get the weight per axial length of the balloon.
C The quantity, WEIGHT, is computed as follows in
C SUBROUTINE BOSDEC:
C
C     WEIGHT = 4.*(ARCOUT*TOUTER*DENSTY(1) +ARCINR*TINNER*DENSTY(1)
C    1        +ARCFOT*TFOUTR*DENSTY(2) +ARCFIN*TFINNR*DENSTY(2)
C    1        +ARCWEB*TFWEBS*DENSTY(3))
C
C in which
C    ARCOUT = total arc length of the outer curved membranes
C    ARCINR = total arc length of the inner curved membranes
C    ARCFOT = total arc length of the outer flat membranes
C    ARCFIN = total arc length of the inner flat membranes
C    ARCWEB = total length of the slanted webs.
```

```
C
      OBJGEN =WEIGHT
C
C
      RETURN
      END
C
C
C
```

**Table 6 This is the file, struct.balloon, which is the "fleshed out" version of the skeletal file, struct.new, that is automatically produced by GENTEXT. In this case the GENOPT user, whose duty it is to do the "fleshing out", added only three statements: CALL OPNGEN, CALL RWDGEN, and CALL CLSGEN. These three lines are shown in bold face.**

```
==========================================================================
C=DECK      STRUCT
      SUBROUTINE STRUCT(IMODX,CONSTX,OBJGEN,CONMAX,NCONSX,IPOINC,
     1 PCWORD,CPLOTX,ILOADX,ISTARX,NUSERC,IBEHV,IDV,IFAST,JJJ1)
C
C  PURPOSE IS TO PERFORM THE ANALYSIS FOR A GIVEN DESIGN AND LOADING.
C  CONSTRAINT CONDITIONS ARE ALSO GENERATED.
C
C  Common blocks already present in the struct.tmpl file, that is,
C  in the "skeletal" file possibly to be augmented by the user:
      COMMON/PRMFIL/IFILEX,IFILE2,IOUT,IPRM(5)
      COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
      COMMON/INDAT/INFILE
      COMMON/LWRUPR/VLBX(50),VUBX(50),CLINKX(50,45),VLINKX(50),VBVX(99)
      COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
      COMMON/PARAMS/PARX(99),VARX(50),ALLOWX(99),CONSXX(99),DECX(50),
     1              ESCX(50)
      COMMON/WORDS1/WORDPX(99),WORDVX(50),WORDAX(99),WORDCC(99),
     1              WORDDX(50)
      COMMON/WORDS2/WORDLX(50),WORDEX(50),WORDIQ(45)
      COMMON/OPTVAR/IDVX(50),ILVX(50),IDLINK(50,45),IEVX(50),JTERMS(45)
      COMMON/NUMPR2/ILARX,ICARX,IOARX,IFLATX,NCASES,NPRINX
      COMMON/PARAM2/FLARX(50),CARX(99),OARX(50),FSAFEX(99),CPWRX(50,45)
      COMMON/PARAM3/CINEQX(45,45),DPWREQ(45,45)
      COMMON/PARAM4/IDINEQ(45,45),NINEQX,JINEQX(45),IEQTYP(45)
      COMMON/WORDS3/WORDFX(50),WORDBX(99),WORDOB(50),WORDSX(99)
      COMMON/WORDS4/WORDMX(99)
      COMMON/PWORD/PHRASE
      COMMON/PWORD2/IBLANK
      COMMON/ISKIPX/ISKIP(30)
      DIMENSION IBEHV(99)
C
C=========================================================================
=
C  Start of first part of STRUCT written by "GENTEXT"
C  INSERT ADDITIONAL COMMON BLOCKS HERE: (THESE ARE "GENTEXT" VARIABLES)
```

```
      COMMON/FV03/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)
      REAL ALPHA2,TEMPER,DENSTY
      COMMON/FV21/PINNER(20)
      REAL PINNER
      COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )
      REAL STRM3,STRM3A,STRM3F
      COMMON/IV01/NMODUL
      INTEGER NMODUL
      COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT
      REAL TFINNR,TFOUTR,TFWEBS,WEIGHT
      COMMON/FV22/PMIDDL(20),POUTER(20)
      REAL PMIDDL,POUTER
C
C
      CHARACTER*80 PHRASE,CODPHR,PCWORD
      CHARACTER*80 WORDPX,WORDVX,WORDAX,WORDCX,WORDDX,WORDLX,WORDEX
      CHARACTER*80 WORDFX,WORDBX,WORDOB,WORDSX,WORDMX,WORDCC,WORDIQ
c     CHARACTER*4 ANSOUT,CHARAC,ANSWER
      CHARACTER*2 CIX
      character*2 CJX
      CHARACTER*13 CODNAM
c     DIMENSION ISUBX(100)
c     LOGICAL ANSL1
C
      DIMENSION CONSTX(*),IPOINC(*),PCWORD(*),CPLOTX(*)
C  End of first part of STRUCT written by "GENTEXT"
C=========================================================================
=
C
C  INSERT ADDITIONAL DIMENSION AND/OR LABELLED COMMON BLOCKS HERE,
C  IF NECESSARY. THESE WOULD BE STATEMENTS THAT ARE CONSISTENT WITH
C  SUBROUTINES THAT YOU OR OTHERS MAY HAVE WRITTEN THAT ARE REQUIRED
C  FOR WHATEVER ANALYSIS YOU ARE PERSUING.  MAKE SURE THAT YOU DO NOT
C  INTRODUCE NAME CONFLICTS WITH THE "GENTEXT" LABELLED COMMON BLOCKS
C  LISTED ABOVE.
```

```
C
C  Please note that you do not have to modify STRUCT.NEW if you would
C  rather provide all of your algorithms via the BEHAVIOR.NEW library.
C  (See instructions in BEHAVIOR.NEW).
C
C  If you are using a lot of software previously written either by
C  yourself or others, or if there are a lot of behavioral constraints
C  that are best generated by looping over array indices (such as
C  occurs, for example, with stress constraints in laminates of
C  composite materials), then it may be best to insert your common
C  blocks and dimension statements here, your subroutine calls
C  below (where indicated), and your subroutines in any of the libraries
C  called ADDCODEn.NEW, n = 1,2,...,5.  Please note that you
C  may also have to add statements to SUBROUTINE TRANFR, the
C  purpose of which is described below (in TRANFR).
C
C  The several test cases provided with GENOPT demonstrate different
C  methods:
C
C  PLATE  : leave STRUCT.NEW unchanged and modify BEHAVIOR.NEW
C  SPHERE : leave STRUCT.NEW unchanged and modify BEHAVIOR.NEW
C  TORISPH: leave BEHAVIOR.NEW unchanged except possibly for the objective
C           function (SUBROUTINE OBJECT), modify STRUCT.NEW,
C           possibly add a subroutine library called ADDCODE1.NEW, and
C           possibly augment the usermake.linux file to collect object
C           libraries from other directories. In the "TORISPH" case
C           BEHAVIOR.NEW remains unchanged, no ADDCODE1.NEW library is
C           added, and usermake.linux is not changed.  Instead, the
C           BIGBOSOR4 code is added and SUBROUTINE BOSDEC is written
C           by the genopt user. The BIGBOSOR4 code and SUBROUTINE
C           BOSDEC must be stored in /home/progs/bosdec/sources, as
C           follows:
C    BIGBOSOR4 code:
C     -rw-r--r--  1 bush bush 579671 Feb 29 07:19 addbosor4.src
C     -rw-r--r--  1 bush bush  83175 Feb 22 09:13 b4plot.src
C     -rw-r--r--  1 bush bush  89671 Feb 28 16:20 b4util.src
C     -rw-r--r--  1 bush bush  22723 Feb 10 14:27 bio.c
C     -rw-r--r--  1 bush bush  31175 Feb 10 14:27 bio_linux.c
C     -rw-r--r--  1 bush bush  37152 Feb 10 14:27 bio_linux.o
C     -rw-r--r--  1 bush bush  15650 Feb 10 14:26 gasp.F
C     -rw-r--r--  1 bush bush  18364 Feb 10 14:26 gasp_linux.o
C     -rw-r--r--  1 bush bush   6310 Feb 13 10:12 opngen.src
C     -rw-r--r--  1 bush bush  22440 Feb 10 14:25 prompter.src
C     -rw-r--r--  1 bush bush  13426 Feb 22 09:14 resetup.src
C    BOSDEC.src code:
C     -rw-r--r--  1 bush bush  33851 Mar  1 08:34 bosdec.src
C
```

```
C  WAVYCYL: both BEHAVIOR.NEW and STRUCT.NEW are both changed. Otherwise
C           the activity is the same as that described for TORISPH,
C           except, of course, that struct.new is different from
C           that used in connection with TORISPH.
C
C  CYLINDER:same as the description for WAVYCYL.
C
C
C  INSERT YOUR ADDITIONAL COMMON BLOCKS FOR THIS GENERIC CASE HERE:
C
C
C  THE FOLLOWING CODE WAS WRITTEN BY "GENTEXT":
C
C=====================================================================
C  Start the second portion of STRUCT written by "GENTEXT":
C
      ICARX   = ISTARX
      INUMTT = 0
      ICONSX = 0
      KCONX   = 0
      IF (IMODX.EQ.0) THEN
         CALL MOVERX(0.,0,CONSTX,1,99)
         CALL MOVERX(0, 0,IPOINC,1,1500)
      ENDIF
C
      IF (ILOADX.EQ.1) THEN
C
C  ESTABLISH FIRST ANY CONSTRAINTS THAT ARE INEQUALITY RELATIONSHIPS
C  AMONG THE VARIABLES IN THE ARRAY VARX(*) (THAT IS, VARIABLES THAT
C  ARE EITHER DECISION VARIABLES, LINKED VARIABLES, ESCAPE VARIABLES,
C  OR CANDIDATES FOR ANY OF THESE TYPES OF VARIABLES.
C
         IF (NINEQX.GT.0)
     1        CALL VARCON(WORDIQ,WORDMX,CINEQX,DPWREQ,IDINEQ,
     1        NINEQX,JINEQX,IEQTYP,INUMTT,IMODX,CONMAX,IPOINC,
     1        ICONSX,CONSTX,VARX,PCWORD,CPLOTX,ICARX)
C
C  NEXT, ESTABLISH USER-WRITTEN CONSTRAINTS. AT PRESENT, THE PROGRAM
C  ALLOWS ONLY ONE USER-WRITTEN CONSTRAINT. HOWEVER, THE USER CAN
C  EASILY EXPAND THIS CAPABILITY SIMPLY BY ADDING SUBROUTINES THAT
C  ARE ANALOGOUS TO USRCON (WITH NAMES SUCH AS USRCN2, USRCN3, ETC.
C  TO THE  BEHAVIOR.NEW LIBRARY, AND ADD CALLS TO THESE ADDITIONAL
C  SUBROUTINES FOLLOWING THE CALL TO USRCON IMMEDIATELY BELOW.
C
         CALL USRCON(INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1            WORDMX,PCWORD,CPLOTX,ICARX,IFILE8)
C
```

```
         NUSERC = ICARX - NINEQX
      ENDIF
C
      IF (NPRINX.GT.0) THEN
         WRITE(IFILE8,'(1X,A,I2,A)')
     1 ' BEHAVIOR FOR ',ILOADX,' ENVIRONMENT (LOAD SET)'
         WRITE(IFILE8,'(A)')'  '
         WRITE(IFILE8,'(A)')
     1 ' CONSTRAINT  BEHAVIOR              DEFINITION'
         WRITE(IFILE8,'(A)')
     1 '   NUMBER     VALUE'
      ENDIF
C
      CALL CONVR2(ILOADX,CIX)
      IF (NPRINX.GT.0) THEN
         WRITE(IFILE8,'(1X,A)')'  '
         WRITE(IFILE8,'(1X,A,I2)')
     1 ' BEHAVIOR FOR LOAD SET NUMBER, ILOADX=',ILOADX
      ENDIF
C
C  End of the second portion of STRUCT written by "GENTEXT"
C=======================================================================
C
C  USER: YOU MAY WANT TO INSERT SUBROUTINE CALLS FROM SOFTWARE DEVELOPED
C        ELSEWHERE FOR ANY CALCULATIONS PERTAINING TO THIS LOAD SET.
C
      CALL OPNGEN
      CALL RWDGEN
C
C=======================================================================
C  Start of the final portion of STRUCT written by "GENTEXT"
C
C  INSERT THE PROGRAM FILE HERE:
C
C  Behavior and constraints generated next for GENBUK:
C  GENBUK = general buckling load factor
C
      PHRASE =
     1 'general buckling load factor'
      CALL BLANKX(PHRASE,IENDP4)
      JXX = 0
      JXX = JXX + 1
      GENBUK(ILOADX) = 0.0
      IF (IBEHV(JXX).EQ.0) CALL BEHX1
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'general buckling load factor')
      IF (GENBUK(ILOADX  ).EQ.0.)  GENBUK(ILOADX  ) = 1.E+10
```

```
      IF (GENBUKA(ILOADX  ).EQ.0.)  GENBUKA(ILOADX  ) = 1.0
      IF (GENBUKF(ILOADX  ).EQ.0.)  GENBUKF(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =GENBUK(ILOADX  )
      WORDCX= '(GENBUK('//CIX//')/GENBUKA('//CIX//
     1  ')) / GENBUKF('//CIX//')'
      CALL CONX(GENBUK(ILOADX  ),GENBUKA(ILOADX  ),GENBUKF(ILOADX  )
     1,'general buckling load factor',
     1 'allowable for general buckling load factor',
     1 'general buckling factor of safety',
     1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  general buckling load factor: '
         IENDP4 =32
         CODNAM ='GENBUK('//CIX//')'
         MLET4 =6 + 4
         WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
         IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
     1    KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
  170 CONTINUE
  171 CONTINUE
C
C  Behavior and constraints generated next for STRM1:
C  STRM1 = stress component in material 1
C
      IF (JSTRM1 .EQ.0) GO TO 191
      IF (NPRINX.GT.0) THEN
         IF (JSTRM1 .GT.1) THEN
            WRITE(IFILE8,'(1X,A)')'  '
            WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
            WRITE(IFILE8,'(1X,A)')
     1      'stress component number'
         ENDIF
      ENDIF
      DO 190  J=1,JSTRM1
      CALL CONVR2(J,CJX)
      PHRASE =
     1 'stress component in material 1'
      CALL BLANKX(PHRASE,IENDP4)
      JXX = JXX + 1
      STRM1(ILOADX,J) = 0.0
      IF (IBEHV(JXX).EQ.0) CALL BEHX2
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
     1 'stress component in material 1')
```

```
      IF (STRM1(ILOADX,J).EQ.0.)  STRM1(ILOADX,J) = 1.E-10
      IF (STRM1A(ILOADX,J).EQ.0.)  STRM1A(ILOADX,J) = 1.0
      IF (STRM1F(ILOADX,J).EQ.0.)  STRM1F(ILOADX,J) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =STRM1(ILOADX,J)
      WORDCX= '(STRM1A('//CIX//','//CJX//')/STRM1('//CIX//','//CJX//
     1  ')) / STRM1F('//CIX//','//CJX//')'
      CALL CONX(STRM1(ILOADX,J),STRM1A(ILOADX,J),STRM1F(ILOADX,J)
     1,'stress component in material 1',
     1 'allowable stress in material 1',
     1 'factor of safety for stress in material 1',
     1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  stress component in material 1: '
         IENDP4 =34
         CODNAM ='STRM1('//CIX//','//CJX//')'
         MLET4 =5 + 7
         WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
         IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
     1    KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
  190 CONTINUE
  191 CONTINUE
C
C  Behavior and constraints generated next for STRM2:
C  STRM2 = stress component in material 2
C
      IF (JSTRM1 .EQ.0) GO TO 206
      IF (NPRINX.GT.0) THEN
         IF (JSTRM1 .GT.1) THEN
            WRITE(IFILE8,'(1X,A)')'   '
            WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
            WRITE(IFILE8,'(1X,A)')
     1      'stress component number'
         ENDIF
      ENDIF
      DO 205  J=1,JSTRM1
      CALL CONVR2(J,CJX)
      PHRASE =
     1 'stress component in material 2'
      CALL BLANKX(PHRASE,IENDP4)
      JXX = JXX + 1
      STRM2(ILOADX,J) = 0.0
      IF (IBEHV(JXX).EQ.0) CALL BEHX3
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
```

```
     1 'stress component in material 2')
      IF (STRM2(ILOADX,J).EQ.0.)  STRM2(ILOADX,J) = 1.E-10
      IF (STRM2A(ILOADX,J).EQ.0.)  STRM2A(ILOADX,J) = 1.0
      IF (STRM2F(ILOADX,J).EQ.0.)  STRM2F(ILOADX,J) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =STRM2(ILOADX,J)
      WORDCX= '(STRM2A('//CIX//','//CJX//')/STRM2('//CIX//','//CJX//
     1  ')) / STRM2F('//CIX//','//CJX//')'
      CALL CONX(STRM2(ILOADX,J),STRM2A(ILOADX,J),STRM2F(ILOADX,J)
     1,'stress component in material 2',
     1 'allowable for stress in material 2',
     1 'factor of safety for stress in material 2',
     1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  stress component in material 2: '
         IENDP4 =34
         CODNAM ='STRM2('//CIX//','//CJX//')'
         MLET4 =5 + 7
         WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
         IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
     1    KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
  205 CONTINUE
  206 CONTINUE
C
C  Behavior and constraints generated next for STRM3:
C  STRM3 = stress component in material 3
C
      IF (JSTRM1 .EQ.0) GO TO 221
      IF (NPRINX.GT.0) THEN
         IF (JSTRM1 .GT.1) THEN
            WRITE(IFILE8,'(1X,A)')'  '
            WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
            WRITE(IFILE8,'(1X,A)')
     1      'stress component number'
         ENDIF
      ENDIF
      DO 220  J=1,JSTRM1
      CALL CONVR2(J,CJX)
      PHRASE =
     1 'stress component in material 3'
      CALL BLANKX(PHRASE,IENDP4)
      JXX = JXX + 1
      STRM3(ILOADX,J) = 0.0
      IF (IBEHV(JXX).EQ.0) CALL BEHX4
```

```
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
     1 'stress component in material 3')
      IF (STRM3(ILOADX,J).EQ.0.)  STRM3(ILOADX,J) = 1.E-10
      IF (STRM3A(ILOADX,J).EQ.0.)  STRM3A(ILOADX,J) = 1.0
      IF (STRM3F(ILOADX,J).EQ.0.)  STRM3F(ILOADX,J) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =STRM3(ILOADX,J)
      WORDCX= '(STRM3A('//CIX//','//CJX//')/STRM3('//CIX//','//CJX//
     1  ')) / STRM3F('//CIX//','//CJX//')'
      CALL CONX(STRM3(ILOADX,J),STRM3A(ILOADX,J),STRM3F(ILOADX,J)
     1,'stress component in material 3',
     1 'allowable for stress in material 3',
     1 'factor of safety for stress in material 3',
     1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  stress component in material 3: '
         IENDP4 =34
         CODNAM ='STRM3('//CIX//','//CJX//')'
         MLET4 =5 + 7
         WORDBX(KCONX)= CODPHR(1:IENDP4)//CODNAM(1:MLET4)
         IF (NPRINX.GT.0) WRITE(IFILE8,'(I5,6X,G14.7,A,A)')
     1    KCONX,CARX(KCONX),CODPHR(1:IENDP4),CODNAM(1:MLET4)
      ENDIF
  220 CONTINUE
  221 CONTINUE
C
C  NEXT, EVALUATE THE OBJECTIVE, OBJGEN:
      IF (ILOADX.EQ.1) THEN
         PHRASE ='weight/length of the balloon'
         CALL BLANKX(PHRASE,IENDP4)
         CALL OBJECT(IFILE8,NPRINX,IMODX,OBJGEN,
     1    'weight/length of the balloon')
      ENDIF
      NCONSX = ICONSX
C
      CALL CLSGEN
C
      RETURN
      END
C
C
C
C
C
C  End of the final portion of STRUCT written by "GENTEXT"
```

```
C=============================================================================
C
C=DECK      TRANFR
      SUBROUTINE TRANFR(ARG1,ARG2,ARG3,ARG4,ARG5)
C
C  USER:  DO NOT FORGET TO MODIFY THE ARGUMENT LIST OF TRANFR AS
C         APPROPRIATE FOR YOUR CASE!
C
C  PURPOSE IS TO TRANSFER DATA FROM THE LABELLED COMMON BLOCKS
C  SET UP BY THE GENOPT CODE TO LABELLED COMMON OR ARGUMENTS IN
C  THE SUBROUTINE ARGUMENT LIST THAT MATCH PREVIOUSLY WRITTEN CODE
C  BY YOURSELF OR OTHER PROGRAM DEVELOPERS.  THE USER SHOULD ESTABLISH
C  THE ARGUMENT LIST AND/OR LABELLED COMMON BLOCKS THAT MATCH VARIABLES
C  IN THE PREVIOUSLY WRITTEN CODE.  FOR AN EXAMPLE, SEE THE DISCUSSION
C  OF THE CASE CALLED "PANEL".
C
C===============================================================================
=
C  Start of part of TRANFR written by "GENTEXT"
C  INSERT ADDITIONAL COMMON BLOCKS HERE: (THESE ARE "GENTEXT" VARIABLES)
      COMMON/FV03/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)
      REAL ALPHA2,TEMPER,DENSTY
      COMMON/FV21/PINNER(20)
      REAL PINNER
      COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )
      REAL STRM3,STRM3A,STRM3F
      COMMON/IV01/NMODUL
      INTEGER NMODUL
      COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT
      REAL TFINNR,TFOUTR,TFWEBS,WEIGHT
      COMMON/FV22/PMIDDL(20),POUTER(20)
      REAL PMIDDL,POUTER
C
C
C  End of part of TRANFR written by "GENTEXT"
```

```
C=========================================================================
=
C  INSERT ADDITIONAL DIMENSION AND/OR LABELLED COMMON BLOCKS HERE,
C  IF NECESSARY. THESE WOULD BE STATEMENTS THAT ARE CONSISTENT WITH
C  SUBROUTINES THAT YOU OR OTHERS MAY HAVE WRITTEN THAT ARE REQUIRED
C  FOR WHATEVER ANALYSIS YOU ARE NOW PERSUING.  MAKE SURE THERE ARE
C  NO NAME CONFLICTS WITH THE "GENTEXT" LABELLED COMMON BLOCKS.
C
C
C  INSERT APPROPRIATE FORTRAN STATEMENTS HERE (DON'T FORGET TO CORRECT
C  THE ARGUMENT LIST OF SUBROUTINE TRANFR!)
C  PROGRAM FILE:
C
C
      RETURN
      END
C
C
C
==================================================================
```

**Table 7 The file, bosdec.balloon, which is created entirely by the GENOPT user. One of the purposes of SUBROUTINE BOSDEC is to create a valid input file for BIGBOSOR4 with the use of data from the generic case, "balloon", for the current design and for perturbed designs. Because of SUBROUTINE BOSDEC, BIGBOSOR4 can be used in the optimization loop.**

```
===========================================================================
C=DECK      BOSDEC
C
C  PURPOSE IS TO SET UP BIGBOSOR4 INPUT FILE FOR "balloon"
C
      SUBROUTINE BOSDEC(INDX,IFIL14,ILOADX,INDIC)
C
C  Insert labelled common blocks: balloon.COM
      COMMON/FV03/EMOD1(10),IEMOD1
      REAL EMOD1
      COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)
      REAL EMOD2,G12,G13,G23,NU,ALPHA1
      COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER
      COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)
      REAL ALPHA2,TEMPER,DENSTY
      COMMON/FV21/PINNER(20)
      REAL PINNER
      COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)
      REAL GENBUK,GENBUKA,GENBUKF
      COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )
      REAL STRM1,STRM1A,STRM1F
      COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )
      REAL STRM2,STRM2A,STRM2F
      COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )
      REAL STRM3,STRM3A,STRM3F
      COMMON/IV01/NMODUL
      INTEGER NMODUL
      COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT
      REAL TFINNR,TFOUTR,TFWEBS,WEIGHT
      COMMON/FV22/PMIDDL(20),POUTER(20)
      REAL PMIDDL,POUTER
C  end of trusscomp.COM
      COMMON/ITRYX/ITRY
      COMMON/FLNFLO/FLINNR,FLOUTR
      COMMON/N2DIFX/N2DIFF(6)
      REAL N2DIFF
      COMMON/FINNER/C44FIN,DELTAT,DELT,NODSEG,MSEGS
```

```
      COMMON/MODULX/MODULL,MODULG
      COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
      COMMON/NUMSEG/NSEGS
      common/caseblock/CASE
      CHARACTER*28 CASE
      CHARACTER*35 CASA2,CASA3
      CHARACTER*10 CN
      COMMON/RBEGX/RBIG0,RBIGL,RBIGG
      DIMENSION RA(40),RB(40),ZA(40),ZB(40)
      DIMENSION RCL(40),RDL(40),ZCL(40),ZDL(40)
      DIMENSION R1(8,40),R2(8,40),Z1(8,40),Z2(8,40)
      DIMENSION RC(8,40),ZC(8,40),SROT(8,40),PFIXED(8,40),PEIGEN(8,40)
      DIMENSION MATTYP(8,40),THICK(8,40),MATLJ(295),THICKJ(295)
      DIMENSION LAYTYP(8,40),LTYPEJ(295)
      DIMENSION R1J(295),Z1J(295),R2J(295),Z2J(295),RCJ(295),ZCJ(295)
      DIMENSION NODJ(295),NSHPJ(295),SROTJ(295),PFIXJ(295),PEIGJ(295)
      DIMENSION NSHAPE(8,40),NGRND(8,40),NODGRD(8,40),NPREV(8,40)
      DIMENSION JPREV1(8,40),NDCUR1(8,40),NDPRV1(8,40)
      DIMENSION JPREV2(8,40),NDCUR2(8,40),NDPRV2(8,40)
      DIMENSION NGRNDJ(295),NODGRJ(295),NPREVJ(295)
      DIMENSION JPREVJ(295,2),NDCURJ(295,2),NDPRVJ(295,2)
      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 = 6
      PI = 3.1415927
      IF (INDX.EQ.0) THEN
         NMODUL = NMODUL
         RBEG = RBIGG
      ENDIF
      IF (INDX.EQ.1) THEN
         NMODUL = NMODUL
         RBEG = RBIGG
      ENDIF
      IF (INDX.EQ.2) THEN
         NMODUL = NMODUL
         RBEG = RBIGG
      ENDIF
      RSTART = RBEG
C
```

```
      CALL MOVER(0.,0,R1,1,320)
      CALL MOVER(0.,0,Z1,1,320)
      CALL MOVER(0.,0,R2,1,320)
      CALL MOVER(0.,0,Z2,1,320)
      CALL MOVER(0.,0,RC,1,320)
      CALL MOVER(0.,0,ZC,1,320)
      CALL MOVER(0.,0,PFIXED,1,320)
      CALL MOVER(0.,0,PEIGEN,1,320)
      CALL MOVER(0,0,NGRND,1,320)
      CALL MOVER(0,0,NODGRD,1,320)
      CALL MOVER(0,0,NPREV,1,320)
      CALL MOVER(0,0,JPREV1,1,320)
      CALL MOVER(0,0,NDCUR1,1,320)
      CALL MOVER(0,0,NDPRV1,1,320)
      CALL MOVER(0,0,JPREV2,1,320)
      CALL MOVER(0,0,NDCUR2,1,320)
      CALL MOVER(0,0,NDPRV2,1,320)
C
C  Single truss-core sandwich 6-segment module...
C
      IF (IMODX.EQ.0) MODULL = NMODUL
      MODULS = MODULL
C
C     The number of modules, "MODULG" should be such that the
C     model for general buckling subtends close to 90 degrees
C     of circumference...
C     IF (IMODX.EQ.0) MODULG = (PI/1.89)*RADIUS/PITCH
C     MODULS = MODULG
C     IF (IMODX.EQ.0.AND.NSEGSM*MODULS.GT.288) THEN
C        WRITE(IFILE4,'(/,A,/,A,2I5,/,A,/,A,/,A,)')
C    1 ' ************* TOO MANY MODULES *********************',
C    1 ' MODULS, MSEGS = NSEGSM*MODULS = ',MODULS, NSEGSM*MODULS,
C    1 ' MSEGS = Too many segments. Maximum number is 288.',
C    1 ' Now reducing the number of modules in the model.',
C    1 ' ***************************************************'
C234567890123456789012345678901234567890123456789012345678901234567890I2
C           RATIOM = FLOAT(280)/FLOAT(NSEGSM*MODULS)
C           MODULG = RATIOM*MODULG
C           MODULS = MODULG
C     ENDIF
C
C
C   INDX   = 1 means SUBROUTINE BOSDEC generates GENERAL buckling model
C   INDX   = 2 means SUBROUTINE BOSDEC computes maximum stresses.
C
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             NMODUL  = number of modules in the buckling model
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   NSHAPE = BIGBOSOR4 index for shape of shell segment:
C             NSHAPE = 1 means cone, cylinder, flat plate
C             NSHAPE = 2 means toroidal, spherical
C   SROT   = BIGBOSOR4 index for direction of travel along
C             a spherical or toroidal shell segment:
C             SROT = 1 means clockwise travel
C             SROT =-1 means anticlockwise travel
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
C
C23456789012345678901234567890123456789012345678901234567890123456789012
C
      DANGLE = 0.5*PI/FLOAT(2*MODULS)
      FLOUTR = 2.*(RADIUS + HEIGHT)*SIN(DANGLE)
      FLINNR = 2.*RADIUS*SIN(DANGLE)
      PHIOUT = ASIN(0.5*FLOUTR/ROUTER)
      PHIINR = ASIN(0.5*FLINNR/RINNER)
      DSMALO = (RADIUS+HEIGHT)*(1.-COS(DANGLE))
      DSMALI = RADIUS*(1.-COS(DANGLE))
      RCOUTR = RADIUS+HEIGHT -DSMALO - ROUTER*COS(PHIOUT)
      RCINNR = RADIUS -DSMALI + RINNER*COS(PHIINR)
C
      IF (IMODX.EQ.0) THEN
        WRITE(IFILE4,'(A,1P,E12.4)')
     1 ' DANGLE=',DANGLE
        WRITE(IFILE4,'(A,1P,E12.4)')
     1 ' FLOUTR=',FLOUTR
        WRITE(IFILE4,'(A,1P,E12.4)')
     1 ' FLINNR=',FLINNR
        WRITE(IFILE4,'(A,1P,E12.4)')
     1 ' PHIOUT=',PHIOUT
        WRITE(IFILE4,'(A,1P,E12.4)')
```

```
     1 ' PHIINR=',PHIINR
          WRITE(IFILE4,'(A,1P,E12.4)')
     1 ' DSMALO=',DSMALO
          WRITE(IFILE4,'(A,1P,E12.4)')
     1 ' DSMALI=',DSMALI
          WRITE(IFILE4,'(A,1P,E12.4)')
     1 ' RCOUTR=',RCOUTR
          WRITE(IFILE4,'(A,1P,E12.4)')
     1 ' RCINNR=',RCINNR
          WRITE(IFILE4,'(A,1P,E12.4)')
     1 ' RSTART=',RSTART
      ENDIF
C     CALL EXIT
C
      NODSEG = 31
C
      DO 100 IMODUL = 1,MODULS
C
         FMODUL = IMODUL
         ANGLE = DANGLE*(2.*FMODUL -1.0)
         ANGLEM = ANGLE - DANGLE
         ANGLEP = ANGLE + DANGLE
         RA(IMODUL) = RSTART + (RADIUS+HEIGHT)*SIN(ANGLE)
         ZA(IMODUL) = (RADIUS + HEIGHT)*COS(ANGLE)
         RB(IMODUL) = RSTART + RADIUS*SIN(ANGLEP)
         ZB(IMODUL) = RADIUS*COS(ANGLEP)
         RCL(IMODUL)= RSTART + RCOUTR*SIN(ANGLEM)
         ZCL(IMODUL)= RCOUTR*COS(ANGLEM)
         RDL(IMODUL)= RSTART + RCINNR*SIN(ANGLE)
         ZDL(IMODUL)= RCINNR*COS(ANGLE)
C
         IF (IMODX.EQ.0) THEN
           WRITE(IFILE4,'(A,I5)')
     1    ' IMODUL=',IMODUL
           WRITE(IFILE4,'(A,1P,E12.4)')
     1    ' ANGLE =',ANGLE
           WRITE(IFILE4,'(A,1P,E12.4)')
     1    ' ANGLEM=',ANGLEM
           WRITE(IFILE4,'(A,1P,E12.4)')
     1    ' ANGLEP=',ANGLEP
           WRITE(IFILE4,'(A,I2,A,1P,E12.4)')
     1    ' RA(',IMODUL,') =',RA(IMODUL)
           WRITE(IFILE4,'(A,I2,A,1P,E12.4)')
     1    ' ZA(',IMODUL,') =',ZA(IMODUL)
           WRITE(IFILE4,'(A,I2,A,1P,E12.4)')
     1    ' RB(',IMODUL,') =',RB(IMODUL)
           WRITE(IFILE4,'(A,I2,A,1P,E12.4)')
     1    ' ZB(',IMODUL,') =',ZB(IMODUL)
```

```
          WRITE(IFILE4,'(A,I2,A,1P,E12.4)')
     1   ' RCL(',IMODUL,')=',RCL(IMODUL)
          WRITE(IFILE4,'(A,I2,A,1P,E12.4)')
     1   ' ZCL(',IMODUL,')=',ZCL(IMODUL)
          WRITE(IFILE4,'(A,I2,A,1P,E12.4)')
     1   ' RDL(',IMODUL,')=',RDL(IMODUL)
          WRITE(IFILE4,'(A,I2,A,1P,E12.4)')
     1   ' ZDL(',IMODUL,')=',ZDL(IMODUL)
        ENDIF
  100 CONTINUE
C
C       CALL EXIT
C
      DO 110 IMODUL = 1,MODULS
C
C  Segment 1 in module, IMODUL
        IMODUL1 = IMODUL - 1
        NSHAPE(1,IMODUL) = 1
        IF (IMODUL.GT.1) THEN
          NGRND(1,IMODUL) = 0
          NPREV(1,IMODUL) = 1
          JPREV1(1,IMODUL) = -6
          NDCUR1(1,IMODUL) = 1
          NDPRV1(1,IMODUL) = NODSEG
          R1(1,IMODUL) = RA(IMODUL1)
          Z1(1,IMODUL) = ZA(IMODUL1)
        ELSE
          NGRND(1,IMODUL) = 1
          NODGRD(1,IMODUL) = 1
          NPREV(1,IMODUL) = 0
          R1(1,IMODUL) = RSTART
          Z1(1,IMODUL) = RADIUS + HEIGHT - DSMALO
        ENDIF
        R2(1,IMODUL) = RA(IMODUL)
        Z2(1,IMODUL) = ZA(IMODUL)
        MATTYP(1,IMODUL) = 2
        THICK(1,IMODUL) = TFOUTR
        LAYTYP(1,IMODUL) = 1
C
C  Segment 2 in module, IMODUL
        NSHAPE(2,IMODUL) = 2
        SROT(2,IMODUL) = 1.
        IF (IMODUL.GT.1) THEN
          NGRND(2,IMODUL) = 0
          NPREV(2,IMODUL) = 2
          JPREV1(2,IMODUL) = -7
          NDCUR1(2,IMODUL) = 1
          NDPRV1(2,IMODUL) = NODSEG
```

```
            JPREV2(2,IMODUL) = -1
            NDCUR2(2,IMODUL) = NODSEG
            NDPRV2(2,IMODUL) = NODSEG
            R1(2,IMODUL) = RA(IMODUL1)
            Z1(2,IMODUL) = ZA(IMODUL1)
          ELSE
            NGRND(2,IMODUL) = 1
            NODGRD(2,IMODUL) = 1
            NPREV(2,IMODUL) = 1
            JPREV1(2,IMODUL) = -1
            NDCUR1(2,IMODUL) = NODSEG
            NDPRV1(2,IMODUL) = NODSEG
            R1(2,IMODUL) = RSTART
            Z1(2,IMODUL) = RCOUTR + ROUTER
          ENDIF
          R2(2,IMODUL) = RA(IMODUL)
          Z2(2,IMODUL) = ZA(IMODUL)
          RC(2,IMODUL) = RCL(IMODUL)
          ZC(2,IMODUL) = ZCL(IMODUL)
          MATTYP(2,IMODUL) = 1
          THICK(2,IMODUL) = TOUTER
          LAYTYP(2,IMODUL) = 2
          PFIXED(2,IMODUL) = -PMIDDL(ILOADX)
          PEIGEN(2,IMODUL) =  POUTER(ILOADX)
C
C   Segment 3 in module, IMODUL
          NSHAPE(3,IMODUL) = 1
          IF (IMODUL.GT.1) THEN
            NGRND(3,IMODUL) = 0
            IF (IMODUL.EQ.MODULS) THEN
               NGRND(3,IMODUL) = 1
               NODGRD(3,IMODUL) = NODSEG
            ENDIF
            NPREV(3,IMODUL) = 1
            JPREV1(3,IMODUL) = -6
            NDCUR1(3,IMODUL) = 1
            NDPRV1(3,IMODUL) = NODSEG
            R1(3,IMODUL) = RB(IMODUL1)
            Z1(3,IMODUL) = ZB(IMODUL1)
          ELSE
            NGRND(3,IMODUL) = 1
            NODGRD(3,IMODUL) = 1
            NPREV(3,IMODUL) = 0
            R1(3,IMODUL) = RSTART
            Z1(3,IMODUL) = RADIUS
          ENDIF
          R2(3,IMODUL) = RB(IMODUL)
          Z2(3,IMODUL) = ZB(IMODUL)
```

```
         MATTYP(3,IMODUL) = 2
         THICK(3,IMODUL) = TFINNR
         LAYTYP(3,IMODUL) = 3
C
C  Segment 4 in module, IMODUL
         NGRND(4,IMODUL) = 0
         NPREV(4,IMODUL) = 2
         IF (IMODUL.EQ.1) JPREV1(4,IMODUL) = -1
         IF (IMODUL.GT.1) JPREV1(4,IMODUL) = -7
         NDCUR1(4,IMODUL) = 1
         IF (IMODUL.EQ.1) NDPRV1(4,IMODUL) = 1
         IF (IMODUL.GT.1) NDPRV1(4,IMODUL) = NODSEG
         JPREV2(4,IMODUL) = -1
         NDCUR2(4,IMODUL) = NODSEG
         NDPRV2(4,IMODUL) = NODSEG
         NSHAPE(4,IMODUL) = 2
         SROT(4,IMODUL) = -1.
         IF (IMODUL.GT.1) THEN
           R1(4,IMODUL) = RB(IMODUL1)
           Z1(4,IMODUL) = ZB(IMODUL1)
         ELSE
           R1(4,IMODUL) = RSTART
           Z1(4,IMODUL) = RADIUS
         ENDIF
         R2(4,IMODUL) = RB(IMODUL)
         Z2(4,IMODUL) = ZB(IMODUL)
         MATTYP(4,IMODUL) = 1
         THICK(4,IMODUL) = TINNER
         LAYTYP(4,IMODUL) = 4
         RC(4,IMODUL) = RDL(IMODUL)
         ZC(4,IMODUL) = ZDL(IMODUL)
         PFIXED(4,IMODUL) = PMIDDL(ILOADX) - PINNER(ILOADX)
         PEIGEN(4,IMODUL) =  0.
C
C  Segment 5 in module, IMODUL
         NGRND(5,IMODUL) = 0
         NPREV(5,IMODUL) = 2
         IF (IMODUL.EQ.1) JPREV1(5,IMODUL) = -2
         IF (IMODUL.GT.1) JPREV1(5,IMODUL) = -8
         NDCUR1(5,IMODUL) = 1
         IF (IMODUL.EQ.1) NDPRV1(5,IMODUL) = 1
         IF (IMODUL.GT.1) NDPRV1(5,IMODUL) = NODSEG
         JPREV2(5,IMODUL) = -4
         NDCUR2(5,IMODUL) = NODSEG
         NDPRV2(5,IMODUL) = NODSEG
         NSHAPE(5,IMODUL) = 1
         IF (IMODUL.GT.1) THEN
           R1(5,IMODUL) = RB(IMODUL1)
```

```
            Z1(5,IMODUL) = ZB(IMODUL1)
          ELSE
            R1(5,IMODUL) = RSTART
            Z1(5,IMODUL) = RADIUS
          ENDIF
          R2(5,IMODUL) = RA(IMODUL)
          Z2(5,IMODUL) = ZA(IMODUL)
          MATTYP(5,IMODUL) = 3
          THICK(5,IMODUL) = TFWEBS
          LAYTYP(5,IMODUL) = 5
C
C  Segment 6 in module, IMODUL
          NGRND(6,IMODUL) = 0
          NPREV(6,IMODUL) = 2
          JPREV1(6,IMODUL) = -3
          NDCUR1(6,IMODUL) = 1
          NDPRV1(6,IMODUL) = NODSEG
          JPREV2(6,IMODUL) = -5
          NDCUR2(6,IMODUL) = NODSEG
          NDPRV2(6,IMODUL) = NODSEG
          NSHAPE(6,IMODUL) = 1
          R1(6,IMODUL) = RB(IMODUL)
          Z1(6,IMODUL) = ZB(IMODUL)
          R2(6,IMODUL) = RA(IMODUL)
          Z2(6,IMODUL) = ZA(IMODUL)
          MATTYP(6,IMODUL) = 3
          THICK(6,IMODUL) = TFWEBS
          LAYTYP(6,IMODUL) = 5
C
C Additional two shell segments at the end of the model...
          IF (IMODUL.EQ.MODULS) THEN
            NGRND(7,IMODUL) = 1
            NODGRD(7,IMODUL) = NODSEG
            NPREV(7,IMODUL) = 1
            JPREV1(7,IMODUL) = -6
            NDCUR1(7,IMODUL) = 1
            NDPRV1(7,IMODUL) = NODSEG
            NSHAPE(7,IMODUL) = 1
            R1(7,IMODUL) = RA(IMODUL)
            Z1(7,IMODUL) = ZA(IMODUL)
            R2(7,IMODUL) = RSTART + RADIUS + HEIGHT -DSMALO
            Z2(7,IMODUL) = 0.
            MATTYP(7,IMODUL) = 2
            THICK(7,IMODUL) = TFOUTR
            LAYTYP(7,IMODUL) = 1
C
            NGRND(8,IMODUL) = 1
            NODGRD(8,IMODUL) = NODSEG
```

```
            NPREV(8,IMODUL) = 1
            JPREV1(8,IMODUL) = -7
            NDCUR1(8,IMODUL) = 1
            NDPRV1(8,IMODUL) = NODSEG
            NSHAPE(8,IMODUL) = 2
            SROT(8,IMODUL) = 1.
            R1(8,IMODUL) = RA(IMODUL)
            Z1(8,IMODUL) = ZA(IMODUL)
            R2(8,IMODUL) = RSTART + RCOUTR + ROUTER
            Z2(8,IMODUL) = 0.
            RC(8,IMODUL) = RSTART + RCOUTR
            ZC(8,IMODUL) = 0.
            MATTYP(8,IMODUL) = 1
            THICK(8,IMODUL) = TOUTER
            LAYTYP(8,IMODUL) = 2
            PFIXED(8,IMODUL) = -PMIDDL(ILOADX)
            PEIGEN(8,IMODUL) =  POUTER(ILOADX)
         ENDIF
  110 CONTINUE
C
      DO 115 J = 1,MODULS
      DO 114 I = 1,8
         WRITE(IFILE4,'(A,2I3,1P6E12.4)')
     1 ' J,I,R1(I,J),Z1(I,J),R2(I,J),Z2(I,J),RC(I,J),ZC(I,J)=',
     1   J,I,R1(I,J),Z1(I,J),R2(I,J),Z2(I,J),RC(I,J),ZC(I,J)
  114 CONTINUE
  115 CONTINUE
C     CALL EXIT
C
      ISEGT = 0
      DO 150 IMODUL = 1,MODULS
         DO 120 ISEG = 1,6
            ISEGT = ISEGT + 1
            NGRNDJ(ISEGT) = NGRND(ISEG,IMODUL)
            NODGRJ(ISEGT) = NODGRD(ISEG,IMODUL)
            NPREVJ(ISEGT) = NPREV(ISEG,IMODUL)
            JPREVJ(ISEGT,1) = JPREV1(ISEG,IMODUL)
            NDCURJ(ISEGT,1) = NDCUR1(ISEG,IMODUL)
            NDPRVJ(ISEGT,1) = NDPRV1(ISEG,IMODUL)
            JPREVJ(ISEGT,2) = JPREV2(ISEG,IMODUL)
            NDCURJ(ISEGT,2) = NDCUR2(ISEG,IMODUL)
            NDPRVJ(ISEGT,2) = NDPRV2(ISEG,IMODUL)
            NSHPJ(ISEGT) = NSHAPE(ISEG,IMODUL)
            NODJ(ISEGT) = NODSEG
            R1J(ISEGT) = R1(ISEG,IMODUL)
            Z1J(ISEGT) = Z1(ISEG,IMODUL)
            R2J(ISEGT) = R2(ISEG,IMODUL)
            Z2J(ISEGT) = Z2(ISEG,IMODUL)
```

```
            MATLJ(ISEGT) = MATTYP(ISEG,IMODUL)
            THICKJ(ISEGT) = THICK(ISEG,IMODUL)
            LTYPEJ(ISEGT) = LAYTYP(ISEG,IMODUL)
            IF (NSHAPE(ISEG,IMODUL).EQ.2) THEN
               RCJ(ISEGT) = RC(ISEG,IMODUL)
               ZCJ(ISEGT) = ZC(ISEG,IMODUL)
               SROTJ(ISEGT) = SROT(ISEG,IMODUL)
               PFIXJ(ISEGT) = PFIXED(ISEG,IMODUL)
               PEIGJ(ISEGT) = PEIGEN(ISEG,IMODUL)
            ENDIF
  120    CONTINUE
         IF (IMODUL.EQ.MODULS) THEN
          DO 130 ISEG = 7,8
            ISEGT = ISEGT + 1
            NGRNDJ(ISEGT) = NGRND(ISEG,IMODUL)
            NODGRJ(ISEGT) = NODGRD(ISEG,IMODUL)
            NPREVJ(ISEGT) = NPREV(ISEG,IMODUL)
            JPREVJ(ISEGT,1) = JPREV1(ISEG,IMODUL)
            NDCURJ(ISEGT,1) = NDCUR1(ISEG,IMODUL)
            NDPRVJ(ISEGT,1) = NDPRV1(ISEG,IMODUL)
            JPREVJ(ISEGT,2) = JPREV2(ISEG,IMODUL)
            NDCURJ(ISEGT,2) = NDCUR2(ISEG,IMODUL)
            NDPRVJ(ISEGT,2) = NDPRV2(ISEG,IMODUL)
            NSHPJ(ISEGT) = NSHAPE(ISEG,IMODUL)
            NODJ(ISEGT) = NODSEG
            R1J(ISEGT) = R1(ISEG,IMODUL)
            Z1J(ISEGT) = Z1(ISEG,IMODUL)
            R2J(ISEGT) = R2(ISEG,IMODUL)
            Z2J(ISEGT) = Z2(ISEG,IMODUL)
            MATLJ(ISEGT) = MATTYP(ISEG,IMODUL)
            THICKJ(ISEGT) = THICK(ISEG,IMODUL)
            LTYPEJ(ISEGT) = LAYTYP(ISEG,IMODUL)
            IF (NSHAPE(ISEG,IMODUL).EQ.2) THEN
               RCJ(ISEGT) = RC(ISEG,IMODUL)
               ZCJ(ISEGT) = ZC(ISEG,IMODUL)
               SROTJ(ISEGT) = SROT(ISEG,IMODUL)
               PFIXJ(ISEGT) = PFIXED(ISEG,IMODUL)
               PEIGJ(ISEGT) = PEIGEN(ISEG,IMODUL)
            ENDIF
  130     CONTINUE
         ENDIF
  150 CONTINUE
C
C Get correct temperature to generate membrane tension in the
C axial direction;
C
C
      FMODS = MODULS
```

```
      ARCOUT = 2.*PHIOUT*ROUTER*FMODS
      ARCINR = 2.*PHIINR*RINNER*FMODS
      SLANT = SQRT((R2J(5)-R1J(5))**2 +(Z2J(5)-Z1J(5))**2)
      ARCFOT = FLOUTR*FMODS
      ARCFIN = FLINNR*FMODS
      ARCWEB = 2.*SLANT*FMODS
      ENDFCE = PMIDDL(ILOADX)*PI*((RADIUS+HEIGHT)**2 -RADIUS**2)/4.
      FNU21O = EMOD1(1)*NU(1)/EMOD2(1)
      FNU21I = EMOD1(2)*NU(2)/EMOD2(2)
      FNU21M = EMOD1(3)*NU(3)/EMOD2(3)
      C22OUT = EMOD2(1)*TOUTER/(1.-NU(1)*FNU21O)
      C22INR = EMOD2(1)*TINNER/(1.-NU(1)*FNU21O)
      C22FOT = EMOD2(2)*TFOUTR/(1.-NU(2)*FNU21I)
      C22FIN = EMOD2(2)*TFINNR/(1.-NU(2)*FNU21I)
      C22WEB = EMOD2(3)*TFWEBS/(1.-NU(3)*FNU21M)
      C44FIN = (EMOD1(2)*TFINNR**3)/(12.*(1.-NU(2)*FNU21I))
C
      ARCTOT = ARCOUT*C22OUT*ALPHA2(1)
     1        +ARCINR*C22INR*ALPHA2(1)
     1        +ARCFOT*C22FOT*ALPHA2(2)
     1        +ARCFIN*C22FIN*ALPHA2(2)
     1        +ARCWEB*C22WEB*ALPHA2(3)
C
C Get the weight per axial length of the balloon:
C
      WEIGHT = 4.*(ARCOUT*TOUTER*DENSTY(1) +ARCINR*TINNER*DENSTY(1)
     1            +ARCFOT*TFOUTR*DENSTY(2) +ARCFIN*TFINNR*DENSTY(2)
     1            +ARCWEB*TFWEBS*DENSTY(3))
C
C  Load set A delta temperature = DELT:
      ENDF12 = 0.
      DELT = 0.
      IF (INDX.EQ.2) THEN
        WRITE(IFILE4,'(/,A,1P,6E12.4)')
     1' N2DIFF(J),J=1,6)=',(N2DIFF(J),J=1,6)
        ENDF12 = N2DIFF(1)*ARCFIN +N2DIFF(2)*ARCFOT +N2DIFF(3)*ARCFIN
     1          +N2DIFF(4)*ARCINR +N2DIFF(5)*ARCWEB
        DELT = -ENDF12/ARCTOT
      ENDIF
C
C  Load set B delta temperature = DELTAT:
      DELTAT = -ENDFCE/ARCTOT
C234567890123456789012345678901234567890123456789012345678901234567890
      WRITE(IFILE4,'(/,A,1P,2E12.4)')
     1' delta temps for generating axial resultants, DELTAT,DELT=',
     1 DELTAT,DELT
      WRITE(IFILE4,'(A,1P,E12.4)') ' PMIDDL(ILOADX)=',PMIDDL(ILOADX)
      WRITE(IFILE4,'(A,1P,E12.4)') ' FMODS =',FMODS
```

```
      WRITE(IFILE4,'(A,1P,E12.4)') ' ENDFCE=',ENDFCE
      WRITE(IFILE4,'(A,1P,E12.4)') ' ARCTOT=',ARCTOT
      WRITE(IFILE4,'(A,1P,E12.4)') ' ARCOUT=',ARCOUT
      WRITE(IFILE4,'(A,1P,E12.4)') ' ARCINR=',ARCINR
      WRITE(IFILE4,'(A,1P,E12.4)') ' ARCFOT=',ARCFOT
      WRITE(IFILE4,'(A,1P,E12.4)') ' ARCFIN=',ARCFIN
      WRITE(IFILE4,'(A,1P,E12.4)') ' ARCWEB=',ARCWEB
      WRITE(IFILE4,'(A,1P,E12.4)') ' SLANT =',SLANT
      WRITE(IFILE4,'(A,1P,E12.4)') ' C22OUT=',C22OUT
      WRITE(IFILE4,'(A,1P,E12.4)') ' C22INR=',C22INR
      WRITE(IFILE4,'(A,1P,E12.4)') ' C22FOT=',C22FOT
      WRITE(IFILE4,'(A,1P,E12.4)') ' C22FIN=',C22FIN
      WRITE(IFILE4,'(A,1P,E12.4)') ' C22WEB=',C22WEB
      WRITE(IFILE4,'(A,1P,E12.4)') ' FNU21O=',FNU21O
      WRITE(IFILE4,'(A,1P,E12.4)') ' FNU21I=',FNU21I
      WRITE(IFILE4,'(A,1P,E12.4)') ' FNU21M=',FNU21M
C     CALL EXIT
C
      MSEGS = ISEGT
C
C2345678901234567890123456789012345678901234567890123456789012345678901
C
C  Next, we generate a valid input data file, *.ALL, for BIGBOSOR4
C
C  Global input before segment data...
C
C     IF (INDX.EQ.0.OR.INDX.EQ.1) WRITE(IFIL14,'(A,I3,A)')
C    1' local buckling, ',MODULS,'-module model (INDIC=0) ixprism'
      IF (INDX.EQ.0) WRITE(IFIL14,'(A,I3,A)')
     1' Load B equilib, ',MODULS,'-module model (INDIC=0) ixprism'
      IF (INDX.EQ.1) WRITE(IFIL14,'(A,I3,A)')
     1' general buckling, ',MODULS,'-module model (INDIC=1) ixprism'
      IF (INDX.EQ.2) WRITE(IFIL14,'(A,I3,A)')
     1' stress components, ',MODULS,'-module model (INDIC=0) ixprism'
C     IF (INDX.EQ.0.OR.INDX.EQ.1)
C    1  WRITE(IFIL14,'(1P,E14.6,A)') LENGTH*FACLEN,
C    1' $ AXIALL = reduced axial length, LENGTH x FACLEN, local buck'
      IF (INDX.EQ.0.OR.INDX.EQ.1.OR.INDX.EQ.2)
     1  WRITE(IFIL14,'(1P,E14.6,A)') LENGTH,
     1' $ AXIALL = axial length of cyl.'
      IF (INDX.EQ.0.OR.INDX.EQ.1) WRITE(IFIL14,'(I5,A,I3,A)')
     1 INDIC,'  1, 0, ',MSEGS,' $ INDIC,NPRT,ISTRESS,NSEG'
      IF (INDX.EQ.2) WRITE(IFIL14,'(I5,A,I3,A)')
     1 INDIC,'  1, 1, ',MSEGS,' $ INDIC,NPRT,ISTRESS,NSEG'
C
C  Segment data...
C
C  First, provide the input for each of the MSEGS shell segments
```

```
C
      DO 200 ISEG = 1,MSEGS
         I = ISEG
         WRITE(IFIL14,'(A,4I6)')' H   $ Segment number ',I,I,I,I
         WRITE(IFIL14,'(I4,A,I3,A)') NODJ(ISEG),', 3, ',NSHPJ(ISEG),
     1 ' $ NMESH,NTYPEH,NSHAPE'
         WRITE(IFIL14,'(1P,4E14.6,A)')
     1   R1J(ISEG),Z1J(ISEG),R2J(ISEG),Z2J(iSEG),' $ R1,Z1,R2,Z2'
C END JUL 2010
         IF (NSHPJ(ISEG).EQ.2) THEN
          WRITE(IFIL14,'(1P,3E14.6,A)') RCJ(ISEG),ZCJ(ISEG),SROTJ(ISEG),
     1  ' $ RC,ZC,SROT'
         ENDIF
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
         WRITE(IFIL14,'(A,1P,E14.6,A)')' 0, 3, ',0.5*THICKJ(ISEG),
     1 ' $ IMP,NTYPEZ,ZVAL'
         WRITE(IFIL14,'(A)')' N  $ do not print r(s), etc.'
         NRINGS = 0
         FOUND = 0.
         LINTYP = 0
C
         IF (INDX.EQ.0) THEN
C
            IDISAB = 1
            WRITE(IFIL14,'(I5,1PE14.6,2I4,A)')
     1       NRINGS,FOUND,LINTYP,IDISAB, ' $ NRINGS,K,LINTYP,IDISAB'
C Load Set A...
            NLTYPE = 3
C normal pressure...
            NPSTAT = 2
            NLOAD1 = 0
            NLOAD2 = 0
            NLOAD3 = 1
            WRITE(IFIL14,'(5I5,A)')
     1      NLTYPE,NPSTAT,NLOAD1,NLOAD2,NLOAD3,
     1    '  $ NLTYPE,NPSTAT,NLOAD(1),NLOAD(2),NLOAD(3)'
            WRITE(IFIL14,'(1P,2E14.6,A)')
     1      PFIXJ(ISEG),PFIXJ(ISEG),'  $ PN(1),PN(2)'
            NTYPE = 1
            ISTA1 = 1
            ISTA2 = NODJ(ISEG)
            WRITE(IFIL14,'(3I5,A)')
     1      NTYPE,ISTA1,ISTA2,'  $ NTYPE,IPOINT(1),IPOINT(2)'
C
C temperature...
            NTSTAT = 2
            NTGRAD = 1
            NLOAD1 = 1
```

```
          NLOAD2 = 0
          NLOAD3 = 0
          WRITE(IFIL14,'(5I5,A)')
     1    NTSTAT,NTGRAD,NLOAD1,NLOAD2,NLOAD3,
     1  '  $ NTSTAT,NTGRAD,NLOAD(1),NLOAD(2),NLOAD(3)'
          MATJ = MATLJ(ISEG)
          WRITE(IFIL14,'(1P,2E14.6,A)')
     1    DELTAT,DELTAT,'  $ T1(1),T1(2)'
          NTYPE = 1
          ISTA1 = 1
          ISTA2 = NODJ(ISEG)
          WRITE(IFIL14,'(3I5,A)')
     1    NTYPE,ISTA1,ISTA2,'  $ NTYPE,IPOINT(1),IPOINT(2)'
C
C     End of INDX.EQ.0 condition
        ELSE
C     Begin  INDX.NE.0 condition
C
          IDISAB = 3
          WRITE(IFIL14,'(I5,1PE14.6,2I4,A)')
     1     NRINGS,FOUND,LINTYP,IDISAB, ' $ NRINGS,K,LINTYP,IDISAB'
C Load set A...
          NLTYPE = 3
C normal pressure...
          NPSTAT = 2
          NLOAD1 = 0
          NLOAD2 = 0
          NLOAD3 = 1
          WRITE(IFIL14,'(5I5,A)')
     1    NLTYPE,NPSTAT,NLOAD1,NLOAD2,NLOAD3,
     1  '  $ NLTYPE,NPSTAT,NLOAD(1),NLOAD(2),NLOAD(3)'
          WRITE(IFIL14,'(1P,2E14.6,A)')
     1    PEIGJ(ISEG),PEIGJ(ISEG),'  $ PN(1),PN(2)'
          NTYPE = 1
          ISTA1 = 1
          ISTA2 = NODJ(ISEG)
          WRITE(IFIL14,'(3I5,A)')
     1    NTYPE,ISTA1,ISTA2,'  $ NTYPE,IPOINT(1),IPOINT(2)'
C
C temperature...
          NTSTAT = 2
          NTGRAD = 1
          NLOAD1 = 1
          NLOAD2 = 0
          NLOAD3 = 0
          WRITE(IFIL14,'(5I5,A)')
     1    NTSTAT,NTGRAD,NLOAD1,NLOAD2,NLOAD3,
     1  '  $ NTSTAT,NTGRAD,NLOAD(1),NLOAD(2),NLOAD(3)'
```

```
            WRITE(IFIL14,'(1P,2E14.6,A)')
     1      DELT,DELT,'  $ T1(1),T1(2)'
            NTYPE = 1
            ISTA1 = 1
            ISTA2 = NODJ(ISEG)
            WRITE(IFIL14,'(3I5,A)')
     1      NTYPE,ISTA1,ISTA2,'  $ NTYPE,IPOINT(1),IPOINT(2)'
C
C Load Set B...
            NLTYPE = 3
C normal pressure...
            NPSTAT = 2
            NLOAD1 = 0
            NLOAD2 = 0
            NLOAD3 = 1
            WRITE(IFIL14,'(5I5,A)')
     1      NLTYPE,NPSTAT,NLOAD1,NLOAD2,NLOAD3,
     1    '  $ NLTYPE,NPSTAT,NLOAD(1),NLOAD(2),NLOAD(3)'
            WRITE(IFIL14,'(1P,2E14.6,A)')
     1      PFIXJ(ISEG),PFIXJ(ISEG),'  $ PN(1),PN(2)'
            NTYPE = 1
            ISTA1 = 1
            ISTA2 = NODJ(ISEG)
            WRITE(IFIL14,'(3I5,A)')
     1      NTYPE,ISTA1,ISTA2,'  $ NTYPE,IPOINT(1),IPOINT(2)'
C
C temperature...
            NTSTAT = 2
            NTGRAD = 1
            NLOAD1 = 1
            NLOAD2 = 0
            NLOAD3 = 0
            WRITE(IFIL14,'(5I5,A)')
     1      NTSTAT,NTGRAD,NLOAD1,NLOAD2,NLOAD3,
     1    '  $ NTSTAT,NTGRAD,NLOAD(1),NLOAD(2),NLOAD(3)'
            MATJ = MATLJ(ISEG)
            WRITE(IFIL14,'(1P,2E14.6,A)')
     1      DELTAT,DELTAT,'  $ T1(1),T1(2)'
            NTYPE = 1
            ISTA1 = 1
            ISTA2 = NODJ(ISEG)
            WRITE(IFIL14,'(3I5,A)')
     1      NTYPE,ISTA1,ISTA2,'  $ NTYPE,IPOINT(1),IPOINT(2)'
C
C   End of INDX.NE.0 condition
         ENDIF
C
C shell wall construction...
```

```
      NWALL = 4
      LAYERS = 1
      WRITE(IFIL14,'(2I3,A)') NWALL,LAYERS,' $ NWALL,NLAYER'
      WRITE(IFIL14,'(I3,A)') LTYPEJ(ISEG),' $ layer index'
      IF (ISEG.LE.5) THEN
         ANGLE2 = 0.
         WRITE(IFIL14,'(A)')' Y  $ is this a new layer type?'
         WRITE(IFIL14,'(1P2E14.6,I3,A)') THICKJ(ISEG),
     1    ANGLE2,MATLJ(ISEG),' $ thickness,angle,material'
      ELSE
         WRITE(IFIL14,'(A)')' N  $ is this a new layer type?'
      ENDIF
      NEWMAT = 0
      IF (ISEG.LE.5) THEN
         IF (ISEG.LE.2.OR.ISEG.EQ.5) NEWMAT = 1
      ENDIF
      IF (NEWMAT.EQ.0) THEN
         IF (ISEG.LE.5)
     1    WRITE(IFIL14,'(A)')' N  $ Is this material new?'
      ELSE
         IMATL = MATLJ(ISEG)
         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    TEMPER(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)'
         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)'
         ENDIF
         IF (IMATL.EQ.3) THEN
           WRITE(IFIL14,'(1P3E14.6,A)') STRM3A(ILOADX,1),
     1     STRM3A(ILOADX,2),STRM3A(ILOADX,3),' $ S(1),S(2),S(3)'
           WRITE(IFIL14,'(1P2E14.6,A)') STRM3A(ILOADX,4),
     1     STRM3A(ILOADX,5),' $ S(4),S(5)'
         ENDIF
      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
C  The old input for NWALL = 5 follows. (It is commented out).
C        WRITE(IFIL14,'(4I5,A)')
C    1   NWALL,LAYERS,NRS,NTYPET,'  $ NWALL,LAYERS,NRS,NTYPET'
C
C        MATJ = MATLJ(ISEG)
C        WRITE(IFIL14,'(1P,3E14.6,A)')
C    1   THICKJ(ISEG),G12(MATJ),EMOD1(MATJ),'  $ T,G,EX'
C        WRITE(IFIL14,'(1P,3E14.6,A)')
C    1   EMOD2(MATJ),NU(MATJ),DENSTY(MATJ),'  $ EY,UXY,SM'
C        WRITE(IFIL14,'(1P,2E14.6,A)')
C    1   ALPHA1(MATJ),ALPHA2(MATJ),'  $ ALPHA1,ALPHA2'
C
C        WRITE(IFIL14,'(A)')' N   $ do you want to print C(i,j)?'
C        WRITE(IFIL14,'(A)')' N   $ do you want to print distrib.loads?'
C23456789012345678901234567890123456789012345678901234567890123456789012
C
  200 CONTINUE
C End of loop over the number of shell segments in the model.
C

C Start the global data...
C
      NLAST = 0
      NOB = 1
      NMINB = 1
      NMAXB = 1
      INCRB = 1
      NVEC  = 1
      IF (INDX.EQ.0) THEN
         IF (ITRY.EQ.1) THEN
            PMULT = 0.1
            DPMULT= 0.1
            TEMP  = 0.1
            DTEMP = 0.1
            NUMSTP = 10
         ELSE
            IF (ITRY.EQ.3) THEN
               PMULT = 0.02
               DPMULT= 0.02
               TEMP  = 0.02
               DTEMP = 0.02
               NUMSTP = 50
            ELSE
               PMULT = 1.0
               DPMULT= 1.0
               TEMP  = 1.0
```

```
               DTEMP = 1.0
               NUMSTP = 1
            ENDIF
         ENDIF
      ELSE
         PMULT = 0.
         DPMULT = 1.0
         TEMP = 1.
         DTEMP = 0.
         NUMSTP = 2
      ENDIF
      OMEGA = 0.
      DOMEGA = 0.

      OMEGA = 0.
      DOMEGA = 0.
      WRITE(IFIL14,'(I5,A)')
     1 NLAST,'  $ NLAST'
      WRITE(IFIL14,'(A)')' N  $ are there any expanded plots?'
      IF (INDIC.EQ.1) WRITE(IFIL14,'(5I5,A)')
     1 NOB,NMINB,NMAXB,INCRB,NVEC,' $ NOB,NMINB,NMAXB,INCRB,NVEC'
      WRITE(IFIL14,'(1P,4E14.6,A)')
     1 PMULT,DPMULT,TEMP,DTEMP,'  $ P,DP,TEMP,DTEMP'
      IF (INDIC.EQ.0) WRITE(IFIL14,'(I5,A)')
     1 NUMSTP,'  $ number of load steps'
      WRITE(IFIL14,'(1P,2E14.6,A)')
     1 OMEGA,DOMEGA,'  $ OMEGA,DOMEGA'
C
C  Constraint conditions follow...
C
      WRITE(IFIL14,'(I5,A)')
     1 MSEGS,'  $ How many segments in the structure?'
C
      DO 300 I = 1,MSEGS
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)') NGRNDJ(I),' $ connect to ground'
         IF (NGRNDJ(I).GT.0) THEN
            NGRNDI = NGRNDJ(I)
            DO 250 J = 1,NGRNDI
               WRITE(IFIL14,'(I3,A)') NODGRJ(I),' $ node to ground'
               IF (I.EQ.1.OR.I.EQ.2.OR.I.EQ.3) THEN
                  IUSTAR = 0
                  IVSTAR = 0
                  IWSTAR = 1
                  ICHI   = 1
```

```
               ELSE
                  IUSTAR = 1
                  IVSTAR = 0
                  IWSTAR = 0
                  ICHI   = 1
               ENDIF
               WRITE(IFIL14,'(4I5,A)')
     1          IUSTAR,IVSTAR,IWSTAR,ICHI,
     1      ' $ IUSTAR,IVSTAR,IWSTAR,ICHI'
               D1 = 0.
               D2 = 0.
               WRITE(IFIL14,'(1P,2E14.6,A)') D1,D2,' $ D1,D2'
               WRITE(IFIL14,'(A)')' Y  $ is constraint same for buck.?'
  250       CONTINUE
         ENDIF
C23456789012345678901234567890123456789012345678901234567890123456789012
C
         IF (I.EQ.1.OR.I.EQ.3) THEN
             WRITE(IFIL14,'(A)')' N  $ joined to previous segments?'
         ELSE
             WRITE(IFIL14,'(A)')' Y  $ joined to previous segments?'
             WRITE(IFIL14,'(I3,A)') NPREVJ(I),' $ connects to prev.segs'
             NPREVI = NPREVJ(I)
             DO 220 J = 1,NPREVI
               WRITE(IFIL14,'(I3,A)') NDCURJ(I,J),' $ node current seg'
               WRITE(IFIL14,'(I3,A)')
     1                           I + JPREVJ(I,J),' $ prev.segment no.'
               WRITE(IFIL14,'(I3,A)') NDPRVJ(I,J),' $ node in prev.seg.'
               WRITE(IFIL14,'(A)')' 1, 1, 1, 0 $ IU,IV,IW,ICHI'
               WRITE(IFIL14,'(A)')' 0., 0. $ D1,D2'
               WRITE(IFIL14,'(A)')' Y  $ is constraint same for buck.?'
C23456789012345678901234567890123456789012345678901234567890123456789012
  220        CONTINUE
         ENDIF
C
  300 CONTINUE
C
      WRITE(IFIL14,'(A)')' N  $ are rigid body motions possible?'
C
      DO 410 ISEG = 1,MSEGS
         WRITE(IFIL14,'(A)')' Y  $ do you want to list seg. output?'
  410 CONTINUE
      WRITE(IFIL14,'(A)')' Y  $ do you want to list ring forces?'
C
      RETURN
      END
=========================================================================
```

## Table 8 The input file, try4.BEG, for the GENOPT processor called "BEGIN".

```
================================================================================
     n          $ Do you want a tutorial session and tutorial output?
   6000.000     $ length of the cylindrical shell: LENGTH
   120.0000     $ inner radius of the cylindrical balloon: RADIUS
        15      $ number of modules over 90 degrees: NMODUL
         3      $ Number IEMOD1  of rows in the array  EMOD1: IEMOD1
  435100.       $ elastic modulus, meridional direction: EMOD1(  1)
  435100.       $ elastic modulus, meridional direction: EMOD1(  2)
  435100.       $ elastic modulus, meridional direction: EMOD1(  3)
  435100.       $ elastic modulus, circumferential direction: EMOD2(  1)
  435100.       $ elastic modulus, circumferential direction: EMOD2(  2)
  435100.       $ elastic modulus, circumferential direction: EMOD2(  3)
  167346.       $ in-plane shear modulus: G12(  1)
  167346.     $ in-plane shear modulus: G12(  2)
  167346.     $ in-plane shear modulus: G12(  3)
  167346.     $ out-of-plane (s,z) shear modulus: G13(  1)
  167346.     $ out-of-plane (s,z) shear modulus: G13(  2)
  167346.     $ out-of-plane (s,z) shear modulus: G13(  3)
  167346.     $ out-of-plane (y,z) shear modulus: G23(  1)
  167346.     $ out-of-plane (y,z) shear modulus: G23(  2)
  167346.     $ out-of-plane (y,z) shear modulus: G23(  3)
  0.3000000     $ Poisson ratio: NU(  1)
  0.3000000     $ Poisson ratio: NU(  2)
  0.3000000     $ Poisson ratio: NU(  3)
  0.1000000E-09 $ meridional coef. thermal expansion: ALPHA1(  1)
  0.1000000E-09 $ meridional coef. thermal expansion: ALPHA1(  2)
  0.1000000E-09 $ meridional coef. thermal expansion: ALPHA1(  3)
  0.1000000E-03 $ circumf.coef.thermal expansion: ALPHA2(  1)
  0.1000000E-03 $ circumf.coef.thermal expansion: ALPHA2(  2)
  0.1000000E-03 $ circumf.coef.thermal expansion: ALPHA2(  3)
  -000.0000     $ delta-T from fabrication temperature: TEMPER(  1)
  -000.0000     $ delta-T from fabrication temperature: TEMPER(  2)
  -000.0000     $ delta-T from fabrication temperature: TEMPER(  3)
  0.1000000     $ weight density of material: DENSTY(  1)
  0.1000000     $ weight density of material: DENSTY(  2)
  0.1000000     $ weight density of material: DENSTY(  3)
   90.00000     $ height from inner to outer membranes: HEIGHT
   8.000000     $ radius of curvature of inner membrane: RINNER
   15.00000     $ radius of curvature of outer membrane: ROUTER
  0.1000000 $ thickness of the inner curved membrane: TINNER
  0.1000000 $ thickness of the outer curved membrane: TOUTER
  0.1000000 $ thickness of inner truss-core segment: TFINNR
  0.1000000 $ thickness of the outer truss segment: TFOUTR
  0.1000000 $ thickness of each truss-core web: TFWEBS
         1      $ Number NCASES  of load cases (environments): NCASES
   0.000000     $ pressure inside the inner membrane: PINNER(  1)
```

```
   60.00000     $ pressure between inner and outer membranes: PMIDDL(  1)
    5.00000     $ pressure outside the outer membrane: POUTER(  1)
   1.000000     $ allowable for general buckling load factor: GENBUKA(  1)
   3.000000     $ general buckling factor of safety: GENBUKF(  1)
         5      $ Number JSTRM1  of columns in the array, STRM1: JSTRM1
   10000.0     $ allowable stress in material 1: STRM1A(  1,  1)
   10000.0     $ allowable stress in material 1: STRM1A(  1,  2)
   10000.0     $ allowable stress in material 1: STRM1A(  1,  3)
   10000.0     $ allowable stress in material 1: STRM1A(  1,  4)
   10000.0     $ allowable stress in material 1: STRM1A(  1,  5)
   1.000000     $ factor of safety for stress in material 1: STRM1F(  1,  1)
   1.000000     $ factor of safety for stress in material 1: STRM1F(  1,  2)
   1.000000     $ factor of safety for stress in material 1: STRM1F(  1,  3)
   1.000000     $ factor of safety for stress in material 1: STRM1F(  1,  4)
   1.000000     $ factor of safety for stress in material 1: STRM1F(  1,  5)
   10000.0     $ allowable for stress in material 2: STRM2A(  1,  1)
   10000.0     $ allowable for stress in material 2: STRM2A(  1,  2)
   10000.0     $ allowable for stress in material 2: STRM2A(  1,  3)
   10000.0     $ allowable for stress in material 2: STRM2A(  1,  4)
   10000.0     $ allowable for stress in material 2: STRM2A(  1,  5)
   1.000000     $ factor of safety for stress in material 2: STRM2F(  1,  1)
   1.000000     $ factor of safety for stress in material 2: STRM2F(  1,  2)
   1.000000     $ factor of safety for stress in material 2: STRM2F(  1,  3)
   1.000000     $ factor of safety for stress in material 2: STRM2F(  1,  4)
   1.000000     $ factor of safety for stress in material 2: STRM2F(  1,  5)
   10000.0     $ allowable for stress in material 3: STRM3A(  1,  1)
   10000.0     $ allowable for stress in material 3: STRM3A(  1,  2)
   10000.0     $ allowable for stress in material 3: STRM3A(  1,  3)
   10000.0     $ allowable for stress in material 3: STRM3A(  1,  4)
   10000.0     $ allowable for stress in material 3: STRM3A(  1,  5)
   1.000000     $ factor of safety for stress in material 3: STRM3F(  1,  1)
   1.000000     $ factor of safety for stress in material 3: STRM3F(  1,  2)
   1.000000     $ factor of safety for stress in material 3: STRM3F(  1,  3)
   1.000000     $ factor of safety for stress in material 3: STRM3F(  1,  4)
   1.000000     $ factor of safety for stress in material 3: STRM3F(  1,  5)
=================================================================================
```

## Table 9 Input file, try4.DEC, for the GENOPT processor, "DECIDE"

```
==========================================================================
      n          $ Do you want a tutorial session and tutorial output?
         1       $ Choose a decision variable (1,2,3,...)
   20.00000      $ Lower bound of variable no.(  1)
  120.00000      $ Upper bound of variable no.(  1)
      n          $ Do you want especially to restrict variable no.(  1)
      y          $ Any more decision variables (Y or N) ?
         4       $ Choose a decision variable (1,2,3,...)
  0.03000000      $ Lower bound of variable no.(  4)
  0.3000000      $ Upper bound of variable no.(  4)
      n          $ Do you want especially to restrict variable no.(  4)
      y          $ Any more decision variables (Y or N) ?
         5       $ Choose a decision variable (1,2,3,...)
  0.0300000      $ Lower bound of variable no.(  5)
  0.3000000      $ Upper bound of variable no.(  5)
      n          $ Do you want especially to restrict variable no.(  5)
      y          $ Any more decision variables (Y or N) ?
         6       $ Choose a decision variable (1,2,3,...)
  0.0300000      $ Lower bound of variable no.(  6)
  0.3000000      $ Upper bound of variable no.(  6)
      n          $ Do you want especially to restrict variable no.(  6)
      y          $ Any more decision variables (Y or N) ?
         7       $ Choose a decision variable (1,2,3,...)
  0.0300000      $ Lower bound of variable no.(  7)
  0.3000000      $ Upper bound of variable no.(  7)
      n          $ Do you want especially to restrict variable no.(  7)
      y          $ Any more decision variables (Y or N) ?
         8       $ Choose a decision variable (1,2,3,...)
  0.0300000      $ Lower bound of variable no.(  8)
  0.3000000      $ Upper bound of variable no.(  8)
      n          $ Do you want especially to restrict variable no.(  8)
      n          $ Any more decision variables (Y or N) ?
      n          $ Any linked variables (Y or N) ?
      n          $ Any inequality relations among variables? (type H)
      y          $ Any escape variables (Y or N) ?
      y          $ Want to have escape variables chosen by default?
======================================================================
```

## Table 10 Input file, try4.OPT, for the GENOPT processor called "MAINSETUP"

```
=============================================================================
      n          $ Do you want a tutorial session and tutorial output?
         0       $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         2       $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
         2       $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
         5       $ How many design iterations in this run (3 to 25)?
      n          $ Take "shortcuts" for perturbed designs (Y or N)?
         1       $ Choose 1 or 2 or 3 or 4 or 5 for IDESIGN
         1       $ Choose 1 or 2 or 3 or 4 or 5 for move limits, IMOVE
      y          $ Do you want default (RATIO=10) for initial move limit jump?
      y          $ Do you want the default perturbation (dx/x = 0.05)?
      n          $ Do you want to have dx/x modified by GENOPT?
      n          $ Do you want to reset total iterations to zero (Type H)?
         1       $ Choose IAUTOF= 1 or 2 or 3 or 4 or 5 or 6 to change X(i)
=============================================================================
```

**Table 11 Output file, try4.OPM, from OPTIMIZE for the optimized design after the first execution of SUPEROPT. This try4.OPM file has been edited slightly for ease of reading. Items in bold face are discussed in Section 8 of the text and identified by an Item Number.**

```
===============================================================================
      n         $ Do you want a tutorial session and tutorial output?
         0      $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         2      $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
Item 1   2      $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
         5      $ How many design iterations in this run (3 to 25)?
      n         $ Take "shortcuts" for perturbed designs (Y or N)?
         2      $ Choose 1 or 2 or 3 or 4 or 5 for IDESIGN
         1      $ Choose 1 or 2 or 3 or 4 or 5 for move limits, IMOVE
      y         $ Do you want default (RATIO=10) for initial move limit jump?
      y         $ Do you want the default perturbation (dx/x = 0.05)?
      n         $ Do you want to have dx/x modified by GENOPT?
      n         $ Do you want to reset total iterations to zero (Type H)?
         1      $ Choose IAUTOF= 1 or 2 or 3 or 4 or 5 or 6 to change X(i)

 ************ END OF THE  try4.OPT  FILE ***********
 ************ AUGUST, 2010 VERSION OF GENOPT ***************
 ******** BEGINNING OF THE  try4.OPM  FILE *********


 *********************** MAIN PROCESSOR *********************
 The purpose of the mainprocessor, OPTIMIZE, is to perform,
 in a batch mode, the work specified by MAINSETUP for the case
 called try4.  Results are stored in the file  try4.OPM.
 Please inspect try4.OPM before doing more design iterations.
 ************************************************************

 Item 2: STRUCTURAL ANALYSIS FOR DESIGN ITERATION NO.   0:

  STRUCTURAL ANALYSIS WITH UNPERTURBED DECISION VARIABLES
 VAR. DEC. ESCAPE LINK. LINKED  LINKING    LOWER     CURRENT     UPPER      DEFINITION
  NO. VAR.  VAR.  VAR.    TO    CONSTANT   BOUND      VALUE      BOUND
   1   Y     N     N      0    0.00E+00  2.00E+01 5.8750E+01  1.20E+02  height from inner to outer
membranes: HEIGHT
   2   N     N     N      0    0.00E+00  0.00E+00 8.0000E+00  0.00E+00  radius of curvature of
inner membrane: RINNER
   3   N     N     N      0    0.00E+00  0.00E+00 1.5000E+01  0.00E+00  radius of curvature of
outer membrane: ROUTER
   4   Y     N     N      0    0.00E+00  3.00E-02 4.8731E-02  3.00E-01  thickness of the inner
curved membrane: TINNER
   5   Y     N     N      0    0.00E+00  3.00E-02 8.0927E-02  3.00E-01  thickness of the outer
curved membrane: TOUTER
   6   Y     N     N      0    0.00E+00  3.00E-02 1.3853E-01  3.00E-01  thickness of inner truss-
core segment: TFINNR
```

```
   7   Y     Y     N      0    0.00E+00  3.00E-02 3.0000E-02  3.00E-01  thickness of the outer
truss segment: TFOUTR
   8   Y     Y     N      0    0.00E+00  3.00E-02 4.7662E-02  3.00E-01  thickness of each truss-
core web: TFWEBS
```

BIGBOSOR4 input file for: general buckling load

**Item 3: try4.BEHX1**

Changes in temperature required to create 2 total axial loads:

1. Change in temperature required to create the axial thermal strain that generates the axial tension due to closing the two ends of the pressurized volume (PMIDDL= 6.0000E+01) between the inner and outer walls of the balloon in

**Item 4:** **Load Step No. 1: DELTAT= -1.1210E+02**

2. Change in temperature required to simulate the Poisson axial expansion caused by the application of the outer pressure, POUTER = 5.0000E+00 in **Load Step No. 2: DELT=0.00**

**Item 5:**

**GENERAL BUCKLING LOAD FACTORS AND MODES (BEHX1)**

**Item 5a: 2.9921E+00 (1 axial half-wave over LENGTH=6000 inches)**

Critical buckling load factor, GENBUK= 2.9921E+00

**Item 5b: Critical number of axial half-waves, NWVCRT = 1**

Differences in the resultants along the axis of the prismatic balloon for each segment, J, of the first module:

**Item 6:**

```
 [N2VAR(J) for the total load] - [N2FIX(J) for the fixed load]=
 N2DIFF(J),J=1,6)= -3.2748E+01 -3.2453E+01 -2.0757E+02 -4.3006E+00
                   -1.2398E+01 -1.2390E+01
 N2VAR(J) (total load) are the resultants from Load Step No. 2.
 N2FIX(J) (fixed load) are the resultants from Load Step No. 1.
 NOTE: The stresses used as behavioral constraints are
       computed from N2VAR(J)/thickness(J). These stresses are
       lower than those computed from N2FIX(J)/thickness(J).
```

**Item 7:** PREBUCKLING STRESS RESULTANTS IN THE FIRST MODULE

"fixed" from Load Step No. 1 total from Load Step No. 2

| Seg.J | Node I | N1FIX(I,J) | N2FIX(I,J) | N1VAR(I,J) | N2VAR(I,J) |
|---|---|---|---|---|---|
| 1 | 1 | 4.0941E+02 | 2.6914E+02 | 3.0026E+02 | 2.3640E+02 |
| 2 | 1 | 9.4526E+02 | 6.7828E+02 | 8.3708E+02 | 6.4583E+02 |
| **3** | **1** | **2.0636E+03** | **1.2948E+03** | **1.3718E+03** | **1.0872E+03** |
| 4 | 1 | 4.9948E+02 | 3.8752E+02 | 4.8514E+02 | 3.8321E+02 |

```
   5       1   5.1709E+02   3.8759E+02   4.7576E+02   3.7519E+02
   6       1   5.1696E+02   3.8755E+02   4.7566E+02   3.7516E+02
```

**Item 8:**

**PREBUCKLING MEMBRANE STRESSES IN THE FIRST MODULE COMPUTED FROM N1FIX/thickness, N2FIX/thickness, N1VAR/thickness, N2VAR/thickness: "fixed" from Load Step No. 1 total from Load Step No. 2**

| Seg.J | Node I | STRS1F(I,J) | STRS2F(I,J) | STRS1V(I,J) | STRS2V(I,J) |
|---|---|---|---|---|---|
| 1 | 1 | 1.3647E+04 | 8.9714E+03 | 1.0009E+04 | 7.8798E+03 |
| 2 | 1 | 1.1680E+04 | 8.3814E+03 | 1.0344E+04 | 7.9804E+03 |
| 3 | 1 | 1.4897E+04 | 9.3462E+03 | 9.9020E+03 | 7.8479E+03 |
| 4 | 1 | 1.0250E+04 | 7.9522E+03 | 9.9556E+03 | 7.8640E+03 |
| 5 | 1 | 1.0849E+04 | 8.1320E+03 | 9.9820E+03 | 7.8719E+03 |

**Item 9:**

**Newton iterations required to solve the nonlinear axisymmetric pre-buckling equilibrium state for the "fixed" loads (PINNER, PMIDDL, DELTAT):        ITER=  7**
**Newton iterations required to solve the nonlinear axisymmetric pre-buckling equilibrium state for the total loads (PINNER, PMIDDL, DELTAT, POUTER): ITER=  2**

```
BEHAVIOR FOR LOAD SET NUMBER, ILOADX= 1
 CONSTRAINT  BEHAVIOR              DEFINITION
   NUMBER     VALUE
```

**Item 10:**

**1      2.992076    general buckling load factor: GENBUK(1)**

```
  BEHAVIOR OVER J =  stress component number
    2        10343.62      stress component in material 1: STRM1(1 ,1 )
    3       0.1000000E-09  stress component in material 1: STRM1(1 ,2 )
    4        7980.362      stress component in material 1: STRM1(1 ,3 )
    5       0.1000000E-09  stress component in material 1: STRM1(1 ,4 )
    6       0.1000000E-09  stress component in material 1: STRM1(1 ,5 )

  BEHAVIOR OVER J =  stress component number
    7        10008.54      stress component in material 2: STRM2(1 ,1 )
    8       0.1000000E-09  stress component in material 2: STRM2(1 ,2 )
    9        7879.838      stress component in material 2: STRM2(1 ,3 )
   10       0.1000000E-09  stress component in material 2: STRM2(1 ,4 )
   11       0.1000000E-09  stress component in material 2: STRM2(1 ,5 )

  BEHAVIOR OVER J =  stress component number
   12        9982.016      stress component in material 3: STRM3(1 ,1 )
   13       0.1000000E-09  stress component in material 3: STRM3(1 ,2 )
   14        7871.880      stress component in material 3: STRM3(1 ,3 )
```

```
  15       0.1000000E-09  stress component in material 3: STRM3(1 ,4 )
  16       0.1000000E-09  stress component in material 3: STRM3(1 ,5 )

 ***** RESULTS FOR LOAD SET NO.  1  ******
 PARAMETERS WHICH DESCRIBE BEHAVIOR (e.g. stress, buckling load)

 BEH.   CURRENT
 NO.     VALUE            DEFINITION
  1    2.992E+00    general buckling load factor: GENBUK(1 )
  2    1.034E+04    stress component in material 1: STRM1(1 ,1 )
  3    1.000E-10    stress component in material 1: STRM1(1 ,2 )
  4    7.980E+03    stress component in material 1: STRM1(1 ,3 )
  5    1.000E-10    stress component in material 1: STRM1(1 ,4 )
  6    1.000E-10    stress component in material 1: STRM1(1 ,5 )
  7    1.001E+04    stress component in material 2: STRM2(1 ,1 )
  8    1.000E-10    stress component in material 2: STRM2(1 ,2 )
  9    7.880E+03    stress component in material 2: STRM2(1 ,3 )
 10    1.000E-10    stress component in material 2: STRM2(1 ,4 )
 11    1.000E-10    stress component in material 2: STRM2(1 ,5 )
 12    9.982E+03    stress component in material 3: STRM3(1 ,1 )
 13    1.000E-10    stress component in material 3: STRM3(1 ,2 )
 14    7.872E+03    stress component in material 3: STRM3(1 ,3 )
 15    1.000E-10    stress component in material 3: STRM3(1 ,4 )
 16    1.000E-10    stress component in material 3: STRM3(1 ,5 )

 ******* NOTE ******* NOTE ******* NOTE ****** NOTE ******
 The phrase, "NOT APPLY", for MARGIN VALUE means that that
 particular margin value is exactly zero.
 *** END NOTE *** END NOTE *** END NOTE *** END NOTE *****


 ***** RESULTS FOR LOAD SET NO.  1  ******
 MARGINS CORRESPONDING TO CURRENT DESIGN (F.S.= FACTOR OF SAFETY)
```

**Item 11:**

```
 MARGIN CURRENT
 NO.     VALUE            DEFINITION
  1   -2.641E-03  (GENBUK(1 )/GENBUKA(1 )) / GENBUKF(1 )-1; F.S.=  3.00
  2   -3.322E-02  (STRM1A(1 ,1 )/STRM1(1 ,1 )) / STRM1F(1 ,1 )-1; F.S.=  1.00
  3    2.531E-01  (STRM1A(1 ,3 )/STRM1(1 ,3 )) / STRM1F(1 ,3 )-1; F.S.=  1.00
  4   -8.538E-04  (STRM2A(1 ,1 )/STRM2(1 ,1 )) / STRM2F(1 ,1 )-1; F.S.=  1.00
  5    2.691E-01  (STRM2A(1 ,3 )/STRM2(1 ,3 )) / STRM2F(1 ,3 )-1; F.S.=  1.00
  6    1.802E-03  (STRM3A(1 ,1 )/STRM3(1 ,1 )) / STRM3F(1 ,1 )-1; F.S.=  1.00
  7    2.703E-01  (STRM3A(1 ,3 )/STRM3(1 ,3 )) / STRM3F(1 ,3 )-1; F.S.=  1.00

 ***************************************************
 ******************** DESIGN OBJECTIVE ******************
 *************                             ***********
```

**Item 12:    CURRENT VALUE OF THE OBJECTIVE FUNCTION:**

```
VAR.   CURRENT
NO.     VALUE              DEFINITION
 1    6.173E+01  weight/length of the balloon: WEIGHT

*************                          ************
******************* DESIGN OBJECTIVE ******************
*****************************************************
***********  ALL 1 LOAD CASES PROCESSED ***********
*****************************************************


PARAMETERS WHICH ARE ALWAYS FIXED.  NONE CAN BE DECISION VARIAB.

VAR.   CURRENT
NO.     VALUE            DEFINITION
```

**Item 13:**

```
  1    6.000E+03  length of the cylindrical shell: LENGTH
  2    1.200E+02  inner radius of the cylindrical balloon: RADIUS
  3    4.351E+05  elastic modulus, meridional direction: EMOD1(1 )
  4    4.351E+05  elastic modulus, meridional direction: EMOD1(2 )
  5    4.351E+05  elastic modulus, meridional direction: EMOD1(3 )
  6    4.351E+05  elastic modulus, circumferential direction: EMOD2(1 )
  7    4.351E+05  elastic modulus, circumferential direction: EMOD2(2 )
  8    4.351E+05  elastic modulus, circumferential direction: EMOD2(3 )
  9    1.673E+05  in-plane shear modulus: G12(1 )
 10    1.673E+05  in-plane shear modulus: G12(2 )
 11    1.673E+05  in-plane shear modulus: G12(3 )
 12    1.673E+05  out-of-plane (s,z) shear modulus: G13(1 )
 13    1.673E+05  out-of-plane (s,z) shear modulus: G13(2 )
 14    1.673E+05  out-of-plane (s,z) shear modulus: G13(3 )
 15    1.673E+05  out-of-plane (y,z) shear modulus: G23(1 )
 16    1.673E+05  out-of-plane (y,z) shear modulus: G23(2 )
 17    1.673E+05  out-of-plane (y,z) shear modulus: G23(3 )
 18    3.000E-01  Poisson ratio: NU(1 )
 19    3.000E-01  Poisson ratio: NU(2 )
 20    3.000E-01  Poisson ratio: NU(3 )
 21    1.000E-10  meridional coef. thermal expansion: ALPHA1(1 )
 22    1.000E-10  meridional coef. thermal expansion: ALPHA1(2 )
 23    1.000E-10  meridional coef. thermal expansion: ALPHA1(3 )
 24    1.000E-04  circumf.coef.thermal expansion: ALPHA2(1 )
 25    1.000E-04  circumf.coef.thermal expansion: ALPHA2(2 )
 26    1.000E-04  circumf.coef.thermal expansion: ALPHA2(3 )
 27    0.000E+00  delta-T from fabrication temperature: TEMPER(1 )
 28    0.000E+00  delta-T from fabrication temperature: TEMPER(2 )
 29    0.000E+00  delta-T from fabrication temperature: TEMPER(3 )
 30    1.000E-01  weight density of material: DENSTY(1 )
 31    1.000E-01  weight density of material: DENSTY(2 )
 32    1.000E-01  weight density of material: DENSTY(3 )
```

PARAMETERS WHICH ARE ENVIRONMENTAL FACTORS (e.g. loads, temps.)

| VAR. NO. | CURRENT VALUE | DEFINITION |
|---|---|---|
| 1 | 0.000E+00 | pressure inside the inner membrane: PINNER(1 ) |
| 2 | 6.000E+01 | pressure between inner and outer membranes: PMIDDL(1 ) |
| 3 | 5.000E+00 | pressure outside the outer membrane: POUTER(1 ) |

PARAMETERS WHICH ARE CLASSIFIED AS ALLOWABLES (e.g. max. stress)

| VAR. NO. | CURRENT VALUE | DEFINITION |
|---|---|---|
| 1 | 1.000E+00 | allowable for general buckling load factor: GENBUKA(1 ) |
| 2 | 1.000E+04 | allowable stress in material 1: STRM1A(1 ,1 ) |
| 3 | 1.000E+04 | allowable stress in material 1: STRM1A(1 ,2 ) |
| 4 | 1.000E+04 | allowable stress in material 1: STRM1A(1 ,3 ) |
| 5 | 1.000E+04 | allowable stress in material 1: STRM1A(1 ,4 ) |
| 6 | 1.000E+04 | allowable stress in material 1: STRM1A(1 ,5 ) |
| 7 | 1.000E+04 | allowable for stress in material 2: STRM2A(1 ,1 ) |
| 8 | 1.000E+04 | allowable for stress in material 2: STRM2A(1 ,2 ) |
| 9 | 1.000E+04 | allowable for stress in material 2: STRM2A(1 ,3 ) |
| 10 | 1.000E+04 | allowable for stress in material 2: STRM2A(1 ,4 ) |
| 11 | 1.000E+04 | allowable for stress in material 2: STRM2A(1 ,5 ) |
| 12 | 1.000E+04 | allowable for stress in material 3: STRM3A(1 ,1 ) |
| 13 | 1.000E+04 | allowable for stress in material 3: STRM3A(1 ,2 ) |
| 14 | 1.000E+04 | allowable for stress in material 3: STRM3A(1 ,3 ) |
| 15 | 1.000E+04 | allowable for stress in material 3: STRM3A(1 ,4 ) |
| 16 | 1.000E+04 | allowable for stress in material 3: STRM3A(1 ,5 ) |

PARAMETERS WHICH ARE FACTORS OF SAFETY

| VAR. NO. | CURRENT VALUE | DEFINITION |
|---|---|---|
| **Item 14:** | | |
| **1** | **3.000E+00** | **general buckling factor of safety: GENBUKF(1 )** |
| 2 | 1.000E+00 | factor of safety for stress in material 1: STRM1F(1 ,1 ) |
| 3 | 1.000E+00 | factor of safety for stress in material 1: STRM1F(1 ,2 ) |
| 4 | 1.000E+00 | factor of safety for stress in material 1: STRM1F(1 ,3 ) |
| 5 | 1.000E+00 | factor of safety for stress in material 1: STRM1F(1 ,4 ) |
| 6 | 1.000E+00 | factor of safety for stress in material 1: STRM1F(1 ,5 ) |
| 7 | 1.000E+00 | factor of safety for stress in material 2: STRM2F(1 ,1 ) |
| 8 | 1.000E+00 | factor of safety for stress in material 2: STRM2F(1 ,2 ) |
| 9 | 1.000E+00 | factor of safety for stress in material 2: STRM2F(1 ,3 ) |
| 10 | 1.000E+00 | factor of safety for stress in material 2: STRM2F(1 ,4 ) |
| 11 | 1.000E+00 | factor of safety for stress in material 2: STRM2F(1 ,5 ) |
| 12 | 1.000E+00 | factor of safety for stress in material 3: STRM3F(1 ,1 ) |
| 13 | 1.000E+00 | factor of safety for stress in material 3: STRM3F(1 ,2 ) |
| 14 | 1.000E+00 | factor of safety for stress in material 3: STRM3F(1 ,3 ) |
| 15 | 1.000E+00 | factor of safety for stress in material 3: STRM3F(1 ,4 ) |
| 16 | 1.000E+00 | factor of safety for stress in material 3: STRM3F(1 ,5 ) |

```
   0 INEQUALITY CONSTRAINTS WHICH MUST BE SATISFIED



 DESCRIPTION OF FILES USED AND GENERATED IN THIS RUN:

 try4.NAM = This file contains only the name of the case.
 try4.OPM = Output data. Please list this file and inspect
            carefully before proceeding.
 try4.OPP = Output file containing evolution of design and
            margins since the beginning of optimization cycles.
 try4.CBL = Labelled common blocks for analysis.
            (This is an unformatted sequential file.)
 try4.OPT = This file contains the input data for MAINSETUP
             as well as OPTIMIZE. The batch command OPTIMIZE
             can be given over and over again without having
             to return to MAINSETUP because try4.OPT exists.
 URPROMPT.DAT= Prompt file for interactive input.

 For further information about files used and generated
 during operation of GENOPT, give the command HELPG FILES.

 Menu of commands: CHOOSEPLOT, OPTIMIZE, MAINSETUP, CHANGE,
                   DECIDE, SUPEROPT

 IN ORDER TO AVOID FALSE CONVERGENCE OF THE DESIGN, BE SURE TO
 RUN  "OPTIMIZE" MANY TIMES DURING AN OPTIMIZATION AND/OR USE
 THE "GLOBAL" OPTIMIZING SCRIPT, "SUPEROPT".
 **** NOTE: It is almost always best to set the number of ****
 **** iterations per execution of "OPTIMIZE" equal to 5   ****
 **** in response to the following prompt in "MAINSETUP": ****
 **** "How many design iterations in this run (3 to 25)?" ****
 **** Hence, the *.OPT file should almost always have the ****
 **** following line in it:                               ****
 **** "5   $ How many design iterations in this run (3 to 25)?"
 ******************** END OF try4.OPM FILE ****************
================================================================
```

**Table 12 Input file, try4.CHG, for the GENOPT processor called "CHANGE". CHANGE is used to save the optimized design obtained after the first execution of SUPEROPT.**

```
==============================================================================
      n         $ Do you want a tutorial session and tutorial output?
      y         $ Do you want to change any values in Parameter Set No. 1?
         1      $ Number of parameter to change (1, 2, 3, . .)
   58.7500      $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         2      $ Number of parameter to change (1, 2, 3, . .)
    8.00000     $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         3      $ Number of parameter to change (1, 2, 3, . .)
   15.0000      $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         4      $ Number of parameter to change (1, 2, 3, . .)
  0.048731 $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         5      $ Number of parameter to change (1, 2, 3, . .)
  0.080927 $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         6      $ Number of parameter to change (1, 2, 3, . .)
  0.138530 $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         7      $ Number of parameter to change (1, 2, 3, . .)
  0.030000 $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         8      $ Number of parameter to change (1, 2, 3, . .)
  0.047662 $ New value of the parameter
      n         $ Want to change any other parameters in this set?
      n         $ Do you want to change values of any "fixed" parameters?
      n         $ Do you want to change any loads?
      n         $ Do you want to change values of allowables?
      n         $ Do you want to change any factors of safety?
==============================================================================
```

**Table 13 Input file, try4.DEC, for the GENOPT processor called "DECIDE" in preparation for the second execution of SUPEROPT**

```
==============================================================================
      n         $ Do you want a tutorial session and tutorial output?
         1      $ Choose a decision variable (1,2,3,...)
   50.00000     $ Lower bound of variable no.(  1)
   60.00000     $ Upper bound of variable no.(  1)
      y         $ Do you want especially to restrict variable no.(  1)
   0.500000     $ Maximum permitted change in variable no.(  1)
      y         $ Any more decision variables (Y or N) ?
         2      $ Choose a decision variable (1,2,3,...)
   6.300000     $ Lower bound of variable no.(  2)
        20      $ Upper bound of variable no.(  2)
      y         $ Do you want especially to restrict variable no.(  2)
  0.2000000     $ Maximum permitted change in variable no.(  2)
      y         $ Any more decision variables (Y or N) ?
         3      $ Choose a decision variable (1,2,3,...)
   10.00000     $ Lower bound of variable no.(  3)
        30      $ Upper bound of variable no.(  3)
      y         $ Do you want especially to restrict variable no.(  3)
  0.2000000     $ Maximum permitted change in variable no.(  3)
      y         $ Any more decision variables (Y or N) ?
         4      $ Choose a decision variable (1,2,3,...)
  0.3000000E-01 $ Lower bound of variable no.(  4)
  0.3000000     $ Upper bound of variable no.(  4)
      n         $ Do you want especially to restrict variable no.(  4)
      y         $ Any more decision variables (Y or N) ?
         5      $ Choose a decision variable (1,2,3,...)
  0.3000000E-01 $ Lower bound of variable no.(  5)
  0.3000000     $ Upper bound of variable no.(  5)
      n         $ Do you want especially to restrict variable no.(  5)
      y         $ Any more decision variables (Y or N) ?
         6      $ Choose a decision variable (1,2,3,...)
  0.3000000E-01 $ Lower bound of variable no.(  6)
  0.3000000     $ Upper bound of variable no.(  6)
      n         $ Do you want especially to restrict variable no.(  6)
      y         $ Any more decision variables (Y or N) ?
         7      $ Choose a decision variable (1,2,3,...)
  0.2000000E-01 $ Lower bound of variable no.(  7)
  0.3000000     $ Upper bound of variable no.(  7)
      n         $ Do you want especially to restrict variable no.(  7)
      y         $ Any more decision variables (Y or N) ?
         8      $ Choose a decision variable (1,2,3,...)
  0.3000000E-01 $ Lower bound of variable no.(  8)
```

```
  0.3000000      $ Upper bound of variable no.(  8)
      n          $ Do you want especially to restrict variable no.(  8)
      n          $ Any more decision variables (Y or N) ?
      n          $ Any linked variables (Y or N) ?
      n          $ Any inequality relations among variables? (type H)
      y          $ Any escape variables (Y or N) ?
      y          $ Want to have escape variables chosen by default?
=====================================================================
```

## Table 14 Output file, try4.OPM, for the optimized design produced by the second execution of SUPEROPT

```
================================================================================
      n         $ Do you want a tutorial session and tutorial output?
         0      $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         2      $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
         2      $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
         5      $ How many design iterations in this run (3 to 25)?
      n         $ Take "shortcuts" for perturbed designs (Y or N)?
         1      $ Choose 1 or 2 or 3 or 4 or 5 for IDESIGN
         1      $ Choose 1 or 2 or 3 or 4 or 5 for move limits, IMOVE
      y         $ Do you want default (RATIO=10) for initial move limit jump?
      y         $ Do you want the default perturbation (dx/x = 0.05)?
      n         $ Do you want to have dx/x modified by GENOPT?
      n         $ Do you want to reset total iterations to zero (Type H)?
         1      $ Choose IAUTOF= 1 or 2 or 3 or 4 or 5 or 6 to change X(i)

 ************ END OF THE  try4.OPT  FILE ***********
 ************ AUGUST, 2010 VERSION OF GENOPT ****************
 ******** BEGINNING OF THE  try4.OPM  FILE *********

 *********************** MAIN PROCESSOR **********************
 The purpose of the mainprocessor, OPTIMIZE, is to perform,
 in a batch mode, the work specified by MAINSETUP for the case
 called try4.  Results are stored in the file  try4.OPM.
 Please inspect try4.OPM before doing more design iterations.
 ************************************************************

 STRUCTURAL ANALYSIS FOR DESIGN ITERATION NO.   0:

  STRUCTURAL ANALYSIS WITH UNPERTURBED DECISION VARIABLES
 VAR. DEC. ESCAPE LINK. LINKED  LINKING   LOWER     CURRENT    UPPER      DEFINITION
  NO. VAR.  VAR.  VAR.    TO   CONSTANT   BOUND      VALUE     BOUND
   1   Y     N     N      0    0.00E+00  4.00E+01 5.0000E+01  5.00E+01  height from inner to outer
membranes: HEIGHT
   2   Y     N     N      0    0.00E+00  6.30E+00 6.3210E+00  8.00E+00  radius of curvature of
inner membrane: RINNER
   3   Y     N     N      0    0.00E+00  9.10E+00 1.0580E+01  1.10E+01  radius of curvature of
outer membrane: ROUTER
   4   Y     Y     N      0    0.00E+00  3.00E-02 5.7210E-02  3.00E-01  thickness of the inner
curved membrane: TINNER
   5   Y     Y     N      0    0.00E+00  3.00E-02 6.2790E-02  3.00E-01  thickness of the outer
curved membrane: TOUTER
   6   Y     Y     N      0    0.00E+00  3.00E-02 1.2250E-01  3.00E-01  thickness of inner truss-
core segment: TFINNR
   7   Y     Y     N      0    0.00E+00  3.00E-02 5.7920E-02  3.00E-01  thickness of the outer
truss segment: TFOUTR
   8   Y     Y     N      0    0.00E+00  3.00E-02 4.5620E-02  3.00E-01  thickness of each truss-
core web: TFWEBS

  BEHAVIOR FOR  1 ENVIRONMENT (LOAD SET)
```

```
 CONSTRAINT  BEHAVIOR              DEFINITION
   NUMBER     VALUE

  BEHAVIOR FOR LOAD SET NUMBER, ILOADX= 1


 BIGBOSOR4 input file for: general buckling load
try4.BEHX1

 Changes in temperature required to create 2 total axial loads:

  1. Change in temperature required to create the axial thermal
     strain that generates the axial tension due to closing the
     two ends of the pressurized volume (PMIDDL=  6.0000E+01)
     between the inner and outer walls of the balloon in
     Load Step No. 1:                                DELTAT= -9.9623E+01

  2. Change in temperature required to simulate the Poisson
     axial expansion caused by the application of the outer
     pressure, POUTER =  5.0000E+00 in Load Step No. 2:  DELT=  0.0000E+00


 GENERAL BUCKLING LOAD FACTORS AND MODES (BEHX2)
        2.9742E+00(     1)
 Critical buckling load factor, GENBUK=  2.9742E+00
 Critical number of axial half-waves, NWVCRT=    1

 Differences in the resultants along the axis of the prismatic
 balloon for each segment, J, of the first module:
 [N2VAR(J) for the total load] - [N2FIX(J) for the fixed load]=
 N2DIFF(J),J=1,6)= -6.5655E+01 -1.9968E+01 -1.8378E+02 -1.1819E+00 -
1.0860E+01 -1.0826E+01
 N2VAR(J) (total load) are the resultants from Load Step No. 2.
 N2FIX(J) (fixed load) are the resultants from Load Step No. 1.
 NOTE: The stresses used as behavioral constraints are
       computed from N2VAR(J)/thickness(J). These stresses are
       lower than those computed from N2FIX(J)/thickness(J).

 PREBUCKLING STRESS RESULTANTS IN THE FIRST MODULE
          "fixed" from Load Step No. 1 total from Load Step No. 2
 Seg.J  Node I  N1FIX(I,J)   N2FIX(I,J)   N1VAR(I,J)   N2VAR(I,J)
    1      1   7.9503E+02   4.8957E+02   5.7618E+02   4.2391E+02
    2      1   6.6087E+02   4.7043E+02   5.9431E+02   4.5046E+02
    3      1   1.8187E+03   1.0766E+03   1.2060E+03   8.9280E+02
    4      1   3.8914E+02   3.6472E+02   3.8520E+02   3.6354E+02
    5      1   4.9059E+02   3.4492E+02   4.5439E+02   3.3406E+02
    6      1   4.9013E+02   3.4478E+02   4.5405E+02   3.3396E+02
```

```
PREBUCKLING MEMBRANE STRESSES IN THE FIRST MODULE COMPUTED FROM
N1FIX/thickness, N2FIX/thickness, N1VAR/thickness, N2VAR/thickness:
          "fixed" from Load Step No. 1 total from Load Step No. 2
Seg.J  Node I STRS1F(I,J)  STRS2F(I,J)  STRS1V(I,J)  STRS2V(I,J)
   1      1   1.3726E+04   8.4525E+03   9.9479E+03   7.3190E+03
   2      1   1.0525E+04   7.4921E+03   9.4651E+03   7.1741E+03
   3      1   1.4846E+04   8.7885E+03   9.8453E+03   7.2882E+03
   4      1   6.8019E+03   6.3752E+03   6.7330E+03   6.3545E+03
   5      1   1.0754E+04   7.5607E+03   9.9602E+03   7.3227E+03

Behavior number, General buckling load factor:

Newton iterations required to solve the nonlinear
axisymmetric pre-buckling equilibrium state for the
"fixed" loads (PINNER, PMIDDL, DELTAT):       ITER=  5
Newton iterations required to solve the nonlinear
axisymmetric pre-buckling equilibrium state for the
total loads (PINNER, PMIDDL, DELTAT, POUTER): ITER=  2

   1        2.974184      general buckling load factor: GENBUK(1 )

 BEHAVIOR OVER J =  stress component number
   2        9465.080      stress component in material 1: STRM1(1 ,1 )
   3       0.1000000E-09  stress component in material 1: STRM1(1 ,2 )
   4        7174.125      stress component in material 1: STRM1(1 ,3 )
   5       0.1000000E-09  stress component in material 1: STRM1(1 ,4 )
   6       0.1000000E-09  stress component in material 1: STRM1(1 ,5 )

 BEHAVIOR OVER J =  stress component number
   7        9947.891      stress component in material 2: STRM2(1 ,1 )
   8       0.1000000E-09  stress component in material 2: STRM2(1 ,2 )
   9        7318.969      stress component in material 2: STRM2(1 ,3 )
  10       0.1000000E-09  stress component in material 2: STRM2(1 ,4 )
  11       0.1000000E-09  stress component in material 2: STRM2(1 ,5 )

 BEHAVIOR OVER J =  stress component number
  12        9960.242      stress component in material 3: STRM3(1 ,1 )
  13       0.1000000E-09  stress component in material 3: STRM3(1 ,2 )
  14        7322.674      stress component in material 3: STRM3(1 ,3 )
  15       0.1000000E-09  stress component in material 3: STRM3(1 ,4 )
  16       0.1000000E-09  stress component in material 3: STRM3(1 ,5 )


***** RESULTS FOR LOAD SET NO.  1  ******
PARAMETERS WHICH DESCRIBE BEHAVIOR (e.g. stress, buckling load)

BEH.   CURRENT
NO.     VALUE            DEFINITION
```

```
  1    2.974E+00    general buckling load factor: GENBUK(1 )
  2    9.465E+03    stress component in material 1: STRM1(1 ,1 )
  3    1.000E-10    stress component in material 1: STRM1(1 ,2 )
  4    7.174E+03    stress component in material 1: STRM1(1 ,3 )
  5    1.000E-10    stress component in material 1: STRM1(1 ,4 )
  6    1.000E-10    stress component in material 1: STRM1(1 ,5 )
  7    9.948E+03    stress component in material 2: STRM2(1 ,1 )
  8    1.000E-10    stress component in material 2: STRM2(1 ,2 )
  9    7.319E+03    stress component in material 2: STRM2(1 ,3 )
 10    1.000E-10    stress component in material 2: STRM2(1 ,4 )
 11    1.000E-10    stress component in material 2: STRM2(1 ,5 )
 12    9.960E+03    stress component in material 3: STRM3(1 ,1 )
 13    1.000E-10    stress component in material 3: STRM3(1 ,2 )
 14    7.323E+03    stress component in material 3: STRM3(1 ,3 )
 15    1.000E-10    stress component in material 3: STRM3(1 ,4 )
 16    1.000E-10    stress component in material 3: STRM3(1 ,5 )

 ******* NOTE ******* NOTE ******* NOTE ****** NOTE ******
 The phrase, "NOT APPLY", for MARGIN VALUE means that that
 particular margin value is exactly zero.
 *** END NOTE *** END NOTE *** END NOTE *** END NOTE *****

 ***** RESULTS FOR LOAD SET NO.  1  ******
 MARGINS CORRESPONDING TO CURRENT DESIGN (F.S.= FACTOR OF SAFETY)

 MARGIN CURRENT
 NO.     VALUE           DEFINITION
  1   -8.605E-03  (GENBUK(1 )/GENBUKA(1 )) / GENBUKF(1 )-1; F.S.=  3.00
  2    5.652E-02  (STRM1A(1 ,1 )/STRM1(1 ,1 )) / STRM1F(1 ,1 )-1; F.S.=  1.00
  3    3.939E-01  (STRM1A(1 ,3 )/STRM1(1 ,3 )) / STRM1F(1 ,3 )-1; F.S.=  1.00
  4    5.238E-03  (STRM2A(1 ,1 )/STRM2(1 ,1 )) / STRM2F(1 ,1 )-1; F.S.=  1.00
  5    3.663E-01  (STRM2A(1 ,3 )/STRM2(1 ,3 )) / STRM2F(1 ,3 )-1; F.S.=  1.00
  6    3.992E-03  (STRM3A(1 ,1 )/STRM3(1 ,1 )) / STRM3F(1 ,1 )-1; F.S.=  1.00
  7    3.656E-01  (STRM3A(1 ,3 )/STRM3(1 ,3 )) / STRM3F(1 ,3 )-1; F.S.=  1.00

******************* DESIGN OBJECTIVE *******************
 *************                             ***********
    CURRENT VALUE OF THE OBJECTIVE FUNCTION:

 VAR.   CURRENT
 NO.     VALUE              DEFINITION
  1    5.738E+01  weight/length of the balloon: WEIGHT
 *************                          ************
 ******************* DESIGN OBJECTIVE *******************
***********  ALL 1 LOAD CASES PROCESSED ***********
===============================================================
```

**Table 15 Input file, try4.CHG, for the GENOPT processor called "CHANGE". CHANGE is used to save the optimized design obtained after the second execution of SUPEROPT**

```
=========================================================================
      n         $ Do you want a tutorial session and tutorial output?
      y         $ Do you want to change any values in Parameter Set No. 1?
         1      $ Number of parameter to change (1, 2, 3, . .)
   50.0000      $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         2      $ Number of parameter to change (1, 2, 3, . .)
    6.32100     $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         3      $ Number of parameter to change (1, 2, 3, . .)
   10.5800      $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         4      $ Number of parameter to change (1, 2, 3, . .)
  0.057210 $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         5      $ Number of parameter to change (1, 2, 3, . .)
  0.062790 $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         6      $ Number of parameter to change (1, 2, 3, . .)
  0.122500 $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         7      $ Number of parameter to change (1, 2, 3, . .)
  0.057920 $ New value of the parameter
      y         $ Want to change any other parameters in this set?
         8      $ Number of parameter to change (1, 2, 3, . .)
  0.045620 $ New value of the parameter
      n         $ Want to change any other parameters in this set?
      n         $ Do you want to change values of any "fixed" parameters?
      n         $ Do you want to change any loads?
      n         $ Do you want to change values of allowables?
      n         $ Do you want to change any factors of safety?
=========================================================================
```

## Table 16 Input file, try4.DEC, for the GENOPT processor called "DECIDE" in preparation for the third execution of SUPEROPT

```
=====================================================================
     n         $ Do you want a tutorial session and tutorial output?
        1      $ Choose a decision variable (1,2,3,...)
   40.00000    $ Lower bound of variable no.(  1)
   50.00000    $ Upper bound of variable no.(  1)
     y         $ Do you want especially to restrict variable no.(  1)
   0.500000    $ Maximum permitted change in variable no.(  1)
     y         $ Any more decision variables (Y or N) ?
        2      $ Choose a decision variable (1,2,3,...)
   6.300000    $ Lower bound of variable no.(  2)
      8.0      $ Upper bound of variable no.(  2)
     y         $ Do you want especially to restrict variable no.(  2)
  0.2000000    $ Maximum permitted change in variable no.(  2)
     y         $ Any more decision variables (Y or N) ?
        3      $ Choose a decision variable (1,2,3,...)
    9.10000    $ Lower bound of variable no.(  3)
     11.0      $ Upper bound of variable no.(  3)
     y         $ Do you want especially to restrict variable no.(  3)
  0.2000000    $ Maximum permitted change in variable no.(  3)
     y         $ Any more decision variables (Y or N) ?
        4      $ Choose a decision variable (1,2,3,...)
  0.3000000E-01 $ Lower bound of variable no.(  4)
  0.3000000    $ Upper bound of variable no.(  4)
     n         $ Do you want especially to restrict variable no.(  4)
     y         $ Any more decision variables (Y or N) ?
        5      $ Choose a decision variable (1,2,3,...)
  0.3000000E-01 $ Lower bound of variable no.(  5)
  0.3000000    $ Upper bound of variable no.(  5)
     n         $ Do you want especially to restrict variable no.(  5)
     y         $ Any more decision variables (Y or N) ?
        6      $ Choose a decision variable (1,2,3,...)
  0.3000000E-01 $ Lower bound of variable no.(  6)
  0.3000000    $ Upper bound of variable no.(  6)
     n         $ Do you want especially to restrict variable no.(  6)
     y         $ Any more decision variables (Y or N) ?
        7      $ Choose a decision variable (1,2,3,...)
  0.3000000E-01 $ Lower bound of variable no.(  7)
  0.3000000    $ Upper bound of variable no.(  7)
     n         $ Do you want especially to restrict variable no.(  7)
     y         $ Any more decision variables (Y or N) ?
        8      $ Choose a decision variable (1,2,3,...)
  0.3000000E-01 $ Lower bound of variable no.(  8)
```

```
  0.3000000      $ Upper bound of variable no.(  8)
      n          $ Do you want especially to restrict variable no.(  8)
      n          $ Any more decision variables (Y or N) ?
      n          $ Any linked variables (Y or N) ?
      n          $ Any inequality relations among variables? (type H)
      y          $ Any escape variables (Y or N) ?
      y          $ Want to have escape variables chosen by default?
=======================================================================
```

## Table 17 Output file, try4.OPM, from the GENOPT processor, OPTIMIZE, for the final optimum design with the outer pressure, POUTER, set equal to 15 psi and the factor of safety for buckling set equal to 1.0

```
================================================================================
      n         $ Do you want a tutorial session and tutorial output?
         0      $ Choose an analysis you DON'T want (1, 2,..), IBEHAV
         2      $ NPRINT= output index (0=GOOD, 1=ok, 2=debug, 3=too much)
         2      $ Choose type of analysis (1=opt., 2=fixed, 3=sensit.) ITYPE
         5      $ How many design iterations in this run (3 to 25)?
      n         $ Take "shortcuts" for perturbed designs (Y or N)?
         1      $ Choose 1 or 2 or 3 or 4 or 5 for IDESIGN
         1      $ Choose 1 or 2 or 3 or 4 or 5 for move limits, IMOVE
      y         $ Do you want default (RATIO=10) for initial move limit jump?
      y         $ Do you want the default perturbation (dx/x = 0.05)?
      n         $ Do you want to have dx/x modified by GENOPT?
      n         $ Do you want to reset total iterations to zero (Type H)?
         1      $ Choose IAUTOF= 1 or 2 or 3 or 4 or 5 or 6 to change X(i)

 ************ END OF THE  try4.OPT  FILE ***********
 ************ AUGUST, 2010 VERSION OF GENOPT ***************
 ******** BEGINNING OF THE  try4.OPM  FILE *********


 *********************** MAIN PROCESSOR **********************
 The purpose of the mainprocessor, OPTIMIZE, is to perform,
 in a batch mode, the work specified by MAINSETUP for the case
 called try4.  Results are stored in the file  try4.OPM.
 Please inspect try4.OPM before doing more design iterations.
 ************************************************************

 STRUCTURAL ANALYSIS FOR DESIGN ITERATION NO.   0:

  STRUCTURAL ANALYSIS WITH UNPERTURBED DECISION VARIABLES
 VAR. DEC. ESCAPE LINK. LINKED  LINKING    LOWER     CURRENT    UPPER       DEFINITION
  NO. VAR.  VAR.  VAR.    TO    CONSTANT   BOUND      VALUE     BOUND
   1   Y     N     N      0    0.00E+00  4.00E+01 5.0000E+01  5.00E+01  height from inner to outer
membranes: HEIGHT
   2   Y     N     N      0    0.00E+00  6.30E+00 6.3210E+00  8.00E+00  radius of curvature of
inner membrane: RINNER
   3   Y     N     N      0    0.00E+00  9.10E+00 1.0580E+01  1.10E+01  radius of curvature of
outer membrane: ROUTER
   4   Y     Y     N      0    0.00E+00  3.00E-02 5.7210E-02  3.00E-01  thickness of the inner
curved membrane: TINNER
   5   Y     Y     N      0    0.00E+00  3.00E-02 6.2790E-02  3.00E-01  thickness of the outer
curved membrane: TOUTER
   6   Y     Y     N      0    0.00E+00  3.00E-02 1.2250E-01  3.00E-01  thickness of inner truss-
core segment: TFINNR
```

```
   7   Y     Y     N      0    0.00E+00  3.00E-02 5.7920E-02  3.00E-01  thickness of the outer
truss segment: TFOUTR
   8   Y     Y     N      0    0.00E+00  3.00E-02 4.5620E-02  3.00E-01  thickness of each truss-
core web: TFWEBS
```

```
  BEHAVIOR FOR  1 ENVIRONMENT (LOAD SET)

 CONSTRAINT  BEHAVIOR              DEFINITION
   NUMBER     VALUE

  BEHAVIOR FOR LOAD SET NUMBER, ILOADX= 1


 BIGBOSOR4 input file for: general buckling load
try4.BEHX1

 Changes in temperature required to create 2 total axial loads:

  1. Change in temperature required to create the axial thermal
     strain that generates the axial tension due to closing the
     two ends of the pressurized volume (PMIDDL=  6.0000E+01)
     between the inner and outer walls of the balloon in
     Load Step No. 1:                                DELTAT= -9.9623E+01

  2. Change in temperature required to simulate the Poisson
     axial expansion caused by the application of the outer
     pressure, POUTER =  1.5000E+01 in Load Step No. 2:  DELT=  0.0000E+00



 GENERAL BUCKLING LOAD FACTORS AND MODES (BEHX2)
        1.0043E+00(     1)
 Critical buckling load factor, GENBUK=  1.0043E+00
 Critical number of axial half-waves, NWVCRT=    1

 Differences in the resultants along the axis of the prismatic
 balloon for each segment, J, of the first module:
 [N2VAR(J) for the total load] - [N2FIX(J) for the fixed load]=
 N2DIFF(J),J=1,6)= -1.9402E+02 -5.7560E+01 -5.4458E+02 -3.4890E+00
-3.2172E+01 -3.2069E+01
 N2VAR(J) (total load) are the resultants from Load Step No. 2.
 N2FIX(J) (fixed load) are the resultants from Load Step No. 1.
 NOTE: The stresses used as behavioral constraints are
       computed from N2VAR(J)/thickness(J). These stresses are
       lower than those computed from N2FIX(J)/thickness(J).

 PREBUCKLING STRESS RESULTANTS IN THE FIRST MODULE
          "fixed" from Load Step No. 1 total from Load Step No. 2
 Seg.J  Node I  N1FIX(I,J)   N2FIX(I,J)   N1VAR(I,J)   N2VAR(I,J)
   1      1   7.9503E+02   4.8957E+02   1.4830E+02   2.9555E+02
```

```
   2      1   6.6087E+02   4.7043E+02   4.6900E+02   4.1287E+02
   3      1   1.8187E+03   1.0766E+03   3.3898E+00   5.3201E+02
   4      1   3.8914E+02   3.6472E+02   3.7751E+02   3.6123E+02
   5      1   4.9059E+02   3.4492E+02   3.8335E+02   3.1275E+02
   6      1   4.9013E+02   3.4478E+02   3.8324E+02   3.1272E+02

 PREBUCKLING MEMBRANE STRESSES IN THE FIRST MODULE COMPUTED FROM
 N1FIX/thickness, N2FIX/thickness, N1VAR/thickness, N2VAR/thickness:
           "fixed" from Load Step No. 1 total from Load Step No. 2
 Seg.J  Node I STRS1F(I,J)  STRS2F(I,J)  STRS1V(I,J)  STRS2V(I,J)
   1      1   1.3726E+04   8.4525E+03   2.5605E+03   5.1027E+03
   2      1   1.0525E+04   7.4921E+03   7.4694E+03   6.5754E+03
   3      1   1.4846E+04   8.7885E+03   2.7672E+01   4.3429E+03
   4      1   6.8019E+03   6.3752E+03   6.5986E+03   6.3142E+03
   5      1   1.0754E+04   7.5607E+03   8.4031E+03   6.8555E+03

 Behavior number, General buckling load factor:

 Newton iterations required to solve the nonlinear
 axisymmetric pre-buckling equilibrium state for the
 "fixed" loads (PINNER, PMIDDL, DELTAT):       ITER=  5
 Newton iterations required to solve the nonlinear
 axisymmetric pre-buckling equilibrium state for the
 total loads (PINNER, PMIDDL, DELTAT, POUTER): ITER=  4

   1        1.004345      general buckling load factor: GENBUK(1 )

  BEHAVIOR OVER J =  stress component number


 BIGBOSOR4 input file for: stress components in materials 1,2,3
try4.BEHX2

 Maximum stress components in the entire structure at the last load step (from
BIGBOSOR4):
  1  1.0811E+00 effect. stress: matl=1 ,  A , seg=26, node=32, layer=1 ,z= 0.03
  2  2.2629E+00 effect. stress: matl=2 ,  A , seg=1 , node=33, layer=1 ,z= 0.03
  3  1.2587E+00 effect. stress: matl=3 ,  A , seg=66, node=33, layer=1 ,z=-0.02

 FIVE STRESS COMPONENTS (including bending) FOR MATL i, STRCi(ILOADX,J), J=1,5:
    fiber tension  fiber compres  transv tension transv compres in-plane shear
  or effect.stress
 Material 1 stress: STRC1(ILOADX,J),J=1,5)=
     9.2497E+03     0.0000E+00     0.0000E+00     0.0000E+00     0.0000E+00
 Material 2 stress: STRC2(ILOADX,J),J=1,5)=
     4.4191E+03     0.0000E+00     0.0000E+00     0.0000E+00     0.0000E+00
 Material 3 stress: STRC3(ILOADX,J),J=1,5)=
     7.9447E+03     0.0000E+00     0.0000E+00     0.0000E+00     0.0000E+00
```

```
Changes in temperature required to create 2 total axial loads:

 1. Change in temperature required to create the axial thermal
    strain that generates the axial tension due to closing the
    two ends of the pressurized volume (PMIDDL=  6.0000E+01)
    between the inner and outer walls of the balloon in
    Load Step No. 1:                                DELTAT= -9.9623E+01

 2. Change in temperature required to simulate the Poisson
    axial expansion caused by the application of the outer
    pressure, POUTER =  1.5000E+01 in Load Step No. 2:  DELT=  2.9784E+01


BEHAVIOR OVER J =  stress component number
   2        7469.398      stress component in material 1: STRM1(1 ,1 )
   3       0.1000000E-09  stress component in material 1: STRM1(1 ,2 )
   4        6575.421      stress component in material 1: STRM1(1 ,3 )
   5       0.1000000E-09  stress component in material 1: STRM1(1 ,4 )
   6       0.1000000E-09  stress component in material 1: STRM1(1 ,5 )

 BEHAVIOR OVER J =  stress component number
   7        2560.475      stress component in material 2: STRM2(1 ,1 )
   8       0.1000000E-09  stress component in material 2: STRM2(1 ,2 )
   9        5102.743      stress component in material 2: STRM2(1 ,3 )
  10       0.1000000E-09  stress component in material 2: STRM2(1 ,4 )
  11       0.1000000E-09  stress component in material 2: STRM2(1 ,5 )

 BEHAVIOR OVER J =  stress component number
  12        8403.080      stress component in material 3: STRM3(1 ,1 )
  13       0.1000000E-09  stress component in material 3: STRM3(1 ,2 )
  14        6855.525      stress component in material 3: STRM3(1 ,3 )
  15       0.1000000E-09  stress component in material 3: STRM3(1 ,4 )
  16       0.1000000E-09  stress component in material 3: STRM3(1 ,5 )

***** RESULTS FOR LOAD SET NO.  1  ******
PARAMETERS WHICH DESCRIBE BEHAVIOR (e.g. stress, buckling load)

BEH.   CURRENT
NO.     VALUE            DEFINITION
 1    1.004E+00    general buckling load factor: GENBUK(1 )
 2    7.469E+03    stress component in material 1: STRM1(1 ,1 )
 3    1.000E-10    stress component in material 1: STRM1(1 ,2 )
 4    6.575E+03    stress component in material 1: STRM1(1 ,3 )
 5    1.000E-10    stress component in material 1: STRM1(1 ,4 )
 6    1.000E-10    stress component in material 1: STRM1(1 ,5 )
 7    2.560E+03    stress component in material 2: STRM2(1 ,1 )
 8    1.000E-10    stress component in material 2: STRM2(1 ,2 )
 9    5.103E+03    stress component in material 2: STRM2(1 ,3 )
```

```
 10    1.000E-10    stress component in material 2: STRM2(1 ,4 )
 11    1.000E-10    stress component in material 2: STRM2(1 ,5 )
 12    8.403E+03    stress component in material 3: STRM3(1 ,1 )
 13    1.000E-10    stress component in material 3: STRM3(1 ,2 )
 14    6.856E+03    stress component in material 3: STRM3(1 ,3 )
 15    1.000E-10    stress component in material 3: STRM3(1 ,4 )
 16    1.000E-10    stress component in material 3: STRM3(1 ,5 )

 ******* NOTE ******* NOTE ******* NOTE ****** NOTE ******
 The phrase, "NOT APPLY", for MARGIN VALUE means that that
 particular margin value is exactly zero.
 *** END NOTE *** END NOTE *** END NOTE *** END NOTE *****


 ***** RESULTS FOR LOAD SET NO.  1  ******
 MARGINS CORRESPONDING TO CURRENT DESIGN (F.S.= FACTOR OF SAFETY)

 MARGIN CURRENT
 NO.     VALUE           DEFINITION
  1    4.345E-03  (GENBUK(1 )/GENBUKA(1 )) / GENBUKF(1 )-1; F.S.=  1.00
  2    3.388E-01  (STRM1A(1 ,1 )/STRM1(1 ,1 )) / STRM1F(1 ,1 )-1; F.S.=  1.00
  3    5.208E-01  (STRM1A(1 ,3 )/STRM1(1 ,3 )) / STRM1F(1 ,3 )-1; F.S.=  1.00
  4    2.906E+00  (STRM2A(1 ,1 )/STRM2(1 ,1 )) / STRM2F(1 ,1 )-1; F.S.=  1.00
  5    9.597E-01  (STRM2A(1 ,3 )/STRM2(1 ,3 )) / STRM2F(1 ,3 )-1; F.S.=  1.00
  6    1.900E-01  (STRM3A(1 ,1 )/STRM3(1 ,1 )) / STRM3F(1 ,1 )-1; F.S.=  1.00
  7    4.587E-01  (STRM3A(1 ,3 )/STRM3(1 ,3 )) / STRM3F(1 ,3 )-1; F.S.=  1.00

 ***************************************************
 ******************** DESIGN OBJECTIVE *******************
 *************                            ***********
    CURRENT VALUE OF THE OBJECTIVE FUNCTION:

 VAR.   CURRENT
 NO.     VALUE             DEFINITION
  1    5.738E+01  weight/length of the balloon: WEIGHT

******************** DESIGN OBJECTIVE *******************
***********  ALL 1 LOAD CASES PROCESSED ***********
==================================================================
```

# Table 18 Output from BIGBOSOR4 corresponding to the axisymmetric nonlinear prebuckling analysis of the final optimized configuration with 15 modules and truss-like (slanted) webs subjected to Load Steps 1 and 2

===============================================================================

**Output contained in the greatly abridged file, try4.OUT, after execution of BIGBOSOR4 independently of the GENOPT system**

Axisymmetric equilibrium from a nonlinear static analysis (INDIC=0)

OUTPUT FROM BIGBOSOR4 FOR **LOAD STEP NO. 1** (Load Set B only, that is, PINNER = 0 psi, PMIDDL = 60 psi, POUTER = 0 psi, DELTAT = -99.623 deg.):

```
 PRESSURE MULTIPLIER, P=   0.000000E+00, TEMPERATURE MULTIPLIER,TEMP =   1.000000E+00
 ANGULAR VELOCITY, OMEGA=   0.000000E+00

 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   0. START FACTORING AND SOLVING
 Factoring done for iteration   0; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  0
 ITERATION NO.  0 MAXIMUM DISPLACEMENT=  4.8040E+00
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   1. START FACTORING AND SOLVING
 Factoring done for iteration   1; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  1
 ITERATION NO.  1 MAXIMUM DISPLACEMENT=  4.4950E+00
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   2. START FACTORING AND SOLVING
 Factoring done for iteration   2; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  2
 ITERATION NO.  2 MAXIMUM DISPLACEMENT=  4.3692E+00
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   3. START FACTORING AND SOLVING
 Factoring done for iteration   3; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  3
 ITERATION NO.  3 MAXIMUM DISPLACEMENT=  4.3409E+00
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   4. START FACTORING AND SOLVING
 Factoring done for iteration   4; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  4
 ITERATION NO.  4 MAXIMUM DISPLACEMENT=  4.3338E+00
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   5. START FACTORING AND SOLVING
 Factoring done for iteration   5; Load step, ISTEP=   1
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  5
 ITERATION NO.  5 MAXIMUM DISPLACEMENT=  4.3338E+00
  NUMBER OF NEWTON-RAPHSON ITERATIONS REQUIRED FOR  CONVERGENCE = ITER =  5
```

OUTPUT FROM BIGBOSOR4 FOR **LOAD STEP NO. 2** (Load Sets B + A, that is, PINNER = 0 psi, PMIDDL = 60 psi, POUTER = 15 psi, DELTAT = -99.623 deg.):

```
 PRESSURE MULTIPLIER, P=   1.000000E+00, TEMPERATURE MULTIPLIER,TEMP =   1.000000E+00
```

```
 ANGULAR VELOCITY, OMEGA=   0.000000E+00

 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   0. START FACTORING AND SOLVING
 Factoring done for iteration   0; Load step, ISTEP=   2
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  0
 ITERATION NO.  0 MAXIMUM DISPLACEMENT=  5.7900E-01
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   1. START FACTORING AND SOLVING
 Factoring done for iteration   1; Load step, ISTEP=   2
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  1
 ITERATION NO.  1 MAXIMUM DISPLACEMENT=  5.6731E-01
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   2. START FACTORING AND SOLVING
 Factoring done for iteration   2; Load step, ISTEP=   2
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  2
 ITERATION NO.  2 MAXIMUM DISPLACEMENT=  5.7204E-01
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   3. START FACTORING AND SOLVING
 Factoring done for iteration   3; Load step, ISTEP=   2
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  3
 ITERATION NO.  3 MAXIMUM DISPLACEMENT=  5.7211E-01
 PRESTRESS STIFFNESS MATRIX CALCULATED FOR ITERATION NO.   4. START FACTORING AND SOLVING
 Factoring done for iteration   4; Load step, ISTEP=   2
 FACTORING AND SOLVING COMPLETED FOR PRESTRESS ITERATION NO.  4
 ITERATION NO.  4 MAXIMUM DISPLACEMENT=  5.7211E-01
  NUMBER OF NEWTON-RAPHSON ITERATIONS REQUIRED FOR  CONVERGENCE = ITER =  4
=====================================================================================
```