Table A16 List of the file, **struct.equivellipse.**
This is the **completed struct.new** library after the GENOPT user
has "**fleshed out**" the skeletal version of struct.new
for the generic case called "equivellipse".
“Fleshing out” SUBROUTINE STRUCT for this particular generic case
represents a major part of the work that led to generation of this report.
See Table a30 for a list of the file, “howto.struct”, which gives guidelines on how to “flesh out” SUBROUTINE STRUCT for a much simpler case called “cylinder”.

```
==================================================================
C=DECK      STRUCT
      SUBROUTINE STRUCT(IMODX,CONSTX,OBJGEN,CONMAX,NCONSX,IPOINC,
     1      PCWORD,CPLOTX,ILOADX,ISTARX,NUSERC,IBEHV,IDV,IFAST,
     1      JJJ1)
C
C  PURPOSE IS TO PERFORM THE ANALYSIS FOR A GIVEN DESIGN AND LOADING.
C  CONSTRAINT CONDITIONS ARE ALSO GENERATED.
C
      COMMON/PRMFIL/IFILEX,IFILE2,IOUT,IPRM(5)
      COMMON/PRMOUT/IFILE3,IFILE4,IFILE8,IFILE9,IFIL11
      COMMON/INDAT/INFILE
      COMMON/LWRUPR/VLBX(50),VUBX(50),CLINKX(50,5),VLINKX(50),VBVX(99)
      COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX
      COMMON/PARAMS/PARX(99),VARX(50),ALLOWX(99),CONSXX(99),DECX(50),
     1              ESCX(50)
      COMMON/WORDS1/WORDPX(99),WORDVX(50),WORDAX(99),WORDCC(99),
     1              WORDDX(50)
      COMMON/WORDS2/WORDLX(50),WORDEX(50),WORDIQ(20)
      COMMON/OPTVAR/IDVX(50),ILVX(50),IDLINK(50,5),IEVX(50),JTERMS(20)
      COMMON/NUMPR2/ILARX,ICARX,IOARX,IFLATX,NCASES,NPRINX
      COMMON/PARAM2/FLARX(50),CARX(99),OARX(50),FSAFEX(99),CPWRX(50,5)
      COMMON/PARAM3/CINEQX(15,20),DPWREQ(15,20)
      COMMON/PARAM4/IDINEQ(15,20),NINEQX,JINEQX(20),IEQTYP(20)
      COMMON/WORDS3/WORDFX(50),WORDBX(99),WORDOB(50),WORDSX(99)
      COMMON/WORDS4/WORDMX(99)
      COMMON/PWORD/PHRASE
      COMMON/PWORD2/IBLANK
      COMMON/ISKIPX/ISKIP(30)
      DIMENSION IBEHV(99)
C
C==================================================================
=
C  Start of first part of STRUCT written by "GENTEXT"
C  INSERT ADDITIONAL COMMON BLOCKS HERE: (THESE ARE "GENTEXT" VARIABLES)
      COMMON/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
```

```
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
C
      CHARACTER*80 PHRASE,CODPHR,PCWORD
      CHARACTER*80 WORDPX,WORDVX,WORDAX,WORDCX,WORDDX,WORDLX,WORDEX
      CHARACTER*80 WORDFX,WORDBX,WORDOB,WORDSX,WORDMX,WORDCC,WORDIQ
c     CHARACTER*4 ANSOUT,CHARAC,ANSWER
      CHARACTER*2 CIX
      character*2 CJX
      CHARACTER*13 CODNAM
c     DIMENSION ISUBX(100)
c     LOGICAL ANSL1
C
      DIMENSION CONSTX(*),IPOINC(*),PCWORD(*),CPLOTX(*)
```

```
C  End of first part of STRUCT written by "GENTEXT"
C=======================================================================
=
C
C  INSERT ADDITIONAL DIMENSION AND/OR LABELLED COMMON BLOCKS HERE,
C  IF NECESSARY. THESE WOULD BE STATEMENTS THAT ARE CONSISTENT WITH
C  SUBROUTINES THAT YOU OR OTHERS MAY HAVE WRITTEN THAT ARE REQUIRED
C  FOR WHATEVER ANALYSIS YOU ARE PERSUING.  MAKE SURE THAT YOU DO NOT
C  INTRODUCE NAME CONFLICTS WITH THE "GENTEXT" LABELLED COMMON BLOCKS
C  LISTED ABOVE.
C
C  Please note that you do not have to modify STRUCT.NEW if you would
C  rather provide all of your algorithms via the BEHAVIOR.NEW library.
C  (See instructions in BEHAVIOR.NEW).
C
C  If you are using a lot of software previously written either by
C  yourself or others, or if there are a lot of behavioral constraints
C  that are best generated by looping over array indices (such as
C  occurs, for example, with stress constraints in laminates of
C  composite materials), then it may be best to insert your common
C  blocks and dimension statements here, your subroutine calls
C  below (where indicated), and your subroutines in any of the libraries
C  called ADDCODEn.NEW, n = 1,2,...,5.  Please note that you will
C  probably also have to add statements to SUBROUTINE TRANFR, the
C  purpose of which is described below (in TRANFR).
C
C  The two test cases provided with GENOPT demonstrate both methods:
C
C  PLATE (Test Case 1): leave STRUCT.NEW unchanged and modify BEHAVIOR.NEW
C  PANEL (Test Case 2): leave BEHAVIOR.NEW unchanged except for the
objective
C                       function (SUBROUTINE OBJECT), modify STRUCT.NEW,
C                       add a subroutine library called ADDCODE1.NEW, and
C                       augment the LINKMAIN.COM file to collect object
C                       libraries from other directories (PANDA2, in this
C                       example.)
C  ********************* INSERTED BOSOR4 COMMON BLOCKS *****************
C  Common block needed for SUBROUTINE BOSDEC...
      COMMON/BUCMDX/WMODEX(10000),WSAVEX(10000),WMODX2(10000)
      COMMON/PMAX0X/PMAX01,PMAX02,PMAX
C  Common blocks needed for SUBROUTINE STRUCT...
C     DIMENSION PSTEPS(200),WMODES(1000,10)
      DIMENSION FMARGM(100),FMARGP(100),FMARG(100)
      DIMENSION BUCSKU1S(30),BUCSTU1S(30),STRMAU1S(30),STRSTU1S(30)
      DIMENSION BUCSKU2S(30),BUCSTU2S(30),STRMAU2S(30),STRSTU2S(30)
      DIMENSION BUCMINS(30),BUCMNSS(30),STFMXSS(30),SKNMAXS(30)
      DIMENSION PHORIG(100),SARCLT(100)
      DIMENSION THSKIN(100),THKSTF(100),HEIGHT(100)
```

```
      DIMENSION XYs(2),XYZg(3)
      COMMON/SAVSTF1/BUCSKU1(30),BUCSTU1(30),STRMAU1(30),STRSTU1(30)
      COMMON/SAVSTF2/BUCSKU2(30),BUCSTU2(30),STRMAU2(30),STRSTU2(30)
      COMMON/MODMLT/M1MULT,M2MULT,M1MULTC,M2MULTC,M1MULTB,M2MULTB
      COMMON/PCOLLX/PMXCOL1,PMXCOL2
      COMMON/EIGALL/EIG1,EIG2,EIG3,EIG4
      COMMON/WAVALL/NWAV1,NWAV2,NWAV3,NWAV4
      COMMON/IDVMOD/IFIL8X,ITESTX
      COMMON/PMAX0B/PMAXBUC1,PMAXBUC2
      COMMON/MODSGN/FSIGN1S,FSIGN2S,FSIGNMX1,FSIGNMX2,WSAVE1,WSAVE2
      COMMON/MODSG2/NODESV1,NODESV2
      COMMON/WRDCLX/WRDCOL
      CHARACTER*45 WRDCOL
      COMMON/SFACT/SMOVE,SFACTR
C  Common blocks from bigbosor4...
C BEG MAR 2008
      COMMON/EIGENV/PPP,OMG2,RHO
C END MAR 2008
      COMMON/SHEL/ISHL(295),IWAL(295),ITHK(295),IARC(295),ILOAD(295)
      COMMON/IPHIHI/IPHIOL(295),IARCLT(295),IHISTF(295),ITHSTF(295)
      COMMON/BUCCON/BUCMIN(295),BUCMNS(295),BUCMNR(295)
      COMMON/STRCON/STFMXS(295),STFMXR(295),SKNMAX(295)
      COMMON/IBIGX1/IBUCMN(295),IBUCMS(295),IBUCMR(295)
      COMMON/IBIGX2/ISTFMS(295),ISTFMR(295),ISKNMX(295)
      COMMON/BUCMOD/WMODE(10000)
      COMMON/BUCIDX/IDBUCK(100)
      COMMON/EIGNO/NVEC,EGV(50),AXB
      COMMON/TOTMAX/TOTMAS
      COMMON/SEGS/NSEG,M2,I5(295),I2,I2G
      COMMON/LOCALX/BUCSKN,BUCSTF,STRSTF
      COMMON/LSTEPS/ISTEP
      COMMON/PSTEPX/PSTEP(200),ENDUVS(200)
      COMMON/ENDUVX/ENDUV,STRMAX,ARCLEN
      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/caseblock/CASE
      CHARACTER*28 CASE
      CHARACTER*33 CASA1,CASA2,CASA3,CASA4,CASA5,CASA6,CASA7,CASA8,CASA9
      CHARACTER*33 CASSTAGS
C   Common block needed in WALTST:
      common /nitnot/ nit,  not
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
C  ****************** END OF INSERTED BOSOR4 COMMON BLOCKS *************
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     End of (ILOADX.EQ.1) condition
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 BEG FEB 2008
      WRITE(IFILE8,'(/,/,A,/,A,/,A,I2,A,I2,A,I2,A,I2,/,A)')
     1' ************************************************************',
     1' Start of all analyses:',
     1' Design iteration', JJJ1,', Load Set',ILOADX,', IMODX=',IMODX,
     1', Dec.var.no.,IDV=',IDV,
     1' ************************************************************'
C END FEB 2008
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (IMODX.EQ.0.AND.NPRINX.GT.0) THEN
         WRITE(IFILE8,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1 ' SUBROUTINE STRUCT computes seven "behaviors" (stress, col-',
     1 ' lapse, bifurcation buckling, etc.). The seven behaviors are:',
     1 ' 1. linear axisymmetric buckling of the perfect ellipsoid in',
     1 '    order to obtain 2 or 4 axisymmetric buckling modes (NCASES',
     1 '    = 2 or 4) which are to be used as initial imperfection',
     1 '    shapes in the following analyses 2 - 7, listed next.',
     1 ' 2. nonlinear axisymmetric stress with mode 1 imperfection',
     1 ' 3. nonlinear axisymmetric stress with mode 2 imperfection',
     1 ' 4. axisymmetric collapse with mode 1 imperfection',
     1 ' 5. axisymmetric collapse with mode 2 imperfection',
     1 ' 6. nonlinear bifurcation buckling with mode 1 imperfection',
     1 ' 7. nonlinear bifurcation buckling with mode 2 imperfection.'
         IF (ILOADX.GT.2) WRITE(IFILE8,'(/,A,/,A)')
     1 ' The 7 analyses are repeated with use of mode 3 and mode 4',
     1 ' imperfection shapes.'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
       WRITE(IFILE8,'(/,A,/,A)')
     1 ' Brief description of each of the seven analyses corresponding',
     1 ' to the seven "behaviors" just listed:'
```

```
  IF (ILOADX.LE.2)
1 WRITE(IFILE8,'(/,A,/,A,/,A,/,A)')
1 ' 1. Ten axisymmetric buckling modes are computed from linear',
1 '    analysis. Only two modes are used for imperfection shapes:',
1 '    A. The mode corresponding to the lowest buckling load, and',
1 '    B. one other mode, usually the 2nd mode.'
  IF (ILOADX.LE.2.AND.IMODE.EQ.1)
1      WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A)')
1 '    One of these two modes MUST have a normalized modal normal',
1 '    displacement w at the apex of the shell of at least 0.7.',
1 '    If not,SUBROUTINE STRUCT searches the 10 modes in order of',
1 '    increasing eigenvalue until it finds the lowest mode which',
1 '    satisfies that rule. Then that mode replaces Mode B,',
1 '    since neither Mode A ("mode 1") nor Mode B ("mode 2") does.'
  IF (ILOADX.LE.2.AND.IMODE.EQ.1)
1 WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
1 '    The signs of both "mode 1" and "mode 2" are established so',
1 '    that the maximum normal modal displacement is inward, that',
1 '    is, if this maximum occurs at the apex of the shell, the',
1 '    corresponding buckling modal imperfection forms a "flat"',
1 '    spot in the neighborhood of the apex of the shell. It is',
1 '    known that the buckling and collapse loads of thin spheri-',
1 '    cal caps are very sensitive to imperfections of this type.'
  IF (ILOADX.LE.2)
1 WRITE(IFILE8,'(/,A,/,A,/,A)')
1 '    For each of mode 1 and mode 2, the actual imperfection is',
1 '    the normalized buckling modal w-deflection times an',
1 '    amplitude factor supplied by the user by means of "BEGIN".'
  IF (ILOADX.GT.2) WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
1 '    Mode 3 and Mode 4 from the linear axisymmetric bifurcation',
1 '    buckling analysis are next used as imperfection shapes.',
1 '    There is no special strategy used to adjust the signs of',
1 '    these higher modes, except that a certain effort is made',
1 '    to ensure that the modes for the perturbed (IMODX=1)',
1 '    designs are not approximately the negatives of those for',
1 '    the unperturbed (IMODX=0) design.'
  IF (ILOADX.EQ.1) THEN
  WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
1 '    In MAINSETUP (*.OPT file) the user can choose whether or',
1 '    not the linear axisymmetric buckling modes, (that is, the',
1 '    imperfection shapes) are to be recomputed for each of the',
1 '    PERTURBED designs. If the user answers the prompt,',
1 '      Take "shortcuts" for perturbed designs (Y or N)?',
1 '    with "N" (NO), then the axisymmetric buckling modal',
1 '    imperfection shapes will be recomputed for each PERTURBED',
1 '    design. (This is the preferred choice, even though it',
1 '    leads to some high constraint gradients). If the user'
  WRITE(IFILE8,'(A,/,A,/,A,/,A)')
```

```
     1 '    answers "Y" (YES), then the imperfection shapes will NOT',
     1 '    be recomputed for the PERTURBED designs. The constraint',
     1 '    gradients will be lower, but GENOPT will usually have a',
     1 '    harder time finding the "global" optimum design.'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1' 2. Nonlinear axisymmetric stress analysis with "mode 1" Wimp:',
     1'    This analysis is performed for both +(mode 1) and -(mode 1)',
     1'    For each of these "sub-analyses" the following is done:',
     1'    a. The nonlinear equilibrium path is traced over the range',
     1'       P(design)/10. < P < P(design) in 10 steps of dP, where',
     1'       P(design) = design pressure and dP = P(design)/10.',
     1'    b. If the shell collapses nonlinearly (convergence failure)',
     1'       for P < P(design), then step 2a is redone with the range',
     1'       P(collapse)/10. < P < P(collapse); dP=P(collapse)/10.',
     1'    c. At the maximum load (either P(collapse) or P(design),',
     1'       whichever is smaller) the following quantities are',
     1'       computed:'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      WRITE(IFILE8,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       Region 1 local skin buckling load factor,       BUCMIN',
     1'       Region 1 isogrid member buckling load factor,   BUCMNS',
     1'       Region 1 skin maximum effective stress,         SKNMAX',
     1'       Region 1 isogrid member max. effective stress,  STFMXS',
     1'       Region 2 local skin buckling load factor,       BUCMIN',
     1'       Region 2 isogrid member buckling load factor,   BUCMNS',
     1'       Region 2 skin maximum effective stress,         SKNMAX',
     1'       Region 2 isogrid member max. effective stress,  STFMXS',
     1'       Normal displacement of the shell at its apex,   ENDUV',
     1'       The quantities, BUCMIN, BUCMNS, etc. may constrain the',
     1'       evolution of the optimum design.'
      WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       Region 1 represents the ellipsoidal cap region, and',
     1'       Region 2 represents the rest of the ellipsoidal shell.',
     1'       Note that typical margins contain the following strings:',
     1'          (SKNBK1(1,1)/SKNBK1A(1,1))/SKNBK1F(1,1)-1',
     1'          (SKNBK1(1,2)/SKNBK1A(1,2))/SKNBK1F(1,2)-1',
     1'       with two-dimensional arrays, SKNBK1, SKNBK1A, SKNBK1F,',
     1'       in this example signifying "skin buckling for mode 1".',
     1'       The analogous margins,',
     1'          (SKNBK2(1,1)/SKNBK2A(1,1))/SKNBK2F(1,1)-1',
     1'          (SKNBK2(1,2)/SKNBK2A(1,2))/SKNBK2F(1,2)-1',
     1'       with two-dimensional arrays, SKNBK2, SKNBK2A, SKNBK2F,',
     1'       in this example signify "skin buckling for mode 2".'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      WRITE(IFILE8,
     1     '(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/)')
     1'       The "i" in the arrays *(i,j) is the load set number.',
```

```
     1'       The "j" is the region number, called "Region 1" for',
     1'       Region no. 1 and "Region 2" for region no. 2 above.',
     1'       Region no. 1: the radial coordinate, x,  0 < x < xlimit.',
     1'       Region no. 2: the radial coordinate, x,xlimit < x < xmax',
     1'       where xmax is the value of the x-coord. at the equator,',
     1'       and xlimit is a user-provided input datum, usually',
     1'       equal to about half the semimajor axis (xlimit=a/2).',
     1'       This scheme of computing minimum buckling load factors',
     1'       and maximum stresses in two regions of the ellipsoidal',
     1'       head and having margins for each smooths the values of',
     1'       the margins from design iteration to iteration, making',
     1'       it easier to find a "global" optimum design.'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      WRITE(IFILE8,'(/,A,/,A,/,A)')
     1'       The quantities, BUCMIN, BUCMNS, SKNMAX, STFMXS, are',
     1'       computed in SUBROUTINE PLOCAL in the BIGBOSOR4 code,',
     1'       ..bosdec/sources/addbosor4.src, as follows:'
      WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       COMPUTATION OF BUCMIN: In the following code fragment',
     1'       the critical buckling resultant is NSCRIT; BUCLOD(I) =',
     1'       buckling load factor at nodal point I in Segment No. IS;',
     1'       BUCMIN(IS) = minimum buckling load factor in Segment IS.',
     1'       FCOEF = 0.5',
     1'       NSCRIT = FCOEF*PI**2*CSKIN(4,4,I)/SIDE**2           (1)',
     1'       NSMAX  = MIN(N1SKIN,N2SKIN)                         (2)',
     1'       BUCLOD(I) = NSCRIT/ABS(NSMAX)                       (3)',
     1'       BUCMIN(IS) = MIN(BUCMIN(IS),BUCLOD(I))              (4)',
     1'       in which the variables used in Eqs.(1-4) are as follows:'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      WRITE(IFILE8,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       CSKIN(i,j,I) = 6 x 6 matrix of shell wall stiffnesses at',
     1'                      nodal point I',
     1'       SIDE = length of a side of the equilateral triangle',
     1'              formed by the isogrid configuration',
     1'       N1SKIN, N2SKIN are the meridional and hoop resultants',
     1'                      in the shell skin, given by:',
     1'       N1SKIN = CSKIN(1,1,I)*EPS1 + CSKIN(1,2,I)*EPS2',
     1'               +CSKIN(1,4,I)*K1   + CSKIN(1,5,I)*K2',
     1'       N2SKIN = CSKIN(1,2,I)*EPS1 + CSKIN(2,2,I)*EPS2',
     1'               +CSKIN(2,4,I)*K1   + CSKIN(2,5,I)*K2'
      WRITE(IFILE8,'(A,/,A,/,A,/,A)')
     1'       EPS1, K1 = meridional reference surface membrane',
     1'                  strain and curvature change at nodal point I',
     1'       EPS2, K2 = circumferential reference surface membrane',
     1'                  strain and curvature change at nodal point I'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       The buckling load, NSCRIT, is for a flat equilateral',
```

```
     1'       triangular piece of skin. The formula for NSCRIT is from',
     1'       NACA TN-3781, July 1957 by Gerard & Becker: "Handbook of',
     1'       Structural Stability, Part I - Buckling of Flat Plates".',
     1'       The formula is for buckling of an equilateral flat plate',
     1'       with N1SKIN = N2SKIN (compression). The result here is',
     1'       approximate because in general N1SKIN is not equal to',
     1'       N2SKIN, and in general the skin is not isotropic.'
       WRITE(IFILE8,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       The prediction of the shell skin buckling load factor',
     1'       should be conservative because:',
     1'       a. The compressive stress resultant used in the formula',
     1'          for buckling load factor is NSMAX=MIN(N1SKIN,N2SKIN).',
     1'       b. The triangular piece of skin is assumed to be flat',
     1'          when in fact it is curved.',
     1'       c. The triangular piece of skin is assumed to be simply',
     1'          supported when in fact it is supported by isogrid',
     1'          stiffeners along all three edges.'
       WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       COMPUTATION OF BUCMNS AND STFMXS: In the code fragment',
     1'       in PLOCAL that computes stiffener buckling and stress,',
     1'       BUCMNS(IS) and STFMXS(IS), useful definitions are:',
     1'       NUSTIF = Poisson ratio for stringer/isogrid member',
     1'       SIGCR  = buckling stress for stringer/isogrid member',
     1'       STRTIP = stress at the tip of stringer/isogrid member',
     1'       STRROT = stress at the root of the stringer/isogrid',
     1'       BUCSTR(I) = buckling load factor for stringer/isogrid',
     1'                   at nodal point I',
     1'       BUCMNS(IS)= minimum buckling load factor for stiffener',
     1'                   in shell segment IS'
C2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
       WRITE(IFILE8,'(A,/,A,/,A,/,A)')
     1'       STRSTR(I) = maximum stress in stringer/isogrid at nodal',
     1'                   point I',
     1'       STFMXS(IS)= maximum stress in stringer/isogrid in shell',
     1'                   segment IS'
       WRITE(IFILE8,
     1'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       The critical buckling load of stiffener is derived from',
     1'       formulas from ROARK: FORMULAS FOR STRESS AND STRAIN,',
     1'       3rd Edition, McGraw-Hill, 1954, Table XVI, p. 312,',
     1'       Formulas 4 (s.s.,free) and 5 (clamped,free). Roark has',
     1'       SIGCR = k*[ESTIFF/(1-NUSTIF**2)]*(TSTIFF/HEIGHT)**2',
     1'       in which k is a coefficient that depends on the aspect',
     1'       ratio of the plate (stiffener). For long, uniformly',
     1'       axially compressed plates:',
     1'       a. k = 0.375 if the plate is simply-supported-free',
     1'       b. k = 1.1   if the plate is clamped-free',
     1'       Later edition of "ROARK":',
```

```
     1'       Seventh Edition by Warren C. Young and Richard G.',
     1'       Budynas, McGraw-Hill 2002, Chapter 15, Table 15.2,',
     1'       Formulas 1.d and 1.e, on p. 730'
C23456789012345678901234567890123456789012345678901234567890123456789012
      WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'     More definitions...',
     1'       IRECT(1,IS) = 1 if stringer/isogrid member has a',
     1'                       rectangular cross section',
     1'                   = 0 if stringer/isogrid member does not have',
     1'                       a rectangular cross section',
     1'       INTEXT(1,IS)= 0 for stringer/isogrid attached to the',
     1'                       leftmost shell skin surface',
     1'                       (e.g. internal smeared stringer/isogrid)',
     1'       INTEXT(1,IS)= 1 for stringer/isogrid attached to the',
     1'                       rightmost shell skin surface'
      WRITE(IFILE8,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       Z(I) = distance from the shell skin leftmost surface to',
     1'              the reference surface at nodal point I. (The',
     1'              reference surface is where the membrane strain',
     1'              and curvature changes (EPS1,K1,EPS2,K2) are',
     1'              measured].',
     1'       T(I) = thickness of shell skin at nodal point I of',
     1'              shell segment IS',
     1'       ZTIP = distance from shell reference surface to the tip',
     1'              of stringer/isogrid',
     1'       STRTIP = stress at the tip of a smeared stringer/isogrid',
     1'                member.'
      WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       STFPRP(j,1,I) = properties of smeared stringer/isogrd',
     1'                       at nodal point I, defined as follows:',
     1'          STFPRP(1,1,I) = stiffener thickness, TSTIFF',
     1'          STFPRP(2,1,I) = stiffener height from nearest shell',
     1'                          skin surface',
     1'          STFPRP(3,1,I) = stiffener spacing: SIDE*SQRT(3.)/2.',
     1'          STFPRP(4,1,I) = stiffener elastic modulus',
     1'       STFPRP(j,2,I), j = 1,2,3,4 = same as above, for smeared',
     1'                           rings.'
C23456789012345678901234567890123456789012345678901234567890123456789012
      WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       SUBROUTINE PLOCAL has the following code for computing',
     1'       buckling and stress in the stiffener/isogrid member:',
     1'       IF (INTEXT(1,IS).EQ.0) ZTIP = -(STFPRP(2,1,I) + Z(I))',
     1'       IF (INTEXT(1,IS).EQ.1) ZTIP = STFPRP(2,1,I) + T(I) -Z(I)',
     1'       STRTIP = STFPRP(4,1,I)*(EPS1 - ZTIP*K1)',
     1'       EDGSTF = 0.5',
     1'       NUSTIF = 0.3',
     1'       SIGCR =(0.375+0.7*EDGSTF)*(STFPRP(4,1,I)/(1.-NUSTIF**2))',
     1'                                 *(STFPRP(1,1,I)/STFPRP(2,1,I))**2'
```

```
      WRITE(IFILE8,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       IF (STRTIP.LT.0.0) THEN',
     1'          BUCSTR(I) = SIGCR/ABS(STRTIP)',
     1'          BUCMNS(IS) = MIN(BUCMNS(IS),BUCSTR(I))',
     1'       ENDIF',
     1'       IF (INTEXT(1,IS).EQ.0) ZROOT = -Z(I)',
     1'       IF (INTEXT(1,IS).EQ.1) ZROOT =  T(I) - Z(I)',
     1'       STRROT = STFPRP(4,1,I)*(EPS1 - ZROOT*K1)',
     1'       STRSTR(I) = MAX(ABS(STRTIP),ABS(STRROT))',
     1'       STFMXS(IS) = MAX(STFMXS(IS),STRSTR(I))'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      WRITE(IFILE8,'(/,A,/,A,/,A)')
     1'       The stiffener buckling load factor and maximum stress',
     1'       used here should be conservative compared to what',
     1'       happens in the case of an actual isogrid member because:'
      WRITE(IFILE8,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       a. The compressive stress STRTIP at the tip of the',
     1'          stiffener is used, which in the worst case would be',
     1'          the maximum compressive stress over the height of the',
     1'          stiffener, whereas the ROARK formula for buckling',
     1'          is for a uniformly compressed flat plate.',
     1'       b. For typical optimum designs the aspect ratio of the',
     1'          plate is about 2.0, for which ROARK gives a buckling',
     1'          coefficient, k = 0.574 for a plate simply supported',
     1'          along one edge and free along the opposite edge.'
      WRITE(IFILE8,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       c. Where the isogrid members intersect the actual b.c.',
     1'          should probably be clamped, whereas the formula is',
     1'          for simple support along plate edges "b".',
     1'       d. The formula for maximum stress at the stiffener tip,',
     1'          STRTIP = STFPRP(4,1,I)*(EPS1 - ZTIP*K1)',
     1'          is based on the assumption that the isogrid member',
     1'          is oriented meridionally. This is the worst possible',
     1'          orientation from the point of view of maximum stress',
     1'          for a stiffener attached to an axisymmetrically',
     1'          deformed shell.'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,/,A,/,A,/,A)')
     1'       COMPUTATION OF SKNMAX: The maximum effective stress in',
     1'       the skin of the shell segment IS is computed by',
     1'       BIGBOSOR4 as it always has been. No new coding was added',
     1'       to BIGBOSOR4 in order to generate SKNMAX(IS).',
     1'       COMPUTATION OF ENDUV:  The normal displacement w at the',
     1'       apex of the ellipsoidal head is computed by BIGBOSOR4',
     1'       as it always has been. No new coding has been added.'
      WRITE(IFILE8,'(A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'       NOTE: prebuckling axial displacement at the first nodal',
     1'       point in the cylindrical segment (Segment NSEG) is set',
```

```
     1'       to zero in the prebuckling phase of the analysis only.',
     1'       This is done so that ENDUV is for the ellipsoidal head',
     1'       by itself (does not include any axial deformation of the',
     1'       cylindrical segment to which the ellipsoidal head is',
     1'       attached).'
C2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
       WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'    d. Steps 2a, 2b, 2c are repeated for the negative of mode 1',
     1'       that is, for -(mode 1).',
     1'    e. Both +(mode 1) and -(mode 1) behavior are investigated',
     1'       for both the UNPERTURBED (current) and PERTURBED designs',
     1'    f. Based on the results from the +(mode 1) and -(mode 1)',
     1'       nonlinear analyses, SUBROUTINE STRUCT may choose which',
     1'       condition is worst for determination of the items listed',
     1'       under 2c (BUCMIN, BUCMNS, etc) and which condition is',
     1'       worst for determination of the collapse pressure, which',
     1'       later becomes one of the margins. These choices hold for',
     1'       the nonlinear stress and collapse analyses of the',
     1'       PERTURBED designs (IMODX = 1).'
       WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'    g. It is generally best to use multiple load sets in order',
     1'       to compute margins with +(modal imperfection shapes) and',
     1'       -(modal imperfection shapes) separately instead of using',
     1'       SUBROUTINE STRUCT to choose the worst of (+) and (-)',
     1'       imperfection shapes in a single load set, as described',
     1'       in f. Experience has demonstrated that processing (+)',
     1'       and (-) imperfection shapes in separate load sets leads',
     1'       to smoother plots of margins vs design iterations and',
     1'       also to smaller minimum weights.'
C2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
       WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1' 3. Nonlinear axisymmetric stress analysis with "mode 2" Wimp:',
     1'    This analysis is performed for both +(mode 2) and -(mode 2)',
     1'    in exactly the same manner as just described for mode 1.',
     1' ',
     1' 4. Axisymmetric collapse with + or - mode 1 imperfection.',
     1'    Which of the +(mode 1) or -(mode 1) imperfections is used',
     1'    has already been determined as described in Steps 2a-f.',
     1'    The nonlinear equilibrium path is traced over the range',
     1'    PMAX/10. < P < 2.*PMAX in 20 steps of dP, where',
     1'    PMAX=either P(design) or P(collapse), whichever is smaller',
     1'    and dP = PMAX/10.'
C2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
       WRITE(IFILE8,'(/,A,/,A,/,A)')
     1' 5. Axisymmetric collapse with + or - mode 2 imperfection.',
     1'    Which of the +(mode 2) or -(mode 2) imperfections is used',
     1'    has already been determined as described in Step 3.'
       WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
```

```
     1' 6. Nonlinear bifurcation buckling with mode 1 imperfection:',
     1'    For the UNPERTURBED (current) design (IMODX=0), nonlinear',
     1'    bifurcation buckling is investigated over a range of',
     1'    circumferential wave numbers from 0 to 10 with the load',
     1'    set equal to PMAX if PMAX = P(design) or 0.9*PMAX if',
     1'    PMAX = P(collapse). This is done for BOTH +(mode 1) and for',
     1'    -(mode 1) imperfections. SUBROUTINE STRUCT decides which',
     1'    of the conditions, +(mode 1) or -(mode 1), is the worst.',
     1'    This choice holds for the mode 1 bifurcation buckling',
     1'    analyses of the PERTURBED (IMODX=1) designs.'
       WRITE(IFILE8,'(/,A,/,A,/,A)')
     1' 7. Nonlinear bifurcation buckling with mode 2 imperfection:',
     1'    This is done in exactly the same way as for the mode 1',
     1'    imperfection; see Step 6.'
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
       WRITE(IFILE8,
     1 '(/,A,/,A,/,A,/,A,/,A,/,A,/,A,A,A,/,/,A,/,A,/,A,/,A)')
     1' A NOTE ABOUT DESIGN Margins...',
     1' The margins for an optimized isogrid-stiffened ellipsoidal',
     1' shell with shell skin thickness and isogrid height varying',
     1' along the meridian (callout points at the pole, at the',
     1' junctions between each toroidal segment of the equivalent',
     1' ellipsoid, and at the equator of the equivalent ellipsoid:',
     1' (case name =',CASE,') are as follows:',
     1' For mode 1 buckling modal imperfection shape:',
     1' Margins CORRESPONDING TO CURRENT DESIGN (FS= FACTOR OF SAFETY)',
     1' MAR. CURRENT',
     1' NO.   VALUE            DEFINITION'
       WRITE(IFILE8,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'  1  2.303E-01 (CLAPS1(1)/CLAPS1A(1))/CLAPS1F(1)-1;FS=1.0',
     1'  2  9.988E-01 (GENBK1(1)/GENBK1A(1))/GENBK1F(1)-1;FS=1.0',
     1'  3  3.853E-02 (SKNBK1(1,1)/SKNBK1A(1,1))/SKNBK1F(1,1)-1;FS=1.0',
     1'  4 -1.235E-02 (SKNBK1(1,2)/SKNBK1A(1,2))/SKNBK1F(1,2)-1;FS=1.0',
     1'  5  6.174E-01 (STFBK1(1,1)/STFBK1A(1,1))/STFBK1F(1,1)-1;FS=1.0',
     1'  6  1.564E-01 (STFBK1(1,2)/STFBK1A(1,2))/STFBK1F(1,2)-1;FS=1.0',
     1'  7  6.878E-02 (SKNST1A(1,1)/SKNST1(1,1))/SKNST1F(1,1)-1;FS=1.0',
     1'  8  1.294E-02 (SKNST1A(1,2)/SKNST1(1,2))/SKNST1F(1,2)-1;FS=1.0',
     1'  9 -3.474E-02 (STFST1A(1,1)/STFST1(1,1))/STFST1F(1,1)-1;FS=1.0',
     1' 10  2.015E-02 (STFST1A(1,2)/STFST1(1,2))/STFST1F(1,2)-1;FS=1.0',
     1' 11  3.439E-01 (WAPEX1A(1)/WAPEX1(1))/WAPEX1F(1)-1;FS=1.0'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
       WRITE(IFILE8,'(/,A,/,A,/,A,/,A)')
     1' For mode 2 buckling modal imperfection shape:',
     1' Margins CORRESPONDING TO CURRENT DESIGN (FS= FACTOR OF SAFETY)',
     1' MAR. CURRENT',
     1' NO.   VALUE            DEFINITION'
       WRITE(IFILE8,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
```

```
     1' 12  8.393E-02 (CLAPS2(1)/CLAPS2A(1))/CLAPS2F(1)-1;FS=1.0',
     1' 13  8.220E-01 (GENBK2(1)/GENBK2A(1))/GENBK2F(1)-1;FS=1.0',
     1' 14  6.012E-02 (SKNBK2(1,1)/SKNBK2A(1,1))/SKNBK2F(1,1)-1;FS=1.0',
     1' 15 -2.458E-02 (SKNBK2(1,2)/SKNBK2A(1,2))/SKNBK2F(1,2)-1;FS=1.0',
     1' 16  2.769E+00 (STFBK2(1,1)/STFBK2A(1,1))/STFBK2F(1,1)-1;FS=1.0',
     1' 17  4.838E-02 (STFBK2(1,2)/STFBK2A(1,2))/STFBK2F(1,2)-1;FS=1.0',
     1' 18  9.176E-02 (SKNST2A(1,1)/SKNST2(1,1))/SKNST2F(1,1)-1;FS=1.0',
     1' 19  1.170E-02 (SKNST2A(1,2)/SKNST2(1,2))/SKNST2F(1,2)-1;FS=1.0',
     1' 20  1.049E-01 (STFST2A(1,1)/STFST2(1,1))/STFST2F(1,1)-1;FS=1.0',
     1' 21 -1.931E-02 (STFST2A(1,2)/STFST2(1,2))/STFST2F(1,2)-1;FS=1.0',
     1' 22  1.185E+00 (WAPEX2A(1)/WAPEX2(1))/WAPEX2F(1)-1;FS=1.0'
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
       WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A)')
     1' In these margins the "A" endings in names such as "CLAPS1A"',
     1' denote "allowable". The "F" endings in names such as "CLAPS1F"',
     1' denote "factor of safety". The margins are equal to the',
     1' corresponding behavioral constraints minus 1.0. The chart',
     1' below lists names that characterize the margin depending on',
     1' its value, as follows:'
C
       WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')
     1'                   The most negative    The most negative',
     1' Designation       margin must be       margin must be less',
     1'                   greater than:        than or equal to:',
     1' -----------       --------------       ------------------',
     1' "FEASIBLE"            -0.01              ------',
     1' "ALMOST FEASIBLE"     -0.05              -0.01',
     1' "MILDLY UNFEASIBLE"   -0.10              -0.05',
     1' "MORE UNFEASIBLE"     -0.15              -0.10',
     1' "MOSTLY UNFEASIBLE"   -0.20              -0.15',
     1' "NOT FEASIBLE"        ------             -0.20'
C
      ENDIF
C     End of (ILOADX.EQ.1) condition
C
      ENDIF
C     End of (IMODX.EQ.0.AND.NPRINX.GT.0) condition
C
      IFIL8X = IFILE8
      ITESTX = 0
C
C   initialize behaviors:
C
C   NOTE: ILOADX = load set number.
C
C    behaviors from mode 1 (and possibly mode 3) imperfection shape(s)...
      CLAPS1(ILOADX) = 10.E+16
```

```
      GENBK1(ILOADX) = 10.E+16
      SKNBK1(ILOADX,1) = 10.E+16
      STFBK1(ILOADX,1) = 10.E+16
      SKNST1(ILOADX,1) = 0.
      STFST1(ILOADX,1) = 0.
      SKNBK1(ILOADX,2) = 10.E+16
      STFBK1(ILOADX,2) = 10.E+16
      SKNST1(ILOADX,2) = 0.
      STFST1(ILOADX,2) = 0.
      WAPEX1(ILOADX) = 0.
C
C    behaviors from mode 2 (and possibly mode 4) imperfection shape(s)...
      CLAPS2(ILOADX) = 10.E+16
      GENBK2(ILOADX) = 10.E+16
      SKNBK2(ILOADX,1) = 10.E+16
      STFBK2(ILOADX,1) = 10.E+16
      SKNST2(ILOADX,1) = 0.
      STFST2(ILOADX,1) = 0.
      SKNBK2(ILOADX,2) = 10.E+16
      STFBK2(ILOADX,2) = 10.E+16
      SKNST2(ILOADX,2) = 0.
      STFST2(ILOADX,2) = 0.
      WAPEX2(ILOADX) = 0.
C
C  Find axisymmetric (n=0) bifurcation buckling modes for perfect shell
C  from linear bifurcation buckling analysis. These modes are to be used
C  as the initial imperfection shapes.
C
C  The mass is stored in TOTMAS, which is one of the BOSOR4 labelled
C  common blocks.
C
      IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
      WRITE(IFILE8,'(A,I2,A,/,A,A,I2)')
     1' ============ Analysis No. 1 for Load Set No.',ILOADX,
     1' ============',
     1' **** Start linear axisymmetric bifurcation buckling of',
     1' perfect shell. IMODX=',IMODX
      IF (ILOADX.LE.2) WRITE(IFILE8,'(A,/,A)')
     1' **** The purpose is to get two axisymmetric buckling modal',
     1' **** imperfection shapes: mode 1 and mode 2.'
      IF (ILOADX.GT.2) WRITE(IFILE8,'(A,/,A)')
     1' **** The purpose is to get two axisymmetric buckling modal',
     1' **** imperfection shapes: mode 3 and mode 4.'
C23456789012345678901234567890123456789012345678901234567890123456789012
      INDIC = 1
      IMPERF = 0
      CALL BOSDEC(1,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
```

```
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
      CALL B4READ
      IF (ITYPEX.EQ.2) THEN
C      Get CASE.ALL1 file for input for BIGBOSOR4...
C      CASE.ALL1 is an input file for BIGBOSOR4 for linear bifurcation
C      buckling of the perfect ellipsoidal shell.
        I=INDEX(CASE,' ')
        IF(I.NE.0) THEN
           CASA1=CASE(:I-1)//'.ALL1'
        ELSE
           CASA1=CASE//'.ALL1'
        ENDIF
        OPEN(UNIT=61,FILE=CASA1,STATUS='UNKNOWN')
        CALL BOSDEC(1,ILOADX,INDIC,IMPERF,61,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
        CLOSE(UNIT=61)
        WRITE(IFILE8,'(A)')
     1 ' BIGBOSOR4 input file for linear buckling,perfect shell=',CASA1
C
C      Get CASE.STAGS file for input for STAGS SUBROUTINE WALL...
C      CASE.STSGS contains meridionally varying quantities
C      possibly to be used later as input data for SUBROUTINE WALL.
        IF (ILOADX.EQ.1) THEN
        I=INDEX(CASE,' ')
        IF(I.NE.0) THEN
           CASSTAGS=CASE(:I-1)//'.STAGS'
        ELSE
           CASSTAGS=CASE//'.STAGS'
        ENDIF
        OPEN(UNIT=61,FILE=CASSTAGS,STATUS='UNKNOWN')
C
C  Next, build the CASSTAGS file...
C  Retrieve angle, PHORIG and arc length S (X-coordinates),
C  shell skin thickness THSKIN, stringer thickness, THKSTF, and
C  stringer height, HEIGHT for constructing a file possibly to
C  be used later by user-subroutine WALL for STAGS models with thickness
C  that varies meridionally.
C       WRITE(IFILE8,'(A,I2)') ' before GASPS: NSEG=',NSEG
        WRITE(61,'(/,A38,I4)')
     1    '     Number of shell segments (units)=',NSEG
        WGTMAT = DNMATL*386.4
        WRITE(61,'(/,A38,1P,4E14.6)')
     1    '   Isogrid spacing,modulus,nu,density=',
```

```
     1        SPACNG,EMATL,NUMATL,WGTMAT
          DO 3 ISEG = 1,NSEG
C            WRITE(IFILE8,'(A,/,6I5)')
C     1     ' ISEG,IPHIOL,IARCLT,ITHK,ITHSTF,IHISTF=',
C     1       ISEG,IPHIOL(ISEG),IARCLT(ISEG),ITHK(ISEG),
C     1       ITHSTF(ISEG),IHISTF(ISEG)
             CALL GASP(PHORIG,I5(ISEG),3,IPHIOL(ISEG))
             CALL GASP(SARCLT,I5(ISEG),3,IARCLT(ISEG))
             CALL GASP(THSKIN,I5(ISEG),3,ITHK(ISEG))
             CALL GASP(THKSTF,I5(ISEG),3,ITHSTF(ISEG))
             CALL GASP(HEIGHT,I5(ISEG),3,IHISTF(ISEG))
             I5I = I5(ISEG)
             WRITE(61,'(/,A38,I3,A2,I4)')
     1     '               Nodal points in Segment',ISEG,' =',I5I
             WRITE(61,'(/,A38,/(1P5E14.6))')
     1     '                 Angle (X-coordinate)=', (PHORIG(I),I=1,I5I)
             WRITE(61,'(/,A38,/(1P5E14.6))')
     1     ' Meridional arc length (X-coordinate)=', (SARCLT(I),I=1,I5I)
             WRITE(61,'(/,A38,/(1P5E14.6))')
     1     '                 Shell skin thickness=', (THSKIN(I),I=1,I5I)
             WRITE(61,'(/,A38,/(1P5E14.6))')
     1     '        Stringer (or isogrid)  height=', (HEIGHT(I),I=1,I5I)
             WRITE(61,'(/,A38,/(1P5E14.6))')
     1     '     Stringer (or isogrid)  thickness=', (THKSTF(I),I=1,I5I)
    3     CONTINUE
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
          CLOSE(UNIT=61)
          WRITE(IFILE8,'(A)')
     1 ' Input file for SUBROUTINE WALL for STAGS models=',CASSTAGS
          ENDIF
C         End of (ILOADX.EQ.1) condition
C
C*******************************************************************
C Test SUBROUTINE WALTST (The user-written "WALL" routine for STAGS)
C         not = 6
C         XYs(1) = 38.79148
C         XYs(2) = 0.
C         iunit = 2
C         ielt = 25
C         kelt = 480
C         XYZg(1) = 18.
C         XYZg(2) = 0.
C         XYZg(3) = 0.
C         CALL WALTST( iunit, ielt, kelt, XYZg, XYs,
C     1                zeta,  ecz,  ilin, iplas )
C         XYs(1) = 67.090936
C         XYs(2) = 0.
```

```
C        iunit = 2
C        ielt = 15
C        kelt = 480
C        XYZg(1) = 8.
C        XYZg(2) = 0.
C        XYZg(3) = 0.
C        CALL WALTST( iunit, ielt, kelt, XYZg, XYs,
C    1                zeta,  ecz,  ilin, iplas )
C*****************************************************************
      ENDIF
C     End of (ITYPEX.EQ.2) condition
C
      IF (IMODX.EQ.0.OR.IFAST.EQ.0) THEN
         CALL B4MAIN
         IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
         CALL B4POST
      ENDIF
      GRAVTY = 386.4
      WEIGHT = GRAVTY*TOTMAS
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      WRITE(IFILE8,'(/,A,2I8)')' *** In STRUCT: IMODX, IDV=',IMODX,IDV
      WRITE(IFILE8,'(A,1P,E12.4)')' ****** WEIGHT=',WEIGHT
      IF (IMODX.EQ.0.OR.IFAST.EQ.0) THEN
         WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' Linear buckling eigenvalues from BIGBOSOR4, EGV(i)=',
     1   (EGV(I),I=1,NVEC)
         BUCPRS = EIGCRT*PRESS(ILOADX)/1000.
         WRITE(IFILE8,'(A,1P,E12.4)')
     1 ' Linear axisymmetric buckling pressure of perfect shell=',
     1   BUCPRS
         WRITE(IFILE8,'(A,1P,E12.4)')
     1 ' Buckling modal normal displacement w at apex of shell,=',
     1   WMODE(1)
      ENDIF
      IF (IMODX.EQ.0.OR.IFAST.EQ.0) THEN
C
       IF (ILOADX.LE.2.AND.IMODE.EQ.2) THEN
         CALL GASP(WSAVEX,I2,3,IDBUCK(1))
         CALL GASP(WMODX2,I2,3,IDBUCK(2))
         GO TO 3050
       ENDIF
C
       IF (ILOADX.GT.2) THEN
         CALL GASP(WSAVEX,I2,3,IDBUCK(3))
         CALL GASP(WMODX2,I2,3,IDBUCK(4))
         GO TO 3050
       ENDIF
```

```
      KOUNT = 1
    5 CONTINUE
      FMULT = -1.0
      IF (ABS(WMODE(1)).GE.0.7) THEN
         FMULT = WMODE(1)/ABS(WMODE(1))
         DO 8 I = 1,I2
            WMODEX(I) = -FMULT*WMODE(I)
            IF (KOUNT.GT.1) WMODX2(I) = WMODEX(I)
            IF (KOUNT.EQ.1) WSAVEX(I) = WMODEX(I)
    8    CONTINUE
         IF (NPRINX.GT.0) THEN
            WRITE(IFILE8,'(A,I3,A,1P,E12.4)')
     1     ' Multiplier for buckling mode shape no.',KOUNT,
     1     ' -FMULT =',-FMULT
         ENDIF
C23456789012345678901234567890123456789012345678901234567890123456789012

      ELSE
C     End of (ABS(WMODE(1)).GE.0.7) condition
         WMAX = 0.
         WMIN = 0.
         DO 10 I = 1,I2
           WMAX = MAX(WMAX,WMODE(I))
           WMIN = MIN(WMIN,WMODE(I))
   10    CONTINUE
         KNODE = 0
         DO 15 I = 1,I2
           DIF1 = ABS(WMAX - WMODE(I))/ABS(WMAX)
           IF (DIF1.LT.0.001) THEN
              KNODE = I
              IF (WMAX.GT.0.95) THEN
                 FMULT = 1.0
                 GO TO 16
              ENDIF
           ENDIF
   15    CONTINUE
   16    CONTINUE
         DO 20 I = 1,I2
            WMODEX(I) = -FMULT*WMODE(I)
   20    CONTINUE
C23456789012345678901234567890123456789012345678901234567890123456789012
         IF (NPRINX.GT.0) THEN
            WRITE(IFILE8,'(/,A,I3,A,1P,E12.4)')
     1     ' Multiplier for buckling mode shape no.',KOUNT,
     1     ' -FMULT =',-FMULT
           WRITE(IFILE8,'(A,1P,E12.4,/,A,1P,E12.4)')
     1     ' Maximum buckling modal displacement, WMAX  =',WMAX,
     1     ' Minimum buckling modal displacement, WMIN  =',WMIN
```

```
            IF (KNODE.GT.0) WRITE(IFILE8,'(A,I4,/,A,I4)')
     1     ' Nodal point at which max. buckling mode occurs, KNODE=',
     1       KNODE,
     1     ' Number of nodal points in spherical cap,        I5(1)=',
     1       I5(1)
         ENDIF
       ENDIF
C      End of (ABS(WMODE(1)).LT.0.7) condition
C
       IF (ABS(WMODE(1)).LT.0.70) THEN
         WRITE(IFILE8,'(/,A,/,A,/,A,1P,E12.4,/,A)')
     1 ' ************* WARNING ************* WARNING *************',
     1 ' Buckling modal displacement at the apex is less than unity',
     1 ' Buckling modeal displacement at apex,WMODE(1) =',WMODE(1),
     1 ' ********** END WARNING ********** END WARNING ***********'
       ENDIF
C
       WMODE1 = WMODE(1)
       IF (ABS(WMODE1).LT.0.7) THEN
          IF (KOUNT.EQ.1) THEN
             DO 21 I = 1,I2
                WSAVEX(I) = WMODEX(I)
   21        CONTINUE
C            IF (I2.LE.1000) CALL MOVER(WMODE,1,WMODES(1,1),1,I2)
          ENDIF
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
          KOUNT = KOUNT + 1
          IF (KOUNT.LE.NVEC) THEN
             CALL GASP(WMODE,I2,3,IDBUCK(KOUNT))
C            IF (I2.LE.1000) CALL MOVER(WMODE,1,WMODES(1,KOUNT),1,I2)
             WRITE(IFILE8,'(/,A,/,A,/,A,I3,/,A,I3,A,1P,E12.4,/,A)')
     1 ' ************* WARNING ************* WARNING *************',
     1 ' Previous mode shape is too small at shell apex. Try again.',
     1 ' This time, try axisymmetric mode no.',KOUNT,
     1 ' Mode no.',KOUNT,' has amplitude at apex,WMODE(1)=',WMODE(1),
     1 ' ********** END WARNING ********** END WARNING ***********'
             GO TO 5
          ELSE
             DO 22 I = 1,I2
               WMODEX(I) = WSAVEX(I)
   22        CONTINUE
          ENDIF
       ENDIF
C      End of (ABS(WMODE(1)).LT.0.7) condition
C
       WMODE1 = WMODE(1)
       IF (KOUNT.EQ.1.AND.ABS(WMODE1).GE.0.7) THEN
```

```
          CALL GASP(WMODE,I2,3,IDBUCK(2))
          FMULT = -1.0
          IF (ABS(WMODE(1)).GE.0.7) THEN
             FMULT = WMODE(1)/ABS(WMODE(1))
             DO 24 I = 1,I2
                WMODX2(I) = -FMULT*WMODE(I)
   24        CONTINUE
             IF (NPRINX.GT.0) THEN
                WRITE(IFILE8,'(/,A,1P,E12.4)')
     1         ' Multiplier for buckling mode shape no. 2,-FMULT=',-FMULT
             ENDIF
C23456789012345678901234567890123456789012345678901234567890123456789012
          ELSE
C         End of (ABS(WMODE(1)).GE.0.7) condition
             WMAX = 0.
             WMIN = 0.
             DO 26 I = 1,I2
               WMAX = MAX(WMAX,WMODE(I))
               WMIN = MIN(WMIN,WMODE(I))
   26        CONTINUE
             KNODE = 0
             DO 28 I = 1,I2
               DIF1 = ABS(WMAX - WMODE(I))/ABS(WMAX)
               IF (DIF1.LT.0.001) THEN
                  KNODE = I
                  IF (WMAX.GT.0.95) THEN
                     FMULT = 1.0
                     GO TO 30
                  ENDIF
               ENDIF
   28        CONTINUE
   30        CONTINUE
             DO 32 I = 1,I2
                WMODX2(I) = -FMULT*WMODE(I)
   32        CONTINUE
C23456789012345678901234567890123456789012345678901234567890123456789012
             IF (NPRINX.GT.0) THEN
                WRITE(IFILE8,'(/,A)')
     1         ' **** Buckling mode shape no. 2 ****'
               WRITE(IFILE8,'(A,1P,E12.4,/,A,1P,E12.4,/,A,1P,E12.4)')
     1         ' Maximum buckling modal displacement, WMAX  =',WMAX,
     1         ' Minimum buckling modal displacement, WMIN  =',WMIN,
     1         ' Multiplier for buckling mode shape, -FMULT =',-FMULT
               IF (KNODE.GT.0) WRITE(IFILE8,'(A,I4,/,A,I4)')
     1         ' Nodal point at which max. buckling mode occurs, KNODE=',
     1           KNODE,
     1         ' Number of nodal points in spherical cap,        I5(1)=',
     1           I5(1)
```

```
            ENDIF
          ENDIF
C         End of (ABS(WMODE(1)).LT.0.7) condition
C
          IF (ABS(WMODE(1)).LT.0.70) THEN
            WRITE(IFILE8,'(/,A,/,A,/,A,1P,E12.4,/,A)')
     1    ' ********** Buckling mode shape no. 2 ************',
     1    ' Buckling modal displacement at the apex is less than unity',
     1    ' Buckling modeal displacement at apex,WMODE(1) =',WMODE(1),
     1    ' ************************************************'
          ENDIF
        ENDIF
C       End of (KOUNT.EQ.1.AND.ABS(WMODE1).GE.0.7) condition
C
 3050  CONTINUE
C
        DO 3100 I = 1,I2
          DIFF = ABS(1. - ABS(WSAVEX(I)))
          IF (DIFF.LT.0.00001) THEN
            FSIGN1 = WSAVEX(I)/ABS(WSAVEX(I))
            IF (IMODX.EQ.0) THEN
              NODESV1 = I
              FSIGN1S = FSIGN1
            ENDIF
            IF (IMODX.EQ.1) THEN
              SIGNA = FSIGN1
              SIGNB = WSAVEX(NODESV1)/ABS(WSAVEX(NODESV1))
              FPROD = SIGNA*SIGNB
              IF (FPROD.LT.0.) FSIGN1 = SIGNB
              IF (ILOADX.GT.2) THEN
                FPROD = FSIGN1S*FSIGN1
                IF (FPROD.LT.0.) THEN
                  DO 3075 J = 1,I2
                    WSAVEX(J) = -1.*WSAVEX(J)
 3075             CONTINUE
                ENDIF
                GO TO 3175
              ENDIF
            ENDIF
            GO TO 3100
          ENDIF
 3100   CONTINUE
        IF (IMODX.EQ.0) THEN
          FSIGNMX1 = FSIGN1
          WSAVE1 = WSAVEX(1)
          IF (ABS(WSAVE1).GE.0.7) THEN
            FSIGN1S = WSAVE1/ABS(WSAVE1)
          ELSE
```

```
            FSIGN1S = FSIGN1
         ENDIF
      ENDIF
      IF (IMODX.EQ.1) THEN
         FSIGN1SS = FSIGN1S
         IF (ABS(WSAVE1).GE.0.7) THEN
            COMPAR = 0.7
            IF (ABS(WSAVE1).GE.0.99999) COMPAR = 0.1
            IF (ABS(WSAVEX(1)).GE.COMPAR) THEN
               FSIGN1 = WSAVEX(1)/ABS(WSAVEX(1))
            ELSE
               FSIGN1S = FSIGNMX1
            ENDIF
         ENDIF
         IF (ABS(WSAVE1).LT.0.7.AND.ABS(WSAVEX(1)).GE.0.7) THEN
            FSIGN1S = FSIGNMX1
            FSIGN1 = WSAVEX(NODESV1)/ABS(WSAVEX(NODESV1))
         ENDIF
         FPROD = FSIGN1S*FSIGN1
         IF (FPROD.LT.0.) THEN
            DO 3150 I = 1,I2
               WSAVEX(I) = -1.*WSAVEX(I)
 3150        CONTINUE
         ENDIF
         FSIGN1S = FSIGN1SS
      ENDIF
C     End of (IMODX.EQ.1) condition
 3175 CONTINUE
C
      IF (ILOADX.LE.2.AND.NPRINX.GT.0) WRITE(IFILE8,'(/,/,A)')
     1' ********* Buckling modal imperfection shape: mode 1 **********'
      IF (ILOADX.GT.2.AND.NPRINX.GT.0) WRITE(IFILE8,'(/,/,A)')
     1' ********* Buckling modal imperfection shape: mode 3 **********'
C
      IALL = 0
      DO 33 ISEG = 1,NSEG
         IBEG = IALL + 1
         IEND = IALL + I5(ISEG)
         IF (ILOADX.LE.2.AND.NPRINX.GT.0)
     1   WRITE(IFILE8,'(/,A,I3,A,/,(1P,5E12.4))')
     1  ' Buckling mode 1 imperfection in Segment no.',ISEG,' WSAVEX=',
     1   (WSAVEX(I),I=IBEG,IEND)
         IF (ILOADX.GT.2.AND.NPRINX.GT.0)
     1   WRITE(IFILE8,'(/,A,I3,A,/,(1P,5E12.4))')
     1  ' Buckling mode 3 imperfection in Segment no.',ISEG,' WSAVEX=',
     1   (WSAVEX(I),I=IBEG,IEND)
         IALL = IALL + I5(ISEG)
   33 CONTINUE
```

```
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      DO 3300 I = 1,I2
         DIFF = ABS(1. - ABS(WMODX2(I)))
         IF (DIFF.LT.0.00001) THEN
            FSIGN2 = WMODX2(I)/ABS(WMODX2(I))
            IF (IMODX.EQ.0) THEN
               NODESV2 = I
               FSIGN2S = FSIGN2
            ENDIF
            IF (IMODX.EQ.1) THEN
               SIGNA = FSIGN2
               SIGNB = WMODX2(NODESV2)/ABS(WMODX2(NODESV2))
               FPROD = SIGNA*SIGNB
               IF (FPROD.LT.0.) FSIGN2 = SIGNB
               IF (ILOADX.GT.2) THEN
                  FPROD = FSIGN2S*FSIGN2
                  IF (FPROD.LT.0.) THEN
                     DO 3275 J = 1,I2
                        WMODX2(J) = -1.*WMODX2(J)
 3275                CONTINUE
                  ENDIF
                  GO TO 3375
               ENDIF
            ENDIF
            GO TO 3300
         ENDIF
 3300  CONTINUE
      IF (IMODX.EQ.0) THEN
         FSIGNMX2 = FSIGN2
         WSAVE2 = WMODX2(1)
         IF (ABS(WSAVE2).GE.0.7) THEN
            FSIGN2S = WSAVE2/ABS(WSAVE2)
         ELSE
            FSIGN2S = FSIGN2
         ENDIF
      ENDIF
      IF (IMODX.EQ.1) THEN
         FSIGN2SS = FSIGN2S
         IF (ABS(WSAVE2).GE.0.7) THEN
            COMPAR = 0.7
            IF (ABS(WSAVE2).GE.0.99999) COMPAR = 0.1
            IF (ABS(WMODX2(1)).GE.COMPAR) THEN
               FSIGN2 = WMODX2(1)/ABS(WMODX2(1))
            ELSE
               FSIGN2S = FSIGNMX2
            ENDIF
         ENDIF
```

```
          IF (ABS(WSAVE2).LT.0.7.AND.ABS(WMODX2(1)).GE.0.7) THEN
             FSIGN2S = FSIGNMX2
             FSIGN2 = WMODX2(NODESV2)/ABS(WMODX2(NODESV2))
          ENDIF
          FPROD = FSIGN2S*FSIGN2
          IF (FPROD.LT.0.) THEN
             DO 3350 I = 1,I2
                WMODX2(I) = -1.*WMODX2(I)
 3350        CONTINUE
          ENDIF
          FSIGN2S = FSIGN2SS
       ENDIF
C      End of (IMODX.EQ.1) condition
 3375  CONTINUE
C
       IF (ILOADX.LE.2.AND.NPRINX.GT.0) WRITE(IFILE8,'(/,/,A)')
     1' ********* Buckling modal imperfection shape: mode 2 **********'
       IF (ILOADX.GT.2.AND.NPRINX.GT.0) WRITE(IFILE8,'(/,/,A)')
     1' ********* Buckling modal imperfection shape: mode 4 **********'
C
       IALL = 0
       DO 34 ISEG = 1,NSEG
          IBEG = IALL + 1
          IEND = IALL + I5(ISEG)
          IF (ILOADX.LE.2.AND.NPRINX.GT.0)
     1    WRITE(IFILE8,'(/,A,I3,A,/,(1P,5E12.4))')
     1  ' Buckling mode 2 imperfection in Segment no.',ISEG,' WMODX2=',
     1    (WMODX2(I),I=IBEG,IEND)
          IF (ILOADX.GT.2.AND.NPRINX.GT.0)
     1    WRITE(IFILE8,'(/,A,I3,A,/,(1P,5E12.4))')
     1  ' Buckling mode 4 imperfection in Segment no.',ISEG,' WMODX2=',
     1    (WMODX2(I),I=IBEG,IEND)
          IALL = IALL + I5(ISEG)
   34  CONTINUE
C
      ENDIF
C     END OF (IMODX.EQ.0.OR.IFAST.EQ.0) SECTION
C
      CALL GASP (DUM1,DUM2,-2,DUM3)
C
C NOTE: If (ITYPEX.EQ.2), that is, analysis of a fixed design, SUBROUTINE
C       STRUCT automatically produces BIGBOSOR4 input files for each type
C       of analysis performed here. The files have the following names:
C
C       1. linear axisymmetric bifurcation buckling of perfect shell:
C          *.ALL1 (the "*" represents the user-provided specific case
name)
C
```

```
C       2. nonlinear axisymmetric stress analysis of mode 1 (and possibly
C          mode 3) imperfect shell:
C          *.ALL2P  and *.ALL2N  and possibly *.ALL3P  and *.ALL3N and
possibly
C          *.ALL2P3 and *.ALL2N3 and possibly *.ALL3P3 and *.ALL3N3
C        (Typical here and below: The "P" in *.ALL2P means "positive mode
x".
C         Typical here and below: The "N" in *.ALL2N means "negative mode
x".
C         Typical here and below; the ending "3" in *.ALL2P3 means "mode x
+ 2".
C
C       3. nonlinear axisymmetric stress analysis of mode 2 (and possibly
C          mode 4) imperfect shell:
C          *.ALL4P  and *.ALL4N  and possibly *.ALL5P  and *.ALL5N and
possibly
C          *.ALL4P3 and *.ALL4N3 and possibly *.ALL5P3 and *.ALL5N3
C
C       4. nonlinear axisymmetric collapse of mode 1 and possibly mode 3
C          imperfect shell:
C          *.ALL6P and *.ALL6N (mode 1) and possibly *.ALL6P3 and *.ALL6N3
(mode 3)
C
C       5. nonlinear axisymmetric collapse of mode 2 and possibly mode 4
C          imperfect shell:
C          *.ALL7P and *.ALL7N (mode 2) and possibly *.ALL7P3 and *.ALL7N3
(mode 4)
C
C       6. nonlinear bifurcation buckling of mode 1 and possibly mode 3
C          imperfect shell.
C          *.ALL8P and *.ALL8N (mode 1) and possibly *.ALL8P3 and *.ALL8N3
(mode 3)
C
C       7. nonlinear bifurcation buckling of mode 2 and possibly mode 4
C          imperfect shell.
C          *.ALL9P and *.ALL9N (mode 2) and possibly *.ALL9P3 and *.ALL9N3
(mode 4)
C
C In order to run BIGBOSOR4, do the following:
C 1. go to a directory where you want to run BIGBOSOR4.
C 2. copy one of the files *.ALL*  (Example: torispec3b.ALL6P)
C    to that directory.
C 3. change the name from *.ALL* to *.ALL
C    Example: cp torispec3b.ALL6P torispec3b.ALL
C 4. type the following BIGBOSOR4 commands:
C    bigbosor4log        (activates the BIGBOSOR4 set of commands)
C    bigbosorall         (executes BIGBOSOR4)
C    (inspect the BIGBOSOR4 output file, *.OUT)
```

```
C    bosorplot           (generates plots)
C
      IF (NCASES.EQ.2.OR.NCASES.EQ.4) THEN
C      The + and - mode shapes are treated as separate load cases.
        IF (ILOADX.EQ.2.OR.ILOADX.EQ.4) THEN
           M1MULT =  -1
           M1MULTC = -1
           M1MULTB = -1
           M2MULT =  -1
           M2MULTC = -1
           M2MULTB = -1
           IBACK1 = 1
           IBACK2 = 1
        ELSE
           M1MULT =   1
           M1MULTC =  1
           M1MULTB =  1
           M2MULT =   1
           M2MULTC =  1
           M2MULTB =  1
           IBACK1 = 0
           IBACK2 = 0
        ENDIF
      ENDIF
C
C  Find axisymmetric nonlinear equilibrium (INDIC=0) of imperfect shell
C  at the design load, PRESS(ILOADX), with use of axisymmetric buckling
C  modal imperfection mode 1.
C
      IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
      IF (IMODX.EQ.0) THEN
         WRITE(IFILE8,'(A,I2,A)')
     1' ============ Analysis No. 2 for Load Set No.',ILOADX,
     1' ============'
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,+(mode 1) imperfection',
     1'      IMODX=',IMODX
         IF (ILOADX.EQ.2) WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,-(mode 1) imperfection',
     1'      IMODX=',IMODX
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,+(mode 3) imperfection',
     1'      IMODX=',IMODX
         IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,-(mode 3) imperfection',
     1'      IMODX=',IMODX
      ELSE
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) THEN
```

```
            IF (M1MULT.EQ.1)  WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,+(mode 1) imperfection',
     1'      IMODX=',IMODX
            IF (M1MULT.EQ.-1) WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,-(mode 1) imperfection',
     1'      IMODX=',IMODX
          ENDIF
          IF (ILOADX.EQ.3.OR.ILOADX.EQ.4) THEN
            IF (M1MULT.EQ.1)  WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,+(mode 3) imperfection',
     1'      IMODX=',IMODX
            IF (M1MULT.EQ.-1) WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,-(mode 3) imperfection',
     1'      IMODX=',IMODX
          ENDIF
        ENDIF
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
        CALL MOVER(WSAVEX,1,WMODEX,1,I2)
        IF (IMODX.EQ.0) THEN
           IF (NCASES.EQ.1) THEN
              IBACK1 = 0
              M1MULT = 1
              M1MULTC = 1
           ENDIF
        ENDIF
     35 CONTINUE
        IF (M1MULT.EQ.-1) THEN
           DO 352 I = 1,I2
            WMODEX(I) = -WSAVEX(I)
    352    CONTINUE
        ENDIF
        INDIC = 0
        IMPERF = 1
        PMAX = PRESS(ILOADX)
        IF (IMODX.EQ.1) PMAX = PMAX01
        CALL BOSDEC(4,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C       Get CASE.ALL2 file for input for BIGBOSOR4...
C
C       CASE.ALL2P is an input file for BIGBOSOR4 for nonlinear stress
C       analysis of the axisymmetrically imperfect ellipsoidal shell
C       with a +(mode 1) imperfection shape. (Load set no. ILOADX=1)
C
C       CASE.ALL2N is an input file for BIGBOSOR4 for nonlinear stress
C       analysis of the axisymmetrically imperfect ellipsoidal shell
```

```
C      with a -(mode 1) imperfection shape. (Load set no. ILOADX=2)
C
C      CASE.ALL2P3 is an input file for BIGBOSOR4 for nonlinear stress
C      analysis of the axisymmetrically imperfect ellipsoidal shell
C      with a +(mode 3) imperfection shape. (Load set no. ILOADX=3)
C
C      CASE.ALL2N3 is an input file for BIGBOSOR4 for nonlinear stress
C      analysis of the axisymmetrically imperfect ellipsoidal shell
C      with a -(mode 3) imperfection shape. (Load set no. ILOADX=4)
C
       IF (ITYPEX.EQ.2) THEN
          I=INDEX(CASE,' ')
          IF (ILOADX.LE.2) THEN
             IF (IBACK1.EQ.0) THEN
                IF(I.NE.0) THEN
                   CASA2=CASE(:I-1)//'.ALL2P'
                ELSE
                   CASA2=CASE//'.ALL2P'
                ENDIF
             ELSE
                IF(I.NE.0) THEN
                   CASA2=CASE(:I-1)//'.ALL2N'
                ELSE
                   CASA2=CASE//'.ALL2N'
                ENDIF
             ENDIF
          ENDIF
          IF (ILOADX.GT.2) THEN
             IF (IBACK1.EQ.0) THEN
                IF(I.NE.0) THEN
                   CASA2=CASE(:I-1)//'.ALL2P3'
                ELSE
                   CASA2=CASE//'.ALL2P3'
                ENDIF
             ELSE
                IF(I.NE.0) THEN
                   CASA2=CASE(:I-1)//'.ALL2N3'
                ELSE
                   CASA2=CASE//'.ALL2N3'
                ENDIF
             ENDIF
          ENDIF
          OPEN(UNIT=62,FILE=CASA2,STATUS='UNKNOWN')
          CALL BOSDEC(4,ILOADX,INDIC,IMPERF,62,IFILE8,
     1                npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
```

```
         CLOSE(UNIT=62)
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
         IF (ILOADX.LE.2) THEN
            IF (IBACK1.EQ.0) THEN
               WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,+(mode 1) imperfect=',
     1         CASA2
            ELSE
               WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,-(mode 1) imperfect=',
     1         CASA2
            ENDIF
         ENDIF
         IF (ILOADX.GT.2) THEN
            IF (IBACK1.EQ.0) THEN
               WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,+(mode 3) imperfect=',
     1         CASA2
            ELSE
               WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,-(mode 3) imperfect=',
     1         CASA2
            ENDIF
         ENDIF
       ENDIF
C      End of (ITYPEX.EQ.2) condition
C
C      IF (IDV.EQ.1.AND.IMODX.EQ.1) THEN
C         ITESTX = 1
C         WRITE(IFILE8,'(A,I3)')' ITESTX=',ITESTX
C      ENDIF
       CALL B4READ
       CALL B4MAIN
c
       bskin1 = 10.E+16
       bstif1 = 10.E+16
       bskin2 = 10.E+16
       bstif2 = 10.E+16
       sknmx1 = 0.
       stfmx1 = 0.
       sknmx2 = 0.
       stfmx2 = 0.
       do 363 iseg = 1,NSEG
          ipoint = iseg + 1
          if (xinput(ipoint).lt.xlimit) then
             bskin1 = min(bskin1,BUCMIN(iseg))
             bstif1 = min(bstif1,BUCMNS(iseg))
             sknmx1 = max(sknmx1,SKNMAX(iseg))
```

```
            stfmx1 = max(stfmx1,STFMXS(iseg))
         else
            bskin2 = min(bskin2,BUCMIN(iseg))
            bstif2 = min(bstif2,BUCMNS(iseg))
            sknmx2 = max(sknmx2,SKNMAX(iseg))
            stfmx2 = max(stfmx2,STFMXS(iseg))
         endif
  363 continue
c
      IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
      IF (NPRINX.GT.0) THEN
        IF (ILOADX.LE.2) WRITE(IFILE8,'(/,A,I2,A)')
     1 ' *** Output from mode 1 INDIC=0, stress analysis; IMODX=',
     1   IMODX,' ***'
         IF (ILOADX.GT.2) WRITE(IFILE8,'(/,A,I2,A)')
     1 ' *** Output from mode 3 INDIC=0, stress analysis; IMODX=',
     1   IMODX,' ***'
         WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' Pressure multiplier, P, for all load steps=',
     1   (PSTEP(I),I=1,ISTEP)
         WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' End displacement, ENDUVS, for all load steps=',
     1   (ENDUVS(I),I=1,ISTEP)
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
         DO 36 I = 1,NSEG
            WRITE(IFILE8,'(/,A,I4)')
     1   ' Local skin and smeared stiffener buckling and stress, Seg.',I
            WRITE(IFILE8,'(A,1P,E12.4,A,I3,/,A,1P,E12.4,A,I3)')
     1    ' Skin buckling load factor,                    BUCMIN=',
     1      BUCMIN(I),' at nodal point',IBUCMN(I),
     1    ' Smeared stringer/isogrid buckling load factor, BUCMNS=',
     1      BUCMNS(I),' at nodal point',IBUCMS(I)
            WRITE(IFILE8,'(A,1P,E12.4,A,I3,/,A,1P,E12.4,A,I3)')
     1    ' Smeared ring buckling load factor,            BUCMNR=',
     1      BUCMNR(I),' at nodal point',IBUCMR(I),
     1    ' Smeared stringer/isogrid maximum eff. stress,  STFMXS=',
     1      STFMXS(I),' at nodal point',ISTFMS(I)
            WRITE(IFILE8,'(A,1P,E12.4,A,I3,/,A,1P,E12.4,A,I3)')
     1    ' Smeared ring maximum effective stress,        STFMXR=',
     1      STFMXR(I),' at nodal point',ISTFMR(I),
     1    ' Shell skin maximum effective stress,          SKNMAX=',
     1      SKNMAX(I),' at nodal point',ISKNMX(I)
   36    CONTINUE
c
         IF (IMODX.EQ.0) THEN
            BUCSKU1(1) = bskin1
            BUCSTU1(1) = bstif1
```

```
            STRMAU1(1) = sknmx1
            STRSTU1(1) = stfmx1
            BUCSKU1(2) = bskin2
            BUCSTU1(2) = bstif2
            STRMAU1(2) = sknmx2
            STRSTU1(2) = stfmx2
         ENDIF
C
         IF (IMODX.EQ.0) THEN
            ENDUVU1= ENDUV
         ENDIF
         WRITE(IFILE8,'(/,A,A)')
     1 '                                                              PERTURBED',
     1 '  UNPERTURBED'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
         WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1 ' Region 1 skin buckling load factor,        bskin1=',
     1   bskin1,BUCSKU1(1),
     1 ' Region 1 stiffener buckling load factor,   bstif1=',
     1   bstif1,BUCSTU1(1)
         WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1 ' Region 1 skin maximum effective stress,    sknmx1=',
     1   sknmx1,STRMAU1(1),
     1 ' Region 1 stiffener max. effective stress,  stfmx1=',
     1   stfmx1,STRSTU1(1)
         WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1 ' Region 2 skin buckling load factor,        bskin2=',
     1   bskin2,BUCSKU1(2),
     1 ' Region 2 stiffener buckling load factor,   bstif2=',
     1   bstif2,BUCSTU1(2)
         WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1 ' Region 2 skin maximum effective stress,    sknmx2=',
     1   sknmx2,STRMAU1(2),
     1 ' Region 2 stiffener max. effective stress,  stfmx2=',
     1   stfmx2,STRSTU1(2)
         WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Normal displacement of shell at apex,       ENDUV=',
     1   ENDUV,ENDUVU1
       ENDIF
C      End of (NPRINX.GT.0) condition
       IF (IMODX.EQ.0) THEN
         PMAX01 = PRESS(ILOADX)
         PSTEP1 = PSTEP(ISTEP)
         IF (NCASES.EQ.1.AND.IBACK1.EQ.0) PSTEP1SV = PSTEP1
         IF (NCASES.EQ.1.AND.IBACK1.EQ.1) THEN
            IF (PSTEP1.LT.0.9999*PSTEP1SV) THEN
               M1MULT = -1
               PMAX01 = PSTEP1
```

```
               PMAX01M= PSTEP1
               CALL GASP (DUM1,DUM2,-2,DUM3)
               GO TO 396
            ENDIF
C
            IF (PSTEP1SV.LT.0.9999*PSTEP1) THEN
               M1MULT = 1
               PMAX01 = PSTEP1SV
               PMAX01M = PSTEP1
               BUCSKU1(1)= BUCSKU1S(1)
               BUCSTU1(1)= BUCSTU1S(1)
               STRMAU1(1)= STRMAU1S(1)
               STRSTU1(1)= STRSTU1S(1)
               BUCSKU1(2)= BUCSKU1S(2)
               BUCSTU1(2)= BUCSTU1S(2)
               STRMAU1(2)= STRMAU1S(2)
               STRSTU1(2)= STRSTU1S(2)
               ENDUVU1= ENDUVU1S
               bskin1 = bskin1s
               bstif1 = bstif1s
               sknmx1 = sknmx1s
               stfmx1 = stfmx1s
               bskin2 = bskin2s
               bstif2 = bstif2s
               sknmx2 = sknmx2s
               stfmx2 = stfmx2s
               ENDUV  = ENDUVSV
               PMAX01 = PMAX01S
               CALL GASP (DUM1,DUM2,-2,DUM3)
               GO TO 396
            ENDIF
         ENDIF
C        End of (NCASES.EQ.1.AND.IBACK1.EQ.1) condition
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
         IF (PSTEP1.LT.PRESS(ILOADX)) THEN
           WRITE(IFILE8,'(/,A,/,A,/,A,1P,E12.4,A,/,A,/,A,/,A)')
     1   ' ****** WARNING ******* WARNING ******** WARNING *********',
     1   ' For the current design the imperfect shell collapses',
     1   ' at a pressure that is lower than the design pressure,',
     1     PRESS(ILOADX),'.',
     1   ' In order to avoid erratic and incorrect gradients of',
     1   ' the stress and local buckling constraints, we must',
     1   ' redo the nonlinear stress analysis with a new maximum'
           WRITE(IFILE8,'(A,1P,E12.4,A,/,A,/,A,/,A,/)')
     1   ' pressure,',PSTEP1,'. This new maximum also affects the',
     1   ' starting pressure and the pressure increment used in the',
     1   ' nonlinear stress, collapse, and buckling analyses.',
     1   ' *** END WARNING **** END WARNING **** END WARNING *******'
```

```
C
          IF (ILOADX.LE.2) WRITE(IFILE8,'(/,A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,   mode 1 imperfection',
     1'      IMODX=',IMODX
          IF (ILOADX.GT.2) WRITE(IFILE8,'(/,A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,   mode 3 imperfection',
     1'      IMODX=',IMODX
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
          CALL GASP (DUM1,DUM2,-2,DUM3)
          PMAX01 = PSTEP1
          PMAX  = PMAX01
          CALL BOSDEC(4,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C         Get CASE.ALL3 file for input for BIGBOSOR4...
C
C         CASE.ALL3P is an input file for BIGBOSOR4 for nonlinear stress
C         analysis of the axisymmetrically imperfect ellipsoidal shell
C         with a +(mode 1) imperfection shape (Load set no. ILOADX=1)
C         in which P(COLLAPSE) is less than P(design).
C
C         CASE.ALL3N is an input file for BIGBOSOR4 for nonlinear stress
C         analysis of the axisymmetrically imperfect ellipsoidal shell
C         with a -(mode 1) imperfection shape (Load set no. ILOADX=2)
C         in which P(COLLAPSE) is less than P(design).
C
C         CASE.ALL3P3 is an input file for BIGBOSOR4 for nonlinear stress
C         analysis of the axisymmetrically imperfect ellipsoidal shell
C         with a +(mode 3) imperfection shape (Load set no. ILOADX=3)
C         in which P(COLLAPSE) is less than P(design).
C
C         CASE.ALL3N3 is an input file for BIGBOSOR4 for nonlinear stress
C         analysis of the axisymmetrically imperfect ellipsoidal shell
C         with a -(mode 3) imperfection shape (Load set no. ILOADX=4)
C         in which P(COLLAPSE) is less than P(design).
C
          IF (ITYPEX.EQ.2) THEN
             I=INDEX(CASE,' ')
             IF (ILOADX.LE.2) THEN
                IF (IBACK1.EQ.0) THEN
                   IF(I.NE.0) THEN
                      CASA3=CASE(:I-1)//'.ALL3P'
                   ELSE
                      CASA3=CASE//'.ALL3P'
                   ENDIF
                ELSE
```

```
                  IF(I.NE.0) THEN
                     CASA3=CASE(:I-1)//'.ALL3N'
                  ELSE
                     CASA3=CASE//'.ALL3N'
                  ENDIF
               ENDIF
            ENDIF
            IF (ILOADX.GT.2) THEN
               IF (IBACK1.EQ.0) THEN
                  IF(I.NE.0) THEN
                     CASA3=CASE(:I-1)//'.ALL3P3'
                  ELSE
                     CASA3=CASE//'.ALL3P3'
                  ENDIF
               ELSE
                  IF(I.NE.0) THEN
                     CASA3=CASE(:I-1)//'.ALL3N3'
                  ELSE
                     CASA3=CASE//'.ALL3N3'
                  ENDIF
               ENDIF
            ENDIF
            OPEN(UNIT=63,FILE=CASA3,STATUS='UNKNOWN')
            CALL BOSDEC(4,ILOADX,INDIC,IMPERF,63,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
            CLOSE(UNIT=63)
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
            IF (ILOADX.LE.2) THEN
               IF (IBACK1.EQ.0) THEN
                  WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,+(mode 1) imperfect=',
     1            CASA3
               ELSE
                  WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,-(mode 1) imperfect=',
     1            CASA3
               ENDIF
            ENDIF
            IF (ILOADX.GT.2) THEN
               IF (IBACK1.EQ.0) THEN
                  WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,+(mode 3) imperfect=',
     1            CASA3
               ELSE
                  WRITE(IFILE8,'(A)')
```

```
     1' BIGBOSOR4 input file for nonlinear stress,-(mode 3) imperfect=',
     1             CASA3
                ENDIF
             ENDIF
          ENDIF
C         End of (ITYPEX.EQ.2) condition
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
          CALL B4READ
          CALL B4MAIN
c
          bskin1 = 10.E+16
          bstif1 = 10.E+16
          bskin2 = 10.E+16
          bstif2 = 10.E+16
          sknmx1 = 0.
          stfmx1 = 0.
          sknmx2 = 0.
          stfmx2 = 0.
          do 383 iseg = 1,NSEG
             ipoint = iseg + 1
             if (xinput(ipoint).lt.xlimit) then
                bskin1 = min(bskin1,BUCMIN(iseg))
                bstif1 = min(bstif1,BUCMNS(iseg))
                sknmx1 = max(sknmx1,SKNMAX(iseg))
                stfmx1 = max(stfmx1,STFMXS(iseg))
             else
                bskin2 = min(bskin2,BUCMIN(iseg))
                bstif2 = min(bstif2,BUCMNS(iseg))
                sknmx2 = max(sknmx2,SKNMAX(iseg))
                stfmx2 = max(stfmx2,STFMXS(iseg))
             endif
  383     continue
c
          IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
          IF (PSTEP(ISTEP).LT.PRESS(ILOADX)) THEN
             ISKIP(JJJ1) = 1
             PMAX01 = PSTEP(ISTEP)
          ENDIF
          IF (NPRINX.GT.0) THEN
            IF (ILOADX.LE.2) WRITE(IFILE8,'(/,A,I2,A)')
     1    ' *** Output from mode 1 INDIC=0, stress analysis; IMODX=',
     1      IMODX,' ***'
            IF (ILOADX.GT.2) WRITE(IFILE8,'(/,A,I2,A)')
     1    ' *** Output from mode 3 INDIC=0, stress analysis; IMODX=',
     1      IMODX,' ***'
            WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1    ' Pressure multiplier, P, for all load steps=',
```

```
     1      (PSTEP(I),I=1,ISTEP)
            WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1     ' End displacement, ENDUVS, for all load steps=',
     1      (ENDUVS(I),I=1,ISTEP)
C2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
            DO 38 I = 1,NSEG
               WRITE(IFILE8,'(/,A,I4)')
     1    ' Local skin and smeared stiffener buckling and stress, Seg.',I
               WRITE(IFILE8,'(A,1P,E12.4,A,I3,/,A,1P,E12.4,A,I3)')
     1       ' Skin buckling load factor,                    BUCMIN=',
     1         BUCMIN(I),' at nodal point',IBUCMN(I),
     1       ' Smeared stringer/isogrid buckling load factor, BUCMNS=',
     1         BUCMNS(I),' at nodal point',IBUCMS(I)
               WRITE(IFILE8,'(A,1P,E12.4,A,I3,/,A,1P,E12.4,A,I3)')
     1       ' Smeared ring buckling load factor,            BUCMNR=',
     1         BUCMNR(I),' at nodal point',IBUCMR(I),
     1       ' Smeared stringer/isogrid maximum eff. stress,  STFMXS=',
     1         STFMXS(I),' at nodal point',ISTFMS(I)
               WRITE(IFILE8,'(A,1P,E12.4,A,I3,/,A,1P,E12.4,A,I3)')
     1       ' Smeared ring maximum effective stress,        STFMXR=',
     1         STFMXR(I),' at nodal point',ISTFMR(I),
     1       ' Shell skin maximum effective stress,          SKNMAX=',
     1         SKNMAX(I),' at nodal point',ISKNMX(I)
   38       CONTINUE
c
            BUCSKU1(1) = bskin1
            BUCSTU1(1) = bstif1
            STRMAU1(1) = sknmx1
            STRSTU1(1) = stfmx1
            BUCSKU1(2) = bskin2
            BUCSTU1(2) = bstif2
            STRMAU1(2) = sknmx2
            STRSTU1(2) = stfmx2
C
            ENDUVU1= ENDUV
            WRITE(IFILE8,'(/,A,A)')
     1 '                                                    PERTURBED',
     1    '  UNPERTURBED'
            WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1    ' Region 1 skin buckling load factor,        bskin1=',
     1      bskin1,BUCSKU1(1),
     1    ' Region 1 stiffener buckling load factor,   bstif1=',
     1      bstif1,BUCSTU1(1)
            WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1    ' Region 1 skin maximum effective stress,    sknmx1=',
     1      sknmx1,STRMAU1(1),
     1    ' Region 1 stiffener max. effective stress,  stfmx1=',
```

```
     1      stfmx1,STRSTU1(1)
            WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1     ' Region 2 skin buckling load factor,        bskin2=',
     1      bskin2,BUCSKU1(2),
     1     ' Region 2 stiffener buckling load factor,   bstif2=',
     1      bstif2,BUCSTU1(2)
            WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1     ' Region 2 skin maximum effective stress,    sknmx2=',
     1      sknmx2,STRMAU1(2),
     1     ' Region 2 stiffener max. effective stress,  stfmx2=',
     1      stfmx2,STRSTU1(2)
            WRITE(IFILE8,'(A,1P,2E12.4)')
     1     ' Normal displacement of shell at apex,       ENDUV=',
     1      ENDUV,ENDUVU1
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
          ENDIF
C         End of IF (NPRINX.GT.0) condition
        ENDIF
C       End of IF (PSTEP1.LT.PRESS(ILOADX)) condition
      ENDIF
C     End of IF (IMODX.EQ.0) condition
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
C     CALL B4POST
      CALL GASP (DUM1,DUM2,-2,DUM3)
C
      IF (NCASES.EQ.2.OR.NCASES.EQ.4) THEN
         PMAX01P = PMAX01
         PMAX01M = PMAX01
      ENDIF
      IF (NCASES.EQ.1.AND.IMODX.EQ.0.AND.IBACK1.EQ.0) THEN
         FMARG(1) = bskin1 /(SKNBK1A(ILOADX,1)*SKNBK1F(ILOADX,1))-1.0
         FMARG(2) = bstif1 /(STFBK1A(ILOADX,1)*STFBK1F(ILOADX,1))-1.0
         FMARG(3) = SKNST1A(ILOADX,1)/(sknmx1 *SKNST1F(ILOADX,1))-1.0
         FMARG(4) = STFST1A(ILOADX,1)/(stfmx1 *STFST1F(ILOADX,1))-1.0
         FMARG(5) = WAPEX1A(ILOADX)/(ABS(ENDUV)*WAPEX1F(ILOADX))-1.0
         FMARG(6) = bskin2 /(SKNBK1A(ILOADX,2)*SKNBK1F(ILOADX,2))-1.0
         FMARG(7) = bstif2 /(STFBK1A(ILOADX,2)*STFBK1F(ILOADX,2))-1.0
         FMARG(8) = SKNST1A(ILOADX,2)/(sknmx2 *SKNST1F(ILOADX,2))-1.0
         FMARG(9) = STFST1A(ILOADX,2)/(stfmx2 *STFST1F(ILOADX,2))-1.0
         FMARG(10)= WAPEX1A(ILOADX)/(ABS(ENDUV)*WAPEX1F(ILOADX))-1.0
         FMARMIN = 100000.
         DO 385 I = 1,10
            FMARMIN = MIN(FMARMIN,FMARG(I))
  385    CONTINUE
         IF (FMARMIN.GT.2.0) THEN
            PMAX01P = PMAX01
            PMAX01M = PMAX01
```

```
            WRITE(IFILE8,'(/,A,/,A,1P,E12.4)')
     1     ' -(mode 1) not investigated because the minimum margin for',
     1     ' +(mode 1) > 2. The minimum margin for +(mode 1), FMARMIN=',
     1       FMARMIN
            GO TO 396
          ENDIF
C
C       Check for negative of mode 1...
          BUCSKU1S(1)= BUCSKU1(1)
          BUCSTU1S(1)= BUCSTU1(1)
          STRMAU1S(1)= STRMAU1(1)
          STRSTU1S(1)= STRSTU1(1)
          BUCSKU1S(2)= BUCSKU1(2)
          BUCSTU1S(2)= BUCSTU1(2)
          STRMAU1S(2)= STRMAU1(2)
          STRSTU1S(2)= STRSTU1(2)
          ENDUVU1S= ENDUVU1
          bskin1s = bskin1
          bstif1s = bstif1
          sknmx1s = sknmx1
          stfmx1s = stfmx1
          bskin2s = bskin2
          bstif2s = bstif2
          sknmx2s = sknmx2
          stfmx2s = stfmx2
          ENDUVSV = ABS(ENDUV)
          PMAX01S = PMAX01
          DO 39 I = 1,I2
           WMODEX(I) = -WSAVEX(I)
   39     CONTINUE
          WRITE(IFILE8,'(/,/,A)')
     1  ' Check for imperfection shape with negative of mode 1'
          WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,-(mode 1) imperfection',
     1'      IMODX=',IMODX
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
          IBACK1 = 1
          PMAX01P = PMAX01
          GO TO 35
        ENDIF
C       End of (NCASES.EQ.1.AND.IMODX.EQ.0.AND.IBACK1.EQ.0) condition
C
        IF (NCASES.EQ.1.AND.IMODX.EQ.0.AND.IBACK1.EQ.1) THEN
          PMAX01M = PMAX01
C       Decide which mode 1 (+ or -) to use for margins
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
          FMARGP(1) =bskin1s/(SKNBK1A(ILOADX,1)*SKNBK1F(ILOADX,1))-1.0
          FMARGM(1) =bskin1 /(SKNBK1A(ILOADX,1)*SKNBK1F(ILOADX,1))-1.0
```

```
      FMARGP(2) =bstif1s/(STFBK1A(ILOADX,1)*STFBK1F(ILOADX,1))-1.0
      FMARGM(2) =bstif1 /(STFBK1A(ILOADX,1)*STFBK1F(ILOADX,1))-1.0
      FMARGP(3) =SKNST1A(ILOADX,1)/(sknmx1s*SKNST1F(ILOADX,1))-1.0
      FMARGM(3) =SKNST1A(ILOADX,1)/(sknmx1 *SKNST1F(ILOADX,1))-1.0
      FMARGP(4) =STFST1A(ILOADX,1)/(stfmx1s*STFST1F(ILOADX,1))-1.0
      FMARGM(4) =STFST1A(ILOADX,1)/(stfmx1 *STFST1F(ILOADX,1))-1.0
      FMARGP(5) = WAPEX1A(ILOADX)/(ENDUVSV*WAPEX1F(ILOADX)) -1.0
      FMARGM(5) = WAPEX1A(ILOADX)/(ABS(ENDUV)*WAPEX1F(ILOADX))-1.0
      FMARGP(6) =bskin2s/(SKNBK1A(ILOADX,2)*SKNBK1F(ILOADX,2))-1.0
      FMARGM(6) =bskin2 /(SKNBK1A(ILOADX,2)*SKNBK1F(ILOADX,2))-1.0
      FMARGP(7) =bstif2s/(STFBK1A(ILOADX,2)*STFBK1F(ILOADX,2))-1.0
      FMARGM(7) =bstif2 /(STFBK1A(ILOADX,2)*STFBK1F(ILOADX,2))-1.0
      FMARGP(8) =SKNST1A(ILOADX,2)/(sknmx2s*SKNST1F(ILOADX,2))-1.0
      FMARGM(8) =SKNST1A(ILOADX,2)/(sknmx2 *SKNST1F(ILOADX,2))-1.0
      FMARGP(9) =STFST1A(ILOADX,2)/(stfmx2s*STFST1F(ILOADX,2))-1.0
      FMARGM(9) =STFST1A(ILOADX,2)/(stfmx2 *STFST1F(ILOADX,2))-1.0
      FMARGP(10)= WAPEX1A(ILOADX)/(ENDUVSV*WAPEX1F(ILOADX)) -1.0
      FMARGM(10)= WAPEX1A(ILOADX)/(ABS(ENDUV)*WAPEX1F(ILOADX))-1.0
C23456789012345678901234567890123456789012345678901234567890123456789012
      M1MULT = 1
      FMARMNP = 100000.
      FMARMNM = 100000.
      DO 392 I = 1,10
         FMARMNP = MIN(FMARMNP,FMARGP(I))
         FMARMNM = MIN(FMARMNM,FMARGM(I))
  392 CONTINUE
      IF (FMARMNM.GT.FMARMNP.OR.FMARMNM.GT.2.0) THEN
         BUCSKU1(1)= BUCSKU1S(1)
         BUCSTU1(1)= BUCSTU1S(1)
         STRMAU1(1)= STRMAU1S(1)
         STRSTU1(1)= STRSTU1S(1)
         BUCSKU1(2)= BUCSKU1S(2)
         BUCSTU1(2)= BUCSTU1S(2)
         STRMAU1(2)= STRMAU1S(2)
         STRSTU1(2)= STRSTU1S(2)
         ENDUVU1= ENDUVU1S
         bskin1 = bskin1s
         bstif1 = bstif1s
         sknmx1 = sknmx1s
         stfmx1 = stfmx1s
         bskin2 = bskin2s
         bstif2 = bstif2s
         sknmx2 = sknmx2s
         stfmx2 = stfmx2s
         ENDUV  = ENDUVSV
         PMAX01 = PMAX01S
         GO TO 396
      ENDIF
```

```
         DO 393 I = 1,10
            DIFF = 1.0
            IF (FMARMNM.NE.0.)
     1         DIFF = ABS(FMARMNM - FMARGM(I))/ABS(FMARMNM)
            IF (DIFF.LT.0.0001) THEN
               K = I
               IF (FMARGM(K).LT.FMARGP(K)) THEN
                  M1MULT = -1
                  GO TO 396
               ENDIF
            ENDIF
  393    CONTINUE
       ENDIF
C      End of (NCASES.EQ.1.AND.IMODX.EQ.0.AND.IBACK1.EQ.1) condition
  396 CONTINUE
C23456789012345678901234567890123456789012345678901234567890123456789012
       IF (NCASES.EQ.1.AND.IMODX.EQ.0) THEN
          IF (M1MULT.EQ.1) THEN
              WRITE(IFILE8,'(/,A)')
     1     ' **** The positive mode 1 is more critical than the negative'
          ENDIF
          IF (M1MULT.EQ.-1) THEN
              WRITE(IFILE8,'(/,A)')
     1     ' **** The negative mode 1 is more critical than the positive'
          ENDIF
          M1MULTC = 1
          IF (PMAX01M.LT.PMAX01P) M1MULTC = -1
          IF (M1MULTC.EQ.1) THEN
              PMAX01 = PMAX01P
C             WRITE(IFILE8,'(/,A,/)')
C    1     ' *** The positive mode 1 is to be used for collapse analysis'
          ENDIF
          IF (M1MULTC.EQ.-1) THEN
              PMAX01 = PMAX01M
C             WRITE(IFILE8,'(/,A,/)')
C    1     ' *** The negative mode 1 is to be used for collapse analysis'
          ENDIF
       ENDIF
C      End of (NCASES.EQ.1.AND.IMODX.EQ.0) condition
C
       IF (NPRINX.GE.1) THEN
          WRITE(IFILE8,'(/,A,A)')
     1 ' The following quantities are used to generate behavioral',
     1 ' constraint conditions and margins:'
          WRITE(IFILE8,'(A,A)')
     1 '                                                    PERTURBED',
     1  '  UNPERTURBED'
          WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
```

```
     1 ' Region 1 skin buckling load factor,        bskin1=',
     1   bskin1,BUCSKU1(1),
     1 ' Region 1 stiffener buckling load factor,   bstif1=',
     1   bstif1,BUCSTU1(1)
         WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1 ' Region 1 skin maximum effective stress,    sknmx1=',
     1   sknmx1,STRMAU1(1),
     1 ' Region 1 stiffener max. effective stress,  stfmx1=',
     1   stfmx1,STRSTU1(1)
         WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1 ' Region 2 skin buckling load factor,        bskin2=',
     1   bskin2,BUCSKU1(2),
     1 ' Region 2 stiffener buckling load factor,   bstif2=',
     1   bstif2,BUCSTU1(2)
         WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1 ' Region 2 skin maximum effective stress,    sknmx2=',
     1   sknmx2,STRMAU1(2),
     1 ' Region 2 stiffener max. effective stress,  stfmx2=',
     1   stfmx2,STRSTU1(2)
         WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Normal displacement of shell at apex,       ENDUV=',
     1   ENDUV,ENDUVU1

      ENDIF
C     End of (NPRINX.GT.0) condition
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (NPRINX.GE.1.AND.PMAX01.LT.0.90*PRESS(ILOADX)) THEN
         WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,1P,E12.4,/,A,1P,E12.4)')
     1 ' ***** NOTE ***** NOTE ***** NOTE ***** NOTE ***** NOTE *****',
     1 ' WE DO NOT USE LOCAL SKIN AND STIFFENER BUCKLING AND STRESS',
     1 ' CONSTRAINTS, NOR APEX DEFLECTION CONSTRAINT BECAUSE THE',
     1 ' SHELL COLLAPSES AT A LOAD THAT IS LESS THAN 90 PER CENT OF',
     1 ' THE DESIGN LOAD:',
     1 ' Collapse pressure, PMAX01=',PMAX01,
     1 ' Design   pressure, PRESS =',PRESS(ILOADX)
      ENDIF
      IF (PMAX01.GE.0.90*PRESS(ILOADX)) THEN
         SKNBK1(ILOADX,1) = bskin1
         STFBK1(ILOADX,1) = bstif1
         SKNST1(ILOADX,1) = sknmx1
         STFST1(ILOADX,1) = stfmx1
         SKNBK1(ILOADX,2) = bskin2
         STFBK1(ILOADX,2) = bstif2
         SKNST1(ILOADX,2) = sknmx2
         STFST1(ILOADX,2) = stfmx2
         WAPEX1(ILOADX) = ABS(ENDUV)
      ENDIF
C
```

```
C     CALL EXIT
C
C  Find axisymmetric nonlinear equilibrium (INDIC=0) of imperfect shell
C  at the design load, PRESS(ILOADX), with use of axisymmetric buckling
C  modal imperfection mode 2.
C
      IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
      IF (IMODX.EQ.0) THEN
         WRITE(IFILE8,'(A,I2,A)')
     1' ============ Analysis No. 3 for Load Set No.',ILOADX,
     1' ============'
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,+(mode 2) imperfection',
     1'      IMODX=',IMODX
         IF (ILOADX.EQ.2) WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,-(mode 2) imperfection',
     1'      IMODX=',IMODX
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,+(mode 4) imperfection',
     1'      IMODX=',IMODX
         IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,-(mode 4) imperfection',
     1'      IMODX=',IMODX
      ELSE
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) THEN
           IF (M2MULT.EQ.1)  WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,+(mode 2) imperfection',
     1'      IMODX=',IMODX
           IF (M2MULT.EQ.-1) WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,-(mode 2) imperfection',
     1'      IMODX=',IMODX
         ENDIF
         IF (ILOADX.EQ.3.OR.ILOADX.EQ.4) THEN
           IF (M2MULT.EQ.1)  WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,+(mode 4) imperfection',
     1'      IMODX=',IMODX
           IF (M2MULT.EQ.-1) WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,-(mode 4) imperfection',
     1'      IMODX=',IMODX
         ENDIF
      ENDIF
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      CALL MOVER(WMODX2,1,WMODEX,1,I2)
      IF (IMODX.EQ.0) THEN
         IF (NCASES.EQ.1) THEN
            IBACK2 = 0
            M2MULT = 1
            M2MULTC = 1
```

```
         ENDIF
       ENDIF
  398 CONTINUE
       IF (M2MULT.EQ.-1) THEN
          DO 399 I = 1,I2
           WMODEX(I) = -WMODX2(I)
  399     CONTINUE
       ENDIF
       INDIC = 0
       IMPERF = 1
       PMAX = PRESS(ILOADX)
       IF (IMODX.EQ.1) PMAX = PMAX02
       CALL BOSDEC(4,ILOADX,INDIC,IMPERF,24,IFILE8,
      1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
      1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
      1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
      1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C       Get CASE.ALL4  file for input for BIGBOSOR4...
C
C       CASE.ALL4P is an input file for BIGBOSOR4 for nonlinear stress
C       analysis of the axisymmetrically imperfect ellipsoidal shell
C       with a +(mode 2) imperfection shape. (Load set no. ILOADX=1)
C
C       CASE.ALL4N is an input file for BIGBOSOR4 for nonlinear stress
C       analysis of the axisymmetrically imperfect ellipsoidal shell
C       with a -(mode 2) imperfection shape. (Load set no. ILOADX=2)
C
C       CASE.ALL4P3 is an input file for BIGBOSOR4 for nonlinear stress
C       analysis of the axisymmetrically imperfect ellipsoidal shell
C       with a +(mode 4) imperfection shape. (Load set no. ILOADX=3)
C
C       CASE.ALL4N3 is an input file for BIGBOSOR4 for nonlinear stress
C       analysis of the axisymmetrically imperfect ellipsoidal shell
C       with a -(mode 4) imperfection shape. (Load set no. ILOADX=4)
C
       IF (ITYPEX.EQ.2) THEN
          I=INDEX(CASE,' ')
          IF (ILOADX.LE.2) THEN
             IF (IBACK2.EQ.0) THEN
                IF(I.NE.0) THEN
                   CASA4=CASE(:I-1)//'.ALL4P'
                ELSE
                   CASA4=CASE//'.ALL4P'
                ENDIF
             ELSE
                IF(I.NE.0) THEN
                   CASA4=CASE(:I-1)//'.ALL4N'
                ELSE
```

```
               CASA4=CASE//'.ALL4N'
            ENDIF
         ENDIF
      ENDIF
      IF (ILOADX.GT.2) THEN
         IF (IBACK2.EQ.0) THEN
            IF(I.NE.0) THEN
               CASA4=CASE(:I-1)//'.ALL4P3'
            ELSE
               CASA4=CASE//'.ALL4P3'
            ENDIF
         ELSE
            IF(I.NE.0) THEN
               CASA4=CASE(:I-1)//'.ALL4N3'
            ELSE
               CASA4=CASE//'.ALL4N3'
            ENDIF
         ENDIF
      ENDIF
      OPEN(UNIT=64,FILE=CASA4,STATUS='UNKNOWN')
      CALL BOSDEC(4,ILOADX,INDIC,IMPERF,64,IFILE8,
     1                 npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                 WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                 THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                 PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
      CLOSE(UNIT=64)
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (ILOADX.LE.2) THEN
         IF (IBACK2.EQ.0) THEN
            WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,+(mode 2) imperfect=',
     1         CASA4
         ELSE
            WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,-(mode 2) imperfect=',
     1         CASA4
         ENDIF
      ENDIF
      IF (ILOADX.GT.2) THEN
         IF (IBACK2.EQ.0) THEN
            WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,+(mode 4) imperfect=',
     1         CASA4
         ELSE
            WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,-(mode 4) imperfect=',
     1         CASA4
         ENDIF
```

```
         ENDIF
       ENDIF
C      End of (ITYPEX.EQ.2) condition
C
C      IF (IDV.EQ.1.AND.IMODX.EQ.1) THEN
C         ITESTX = 1
C         WRITE(IFILE8,'(A,I3)')' ITESTX=',ITESTX
C      ENDIF
       CALL B4READ
       CALL B4MAIN
c
       bskin1 = 10.E+16
       bstif1 = 10.E+16
       bskin2 = 10.E+16
       bstif2 = 10.E+16
       sknmx1 = 0.
       stfmx1 = 0.
       sknmx2 = 0.
       stfmx2 = 0.
       do 403 iseg = 1,NSEG
          ipoint = iseg + 1
          if (xinput(ipoint).lt.xlimit) then
             bskin1 = min(bskin1,BUCMIN(iseg))
             bstif1 = min(bstif1,BUCMNS(iseg))
             sknmx1 = max(sknmx1,SKNMAX(iseg))
             stfmx1 = max(stfmx1,STFMXS(iseg))
          else
             bskin2 = min(bskin2,BUCMIN(iseg))
             bstif2 = min(bstif2,BUCMNS(iseg))
             sknmx2 = max(sknmx2,SKNMAX(iseg))
             stfmx2 = max(stfmx2,STFMXS(iseg))
          endif
   403 continue
c
       IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
       IF (NPRINX.GT.0) THEN
          IF (ILOADX.LE.2) WRITE(IFILE8,'(/,A,I2,A)')
      1 ' *** Output from mode 2 INDIC=0, stress analysis; IMODX=',
      1   IMODX,' ***'
          IF (ILOADX.GT.2) WRITE(IFILE8,'(/,A,I2,A)')
      1 ' *** Output from mode 4 INDIC=0, stress analysis; IMODX=',
      1   IMODX,' ***'
          WRITE(IFILE8,'(A,/,(1P,5E12.4))')
      1 ' Pressure multiplier, P, for all load steps=',
      1   (PSTEP(I),I=1,ISTEP)
          WRITE(IFILE8,'(A,/,(1P,5E12.4))')
      1 ' End displacement, ENDUVS, for all load steps=',
      1   (ENDUVS(I),I=1,ISTEP)
```

```
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
         DO 40 I = 1,NSEG
            WRITE(IFILE8,'(/,A,I4)')
     1   ' Local skin and smeared stiffener buckling and stress, Seg.',I
            WRITE(IFILE8,'(A,1P,E12.4,A,I3,/,A,1P,E12.4,A,I3)')
     1    ' Skin buckling load factor,                    BUCMIN=',
     1      BUCMIN(I),' at nodal point',IBUCMN(I),
     1    ' Smeared stringer/isogrid buckling load factor, BUCMNS=',
     1      BUCMNS(I),' at nodal point',IBUCMS(I)
            WRITE(IFILE8,'(A,1P,E12.4,A,I3,/,A,1P,E12.4,A,I3)')
     1    ' Smeared ring buckling load factor,            BUCMNR=',
     1      BUCMNR(I),' at nodal point',IBUCMR(I),
     1    ' Smeared stringer/isogrid maximum eff. stress,  STFMXS=',
     1      STFMXS(I),' at nodal point',ISTFMS(I)
            WRITE(IFILE8,'(A,1P,E12.4,A,I3,/,A,1P,E12.4,A,I3)')
     1    ' Smeared ring maximum effective stress,        STFMXR=',
     1      STFMXR(I),' at nodal point',ISTFMR(I),
     1    ' Shell skin maximum effective stress,          SKNMAX=',
     1      SKNMAX(I),' at nodal point',ISKNMX(I)
   40    CONTINUE
c
         IF (IMODX.EQ.0) THEN
            BUCSKU2(1) = bskin1
            BUCSTU2(1) = bstif1
            STRMAU2(1) = sknmx1
            STRSTU2(1) = stfmx1
            BUCSKU2(2) = bskin2
            BUCSTU2(2) = bstif2
            STRMAU2(2) = sknmx2
            STRSTU2(2) = stfmx2
         ENDIF
C
         IF (IMODX.EQ.0) THEN
            ENDUVU2= ENDUV
         ENDIF
         WRITE(IFILE8,'(/,A,A)')
     1 '                                                    PERTURBED',
     1 '  UNPERTURBED'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
         WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1 ' Region 1 skin buckling load factor,        bskin1=',
     1   bskin1,BUCSKU2(1),
     1 ' Region 1 stiffener buckling load factor,   bstif1=',
     1   bstif1,BUCSTU2(1)
         WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1 ' Region 1 skin maximum effective stress,    sknmx1=',
     1   sknmx1,STRMAU2(1),
```

```
     1 ' Region 1 stiffener max. effective stress,  stfmx1=',
     1   stfmx1,STRSTU2(1)
         WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1 ' Region 2 skin buckling load factor,        bskin2=',
     1   bskin2,BUCSKU2(2),
     1 ' Region 2 stiffener buckling load factor,   bstif2=',
     1   bstif2,BUCSTU2(2)
         WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1 ' Region 2 skin maximum effective stress,    sknmx2=',
     1   sknmx2,STRMAU2(2),
     1 ' Region 2 stiffener max. effective stress,  stfmx2=',
     1   stfmx2,STRSTU2(2)
         WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Normal displacement of shell at apex,       ENDUV=',
     1   ENDUV,ENDUVU2
      ENDIF
C     End of (NPRINX.GT.0) condition
      IF (IMODX.EQ.0) THEN
        PMAX02 = PRESS(ILOADX)
        PSTEP2 = PSTEP(ISTEP)
        IF (NCASES.EQ.1.AND.IBACK2.EQ.0) PSTEP2SV = PSTEP2
        IF (NCASES.EQ.1.AND.IBACK2.EQ.1) THEN
           IF (PSTEP2.LT.0.9999*PSTEP2SV) THEN
              M2MULT = -1
              PMAX02 = PSTEP2
              PMAX02M= PSTEP2
              CALL GASP (DUM1,DUM2,-2,DUM3)
              GO TO 436
           ENDIF
C
           IF (PSTEP2SV.LT.0.9999*PSTEP2) THEN
              M2MULT = 1
              PMAX02 = PSTEP2SV
              PMAX02M = PSTEP2
              BUCSKU2(1)= BUCSKU2S(1)
              BUCSTU2(1)= BUCSTU2S(1)
              STRMAU2(1)= STRMAU2S(1)
              STRSTU2(1)= STRSTU2S(1)
              BUCSKU2(2)= BUCSKU2S(2)
              BUCSTU2(2)= BUCSTU2S(2)
              STRMAU2(2)= STRMAU2S(2)
              STRSTU2(2)= STRSTU2S(2)
              ENDUVU2= ENDUVU2S
              bskin1 = bskin1s
              bstif1 = bstif1s
              sknmx1 = sknmx1s
              stfmx1 = stfmx1s
              bskin2 = bskin2s
```

```
              bstif2 = bstif2s
              sknmx2 = sknmx2s
              stfmx2 = stfmx2s
              ENDUV  = ENDUVSV
              PMAX02 = PMAX02S
              CALL GASP (DUM1,DUM2,-2,DUM3)
              GO TO 436
            ENDIF
          ENDIF
C         End of (NCASES.EQ.1.AND.IBACK2.EQ.1) condition
          IF (PSTEP2.LT.PRESS(ILOADX)) THEN
            WRITE(IFILE8,'(/,A,/,A,/,A,1P,E12.4,A,/,A,/,A,/,A)')
     1    ' ****** WARNING ******* WARNING ******** WARNING *********',
     1    ' For the current design the imperfect shell collapses',
     1    ' at a pressure that is lower than the design pressure,',
     1      PRESS(ILOADX),'.',
     1    ' In order to avoid erratic and incorrect gradients of',
     1    ' the stress and local buckling constraints, we must',
     1    ' redo the nonlinear stress analysis with a new maximum'
            WRITE(IFILE8,'(A,1P,E12.4,A,/,A,/,A,/,A,/)')
     1    ' pressure,',PSTEP2,'. This new maximum also affects the',
     1    ' starting pressure and the pressure increment used in the',
     1    ' nonlinear stress, collapse, and buckling analyses.',
     1    ' *** END WARNING **** END WARNING **** END WARNING *******'
C
            IF (ILOADX.LE.2) WRITE(IFILE8,'(/,A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,   mode 2 imperfection',
     1'       IMODX=',IMODX
            IF (ILOADX.GT.2) WRITE(IFILE8,'(/,A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,   mode 4 imperfection',
     1'       IMODX=',IMODX
C23456789012345678901234567890123456789012345678901234567890123456789012
            CALL GASP (DUM1,DUM2,-2,DUM3)
            PMAX02 = PSTEP2
            PMAX  = PMAX02
            CALL BOSDEC(4,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C         Get CASE.ALL5 file for input for BIGBOSOR4...
C
C         CASE.ALL5P is an input file for BIGBOSOR4 for nonlinear stress
C         analysis of the axisymmetrically imperfect ellipsoidal shell
C         with a +(mode 2) imperfection shape (Load set no. ILOADX=1)
C         in which P(COLLAPSE) is less than P(design).
C
C         CASE.ALL5N is an input file for BIGBOSOR4 for nonlinear stress
```

```
C       analysis of the axisymmetrically imperfect ellipsoidal shell
C       with a -(mode 2) imperfection shape (Load set no. ILOADX=2)
C       in which P(COLLAPSE) is less than P(design).
C
C       CASE.ALL5P3 is an input file for BIGBOSOR4 for nonlinear stress
C       analysis of the axisymmetrically imperfect ellipsoidal shell
C       with a +(mode 4) imperfection shape (Load set no. ILOADX=3)
C       in which P(COLLAPSE) is less than P(design).
C
C       CASE.ALL5N3 is an input file for BIGBOSOR4 for nonlinear stress
C       analysis of the axisymmetrically imperfect ellipsoidal shell
C       with a -(mode 4) imperfection shape (Load set no. ILOADX=4)
C       in which P(COLLAPSE) is less than P(design).
C
        IF (ITYPEX.EQ.2) THEN
           I=INDEX(CASE,' ')
           IF (ILOADX.LE.2) THEN
              IF (IBACK2.EQ.0) THEN
                 IF(I.NE.0) THEN
                    CASA5=CASE(:I-1)//'.ALL5P'
                 ELSE
                    CASA5=CASE//'.ALL5P'
                 ENDIF
              ELSE
                 IF(I.NE.0) THEN
                    CASA5=CASE(:I-1)//'.ALL5N'
                 ELSE
                    CASA5=CASE//'.ALL5N'
                 ENDIF
              ENDIF
           ENDIF
           IF (ILOADX.GT.2) THEN
              IF (IBACK2.EQ.0) THEN
                 IF(I.NE.0) THEN
                    CASA5=CASE(:I-1)//'.ALL5P3'
                 ELSE
                    CASA5=CASE//'.ALL5P3'
                 ENDIF
              ELSE
                 IF(I.NE.0) THEN
                    CASA5=CASE(:I-1)//'.ALL5N3'
                 ELSE
                    CASA5=CASE//'.ALL5N3'
                 ENDIF
              ENDIF
           ENDIF
           OPEN(UNIT=65,FILE=CASA5,STATUS='UNKNOWN')
           CALL BOSDEC(4,ILOADX,INDIC,IMPERF,65,IFILE8,
```

```
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
            CLOSE(UNIT=65)
C23456789012345678901234567890123456789012345678901234567890123456789012
            IF (ILOADX.LE.2) THEN
               IF (IBACK2.EQ.0) THEN
                  WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,+(mode 2) imperfect=',
     1             CASA5
               ELSE
                  WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,-(mode 2) imperfect=',
     1             CASA5
               ENDIF
            ENDIF
            IF (ILOADX.GT.2) THEN
               IF (IBACK2.EQ.0) THEN
                  WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,+(mode 4) imperfect=',
     1             CASA5
               ELSE
                  WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file for nonlinear stress,-(mode 4) imperfect=',
     1             CASA5
               ENDIF
            ENDIF
C23456789012345678901234567890123456789012345678901234567890123456789012
         ENDIF
C        End of (ITYPEX.EQ.2) condition
C
         CALL B4READ
         CALL B4MAIN
c
         bskin1 = 10.E+16
         bstif1 = 10.E+16
         bskin2 = 10.E+16
         bstif2 = 10.E+16
         sknmx1 = 0.
         stfmx1 = 0.
         sknmx2 = 0.
         stfmx2 = 0.
         do 423 iseg = 1,NSEG
            ipoint = iseg + 1
            if (xinput(ipoint).lt.xlimit) then
               bskin1 = min(bskin1,BUCMIN(iseg))
               bstif1 = min(bstif1,BUCMNS(iseg))
```

```
                sknmx1 = max(sknmx1,SKNMAX(iseg))
                stfmx1 = max(stfmx1,STFMXS(iseg))
             else
                bskin2 = min(bskin2,BUCMIN(iseg))
                bstif2 = min(bstif2,BUCMNS(iseg))
                sknmx2 = max(sknmx2,SKNMAX(iseg))
                stfmx2 = max(stfmx2,STFMXS(iseg))
             endif
  423     continue
c
          IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
          IF (PSTEP(ISTEP).LT.PRESS(ILOADX)) THEN
             ISKIP(JJJ1) = 1
             PMAX02 = PSTEP(ISTEP)
          ENDIF
          IF (NPRINX.GT.0) THEN
            IF (ILOADX.LE.2) WRITE(IFILE8,'(/,A,I2,A)')
     1    ' *** Output from mode 2 INDIC=0, stress analysis; IMODX=',
     1      IMODX,' ***'
            IF (ILOADX.GT.2) WRITE(IFILE8,'(/,A,I2,A)')
     1    ' *** Output from mode 4 INDIC=0, stress analysis; IMODX=',
     1      IMODX,' ***'
            WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1    ' Pressure multiplier, P, for all load steps=',
     1      (PSTEP(I),I=1,ISTEP)
            WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1    ' End displacement, ENDUVS, for all load steps=',
     1      (ENDUVS(I),I=1,ISTEP)
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
            DO 42 I = 1,NSEG
               WRITE(IFILE8,'(/,A,I4)')
     1   ' Local skin and smeared stiffener buckling and stress, Seg.',I
               WRITE(IFILE8,'(A,1P,E12.4,A,I3,/,A,1P,E12.4,A,I3)')
     1       ' Skin buckling load factor,                    BUCMIN=',
     1         BUCMIN(I),' at nodal point',IBUCMN(I),
     1       ' Smeared stringer/isogrid buckling load factor, BUCMNS=',
     1         BUCMNS(I),' at nodal point',IBUCMS(I)
               WRITE(IFILE8,'(A,1P,E12.4,A,I3,/,A,1P,E12.4,A,I3)')
     1       ' Smeared ring buckling load factor,            BUCMNR=',
     1         BUCMNR(I),' at nodal point',IBUCMR(I),
     1       ' Smeared stringer/isogrid maximum eff. stress,  STFMXS=',
     1         STFMXS(I),' at nodal point',ISTFMS(I)
               WRITE(IFILE8,'(A,1P,E12.4,A,I3,/,A,1P,E12.4,A,I3)')
     1       ' Smeared ring maximum effective stress,        STFMXR=',
     1         STFMXR(I),' at nodal point',ISTFMR(I),
     1       ' Shell skin maximum effective stress,          SKNMAX=',
     1         SKNMAX(I),' at nodal point',ISKNMX(I)
```

```
   42       CONTINUE
C
            BUCSKU2(1) = bskin1
            BUCSTU2(1) = bstif1
            STRMAU2(1) = sknmx1
            STRSTU2(1) = stfmx1
            BUCSKU2(2) = bskin2
            BUCSTU2(2) = bstif2
            STRMAU2(2) = sknmx2
            STRSTU2(2) = stfmx2
C
            ENDUVU2= ENDUV
            WRITE(IFILE8,'(/,A,A)')
     1 '                                                    PERTURBED',
     1    '  UNPERTURBED'
            WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1    ' Region 1 skin buckling load factor,        bskin1=',
     1      bskin1,BUCSKU2(1),
     1    ' Region 1 stiffener buckling load factor,   bstif1=',
     1      bstif1,BUCSTU2(1)
            WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1    ' Region 1 skin maximum effective stress,    sknmx1=',
     1      sknmx1,STRMAU2(1),
     1    ' Region 1 stiffener max. effective stress,  stfmx1=',
     1      stfmx1,STRSTU2(1)
            WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1    ' Region 2 skin buckling load factor,        bskin2=',
     1      bskin2,BUCSKU2(2),
     1    ' Region 2 stiffener buckling load factor,   bstif2=',
     1      bstif2,BUCSTU2(2)
            WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
     1    ' Region 2 skin maximum effective stress,    sknmx2=',
     1      sknmx2,STRMAU2(2),
     1    ' Region 2 stiffener max. effective stress,  stfmx2=',
     1      stfmx2,STRSTU2(2)
            WRITE(IFILE8,'(A,1P,2E12.4)')
     1    ' Normal displacement of shell at apex,       ENDUV=',
     1      ENDUV,ENDUVU2
          ENDIF
C         End of IF (NPRINX.GT.0) condition
        ENDIF
C       End of IF (PSTEP1.LT.PRESS(ILOADX)) condition
      ENDIF
C     End of IF (IMODX.EQ.0) condition
C23456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
C     CALL B4POST
      CALL GASP (DUM1,DUM2,-2,DUM3)
```

```
C
      IF (NCASES.EQ.2.OR.NCASES.EQ.4) THEN
         PMAX02P = PMAX02
         PMAX02M = PMAX02
      ENDIF
      IF (NCASES.EQ.1.AND.IMODX.EQ.0.AND.IBACK2.EQ.0) THEN
         FMARG(1) = bskin1 /(SKNBK2A(ILOADX,1)*SKNBK2F(ILOADX,1))-1.0
         FMARG(2) = bstif1 /(STFBK2A(ILOADX,1)*STFBK2F(ILOADX,1))-1.0
         FMARG(3) = SKNST2A(ILOADX,1)/(sknmx1 *SKNST2F(ILOADX,1))-1.0
         FMARG(4) = STFST2A(ILOADX,1)/(stfmx1 *STFST2F(ILOADX,1))-1.0
         FMARG(5) = WAPEX2A(ILOADX)/(ABS(ENDUV)*WAPEX2F(ILOADX))-1.0
         FMARG(6) = bskin2 /(SKNBK2A(ILOADX,2)*SKNBK2F(ILOADX,2))-1.0
         FMARG(7) = bstif2 /(STFBK2A(ILOADX,2)*STFBK2F(ILOADX,2))-1.0
         FMARG(8) = SKNST2A(ILOADX,2)/(sknmx2 *SKNST2F(ILOADX,2))-1.0
         FMARG(9) = STFST2A(ILOADX,2)/(stfmx2 *STFST2F(ILOADX,2))-1.0
         FMARG(10)= WAPEX2A(ILOADX)/(ABS(ENDUV)*WAPEX2F(ILOADX))-1.0
         FMARMIN = 100000.
         DO 425 I = 1,10
            FMARMIN = MIN(FMARMIN,FMARG(I))
  425    CONTINUE
         IF (FMARMIN.GT.2.0) THEN
            PMAX02P = PMAX02
            PMAX02M = PMAX02
            WRITE(IFILE8,'(/,A,/,A,1P,E12.4)')
     1     ' -(mode 2) not investigated because the minimum margin for',
     1     ' +(mode 2) > 2. The minimum margin for +(mode 2), FMARMIN=',
     1       FMARMIN
            GO TO 436
         ENDIF
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
C      Check for negative of mode 2...
         BUCSKU2S(1)= BUCSKU2(1)
         BUCSTU2S(1)= BUCSTU2(1)
         STRMAU2S(1)= STRMAU2(1)
         STRSTU2S(1)= STRSTU2(1)
         BUCSKU2S(2)= BUCSKU2(2)
         BUCSTU2S(2)= BUCSTU2(2)
         STRMAU2S(2)= STRMAU2(2)
         STRSTU2S(2)= STRSTU2(2)
         ENDUVU2S= ENDUVU2
         bskin1s = bskin1
         bstif1s = bstif1
         sknmx1s = sknmx1
         stfmx1s = stfmx1
         bskin2s = bskin2
         bstif2s = bstif2
         sknmx2s = sknmx2
```

```
         stfmx2s = stfmx2
         ENDUVSV = ABS(ENDUV)
         PMAX02S = PMAX02
         DO 43 I = 1,I2
          WMODEX(I) = -WMODX2(I)
   43    CONTINUE
         WRITE(IFILE8,'(/,/,A)')
     1 ' Check for imperfection shape with negative of mode 2'
         WRITE(IFILE8,'(A,A,I2)')
     1' *** Start nonlinear axisymmetric stress,-(mode 2) imperfection',
     1'      IMODX=',IMODX
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
         IBACK2 = 1
         PMAX02P = PMAX02
         GO TO 398
      ENDIF
C     End of (NCASES.EQ.1.AND.IMODX.EQ.0.AND.IBACK2.EQ.0) condition
C
      IF (NCASES.EQ.1.AND.IMODX.EQ.0.AND.IBACK2.EQ.1) THEN
         PMAX02M = PMAX02
C      Decide which mode 1 (+ or -) to use for margins
         FMARGP(1) =bskin1s/(SKNBK2A(ILOADX,1)*SKNBK2F(ILOADX,1))-1.0
         FMARGM(1) =bskin1 /(SKNBK2A(ILOADX,1)*SKNBK2F(ILOADX,1))-1.0
         FMARGP(2) =bstif1s/(STFBK2A(ILOADX,1)*STFBK2F(ILOADX,1))-1.0
         FMARGM(2) =bstif1 /(STFBK2A(ILOADX,1)*STFBK2F(ILOADX,1))-1.0
         FMARGP(3) =SKNST2A(ILOADX,1)/(sknmx1s*SKNST2F(ILOADX,1))-1.0
         FMARGM(3) =SKNST2A(ILOADX,1)/(sknmx1 *SKNST2F(ILOADX,1))-1.0
         FMARGP(4) =STFST2A(ILOADX,1)/(stfmx1s*STFST2F(ILOADX,1))-1.0
         FMARGM(4) =STFST2A(ILOADX,1)/(stfmx1 *STFST2F(ILOADX,1))-1.0
         FMARGP(5) = WAPEX2A(ILOADX)/(ENDUVSV*WAPEX2F(ILOADX)) -1.0
         FMARGM(5) = WAPEX2A(ILOADX)/(ABS(ENDUV)*WAPEX2F(ILOADX))-1.0
         FMARGP(6) =bskin2s/(SKNBK2A(ILOADX,2)*SKNBK2F(ILOADX,2))-1.0
         FMARGM(6) =bskin2 /(SKNBK2A(ILOADX,2)*SKNBK2F(ILOADX,2))-1.0
         FMARGP(7) =bstif2s/(STFBK2A(ILOADX,2)*STFBK2F(ILOADX,2))-1.0
         FMARGM(7) =bstif2 /(STFBK2A(ILOADX,2)*STFBK2F(ILOADX,2))-1.0
         FMARGP(8) =SKNST2A(ILOADX,2)/(sknmx2s*SKNST2F(ILOADX,2))-1.0
         FMARGM(8) =SKNST2A(ILOADX,2)/(sknmx2 *SKNST2F(ILOADX,2))-1.0
         FMARGP(9) =STFST2A(ILOADX,2)/(stfmx2s*STFST2F(ILOADX,2))-1.0
         FMARGM(9) =STFST2A(ILOADX,2)/(stfmx2 *STFST2F(ILOADX,2))-1.0
         FMARGP(10)= WAPEX2A(ILOADX)/(ENDUVSV*WAPEX2F(ILOADX)) -1.0
         FMARGM(10)= WAPEX2A(ILOADX)/(ABS(ENDUV)*WAPEX2F(ILOADX))-1.0
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
         M2MULT = 1
         FMARMNP = 100000.
         FMARMNM = 100000.
         DO 432 I = 1,10
            FMARMNP = MIN(FMARMNP,FMARGP(I))
            FMARMNM = MIN(FMARMNM,FMARGM(I))
```

```
  432    CONTINUE
         IF (FMARMNM.GT.FMARMNP.OR.FMARMNM.GT.2.0) THEN
            BUCSKU2(1)= BUCSKU2S(1)
            BUCSTU2(1)= BUCSTU2S(1)
            STRMAU2(1)= STRMAU2S(1)
            STRSTU2(1)= STRSTU2S(1)
            BUCSKU2(2)= BUCSKU2S(2)
            BUCSTU2(2)= BUCSTU2S(2)
            STRMAU2(2)= STRMAU2S(2)
            STRSTU2(2)= STRSTU2S(2)
            ENDUVU2= ENDUVU2S
            bskin1 = bskin1s
            bstif1 = bstif1s
            sknmx1 = sknmx1s
            stfmx1 = stfmx1s
            bskin2 = bskin2s
            bstif2 = bstif2s
            sknmx2 = sknmx2s
            stfmx2 = stfmx2s
            ENDUV  = ENDUVSV
            PMAX02 = PMAX02S
            GO TO 436
         ENDIF
         DO 433 I = 1,10
            DIFF = 1.0
            IF (FMARMNM.NE.0.)
     1         DIFF = ABS(FMARMNM - FMARGM(I))/ABS(FMARMNM)
            IF (DIFF.LT.0.0001) THEN
               K = I
               IF (FMARGM(K).LT.FMARGP(K)) THEN
                  M2MULT = -1
                  GO TO 436
               ENDIF
            ENDIF
  433    CONTINUE
       ENDIF
C      End of (NCASES.EQ.1.AND.IMODX.EQ.0.AND.IBACK2.EQ.1) condition
  436 CONTINUE
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
       IF (NCASES.EQ.1.AND.IMODX.EQ.0) THEN
          IF (M2MULT.EQ.1) THEN
              WRITE(IFILE8,'(/,A)')
     1     ' **** The positive mode 2 is more critical than the negative'
          ENDIF
          IF (M2MULT.EQ.-1) THEN
              WRITE(IFILE8,'(/,A)')
     1     ' **** The negative mode 2 is more critical than the positive'
          ENDIF
```

```
          M2MULTC = 1
          IF (PMAX02M.LT.PMAX02P) M2MULTC = -1
          IF (M2MULTC.EQ.1) THEN
              PMAX02 = PMAX02P
C             WRITE(IFILE8,'(/,A,/)')
C     1        ' *** The positive mode 2 is to be used for collapse analysis'
          ENDIF
          IF (M2MULTC.EQ.-1) THEN
              PMAX02 = PMAX02M
C             WRITE(IFILE8,'(/,A,/)')
C     1        ' *** The negative mode 2 is to be used for collapse analysis'
          ENDIF
       ENDIF
C      End of (NCASES.EQ.1.AND.IMODX.EQ.0) condition
C
       IF (NPRINX.GE.1) THEN
          WRITE(IFILE8,'(/,A,A)')
      1 ' The following quantities are used to generate behavioral',
      1 ' constraint conditions and margins:'
          WRITE(IFILE8,'(A,A)')
      1 '                                                        PERTURBED',
      1     '  UNPERTURBED'
          WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
      1 ' Region 1 skin buckling load factor,        bskin1=',
      1   bskin1,BUCSKU2(1),
      1 ' Region 1 stiffener buckling load factor,   bstif1=',
      1   bstif1,BUCSTU2(1)
          WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
      1 ' Region 1 skin maximum effective stress,    sknmx1=',
      1   sknmx1,STRMAU2(1),
      1 ' Region 1 stiffener max. effective stress,  stfmx1=',
      1   stfmx1,STRSTU2(1)
          WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
      1 ' Region 2 skin buckling load factor,        bskin2=',
      1   bskin2,BUCSKU2(2),
      1 ' Region 2 stiffener buckling load factor,   bstif2=',
      1   bstif2,BUCSTU2(2)
          WRITE(IFILE8,'(A,1P,2E12.4,/,A,1P,2E12.4)')
      1 ' Region 2 skin maximum effective stress,    sknmx2=',
      1   sknmx2,STRMAU2(2),
      1 ' Region 2 stiffener max. effective stress,  stfmx2=',
      1   stfmx2,STRSTU2(2)
          WRITE(IFILE8,'(A,1P,2E12.4)')
      1 ' Normal displacement of shell at apex,       ENDUV=',
      1   ENDUV,ENDUVU2
       ENDIF
C      End of (NPRINX.GT.0) condition
C23456789012345678901234567890123456789012345678901234567890123456789012345678901
```

```
      IF (NPRINX.GE.1.AND.PMAX02.LT.0.90*PRESS(ILOADX)) THEN
         WRITE(IFILE8,'(/,A,/,A,/,A,/,A,/,A,/,A,1P,E12.4,/,A,1P,E12.4)')
     1 ' ***** NOTE ***** NOTE ***** NOTE ***** NOTE ***** NOTE *****',
     1 ' WE DO NOT USE LOCAL SKIN AND STIFFENER BUCKLING AND STRESS',
     1 ' CONSTRAINTS, NOR APEX DEFLECTION CONSTRAINT BECAUSE THE',
     1 ' SHELL COLLAPSES AT A LOAD THAT IS LESS THAN 90 PER CENT OF',
     1 ' THE DESIGN LOAD:',
     1 ' Collapse pressure, PMAX02=',PMAX02,
     1 ' Design   pressure, PRESS =',PRESS(ILOADX)
      ENDIF
      IF (PMAX02.GE.0.90*PRESS(ILOADX)) THEN
         SKNBK2(ILOADX,1) = bskin1
         STFBK2(ILOADX,1) = bstif1
         SKNST2(ILOADX,1) = sknmx1
         STFST2(ILOADX,1) = stfmx1
         SKNBK2(ILOADX,2) = bskin2
         STFBK2(ILOADX,2) = bstif2
         SKNST2(ILOADX,2) = sknmx2
         STFST2(ILOADX,2) = stfmx2
         WAPEX2(ILOADX) = ABS(ENDUV)
      ENDIF
C
C     CALL EXIT
C
C
      IF (IMODX.EQ.0) THEN
C
      IF (ILOADX.EQ.1.OR.ILOADX.EQ.3) THEN
C
C  Find axisymmetric collapse (INDIC=0) of imperfect shell, +(mode 1)
C  or +(mode 3)
      IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
      IF (NPRINX.GT.0) WRITE(IFILE8,'(A,I2,A)')
     1' ============ Analysis No. 4 for Load Set No.',ILOADX,
     1' ============'
      IF (ILOADX.EQ.1) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,+(mode 1) imperfection',
     1'      IMODX=',IMODX
      IF (ILOADX.EQ.3) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,+(mode 3) imperfection',
     1'      IMODX=',IMODX
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      CALL MOVER(WSAVEX,1,WMODEX,1,I2)
      INDIC = 0
      IMPERF = 1
      PMAX = PMAX01P
      CALL BOSDEC(2,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
```

```
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C      Get CASE.ALL6P file for input for BIGBOSOR4...
C
C      CASE.ALL6P is an input file for BIGBOSOR4 for nonlinear collapse
C      analysis of the axisymmetrically imperfect ellipsoidal shell
C      with a +(mode 1) imperfection shape if Load set no. ILOADX=1 .
C
C      CASE.ALL6P3 is an input file for BIGBOSOR4 for nonlinear collapse
C      analysis of the axisymmetrically imperfect ellipsoidal shell
C      with a +(mode 3) imperfection shape if Load set no. ILOADX=3 .
C
      IF (ITYPEX.EQ.2) THEN
         I=INDEX(CASE,' ')
         IF (ILOADX.EQ.1) THEN
            IF(I.NE.0) THEN
               CASA6=CASE(:I-1)//'.ALL6P'
            ELSE
               CASA6=CASE//'.ALL6P'
            ENDIF
         ENDIF
         IF (ILOADX.EQ.3) THEN
            IF(I.NE.0) THEN
               CASA6=CASE(:I-1)//'.ALL6P3'
            ELSE
               CASA6=CASE//'.ALL6P3'
            ENDIF
         ENDIF
         OPEN(UNIT=66,FILE=CASA6,STATUS='UNKNOWN')
         CALL BOSDEC(2,ILOADX,INDIC,IMPERF,66,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
         CLOSE(UNIT=66)
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, axisymmetric collpse, +mode 1 imperfect=',
     1   CASA6
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, axisymmetric collpse, +mode 3 imperfect=',
     1   CASA6
      ENDIF
C
C     IF (IDV.EQ.1.AND.IMODX.EQ.1) THEN
C        ITESTX = 1
C        WRITE(IFILE8,'(A,I3)')' ITESTX=',ITESTX
C     ENDIF
```

```
      CALL B4READ
      CALL B4MAIN
      IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C     CALL B4POST
      CALL GASP (DUM1,DUM2,-2,DUM3)
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (NPRINX.GT.0) THEN
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(/,A,I2,A)')
     1 ' *** Output from +(mode 1) INDIC=0, collapse analysis; IMODX=',
     1   IMODX,' *********'
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(/,A,I2,A)')
     1 ' *** Output from +(mode 3) INDIC=0, collapse analysis; IMODX=',
     1   IMODX,' *********'
         WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' Pressure multiplier, P, for all load steps=',
     1   (PSTEP(I),I=1,ISTEP)
         WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' End displacement, ENDUVS, for all load steps=',
     1   (ENDUVS(I),I=1,ISTEP)
C
         PSTEPU1 = PSTEP(ISTEP)
         WRITE(IFILE8,'(/,A,A)')
     1 '                                                PERTURBED',
     1 '  UNPERTURBED'
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with +(mode 1): PSTEP(ISTEP)=',PSTEP(ISTEP),
     1   PSTEPU1
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with +(mode 3): PSTEP(ISTEP)=',PSTEP(ISTEP),
     1   PSTEPU1
      ENDIF
C     End of (NPRINX.GT.0) condition
C
      PPLUS = PSTEP(ISTEP)
      PPLUS1= PPLUS
C
C     CALL EXIT
C
      PMXCOL1 = PMAX01P
C
      ENDIF
C     End of (ILOADX.EQ.1.OR.ILOADX.EQ.3) condition
C
      IF ((NCASES.EQ.1.AND.ILOADX.EQ.1)
     1                 .OR.ILOADX.EQ.2.OR.ILOADX.EQ.4) THEN
C
C  Find axisymmetric collapse (INDIC=0) of imperfect shell, -(mode 1)
```

```
C
      IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
      IF (NPRINX.GT.0) WRITE(IFILE8,'(A,I2,A)')
     1' ============ Analysis No. 4 for Load Set No.',ILOADX,
     1' ============'
      IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,-(mode 1) imperfection',
     1'      IMODX=',IMODX
      IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,-(mode 3) imperfection',
     1'      IMODX=',IMODX
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      DO 46 I = 1,I2
         WMODEX(I) = -WSAVEX(I)
   46 CONTINUE
      INDIC = 0
      IMPERF = 1
      PMAX = PMAX01M
      CALL BOSDEC(2,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C      Get CASE.ALL6N file for input for BIGBOSOR4...
C
C      CASE.ALL6N is an input file for BIGBOSOR4 for nonlinear collapse
C      analysis of the axisymmetrically imperfect ellipsoidal shell
C      with a -(mode 1) imperfection shape if Load set no. ILOADX=2 .
C
C      CASE.ALL6N3 is an input file for BIGBOSOR4 for nonlinear collapse
C      analysis of the axisymmetrically imperfect ellipsoidal shell
C      with a -(mode 3) imperfection shape if Load set no. ILOADX=4 .
C
      IF (ITYPEX.EQ.2) THEN
         I=INDEX(CASE,' ')
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) THEN
            IF(I.NE.0) THEN
               CASA6=CASE(:I-1)//'.ALL6N'
            ELSE
               CASA6=CASE//'.ALL6N'
            ENDIF
         ENDIF
         IF (ILOADX.EQ.4) THEN
            IF(I.NE.0) THEN
               CASA6=CASE(:I-1)//'.ALL6N3'
            ELSE
               CASA6=CASE//'.ALL6N3'
            ENDIF
```

```
         ENDIF
         OPEN(UNIT=66,FILE=CASA6,STATUS='UNKNOWN')
         CALL BOSDEC(2,ILOADX,INDIC,IMPERF,66,IFILE8,
     1                 npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                 WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                 THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                 PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
         CLOSE(UNIT=66)
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, axisymmetric collpse, -mode 1 imperfect=',
     1   CASA6
         IF (ILOADX.EQ.4) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, axisymmetric collpse, -mode 3 imperfect=',
     1   CASA6
       ENDIF
C      End of (ITYPEX.EQ.2) condition
C
C      IF (IDV.EQ.1.AND.IMODX.EQ.1) THEN
C         ITESTX = 1
C         WRITE(IFILE8,'(A,I3)')' ITESTX=',ITESTX
C      ENDIF
       CALL B4READ
       CALL B4MAIN
       IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C      CALL B4POST
       CALL GASP (DUM1,DUM2,-2,DUM3)
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
       IF (NPRINX.GT.0) THEN
          IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(/,A,I2,A)')
     1 ' *** Output from -(mode 1) INDIC=0, collapse analysis; IMODX=',
     1   IMODX,' *********'
          IF (ILOADX.EQ.3) WRITE(IFILE8,'(/,A,I2,A)')
     1 ' *** Output from -(mode 3) INDIC=0, collapse analysis; IMODX=',
     1   IMODX,' *********'
          WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' Pressure multiplier, P, for all load steps=',
     1   (PSTEP(I),I=1,ISTEP)
          WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' End displacement, ENDUVS, for all load steps=',
     1   (ENDUVS(I),I=1,ISTEP)
C
          PSTEPU1 = PSTEP(ISTEP)
          WRITE(IFILE8,'(/,A,A)')
     1 '                                                PERTURBED',
     1 '  UNPERTURBED'
          IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with -(mode 1): PSTEP(ISTEP)=',PSTEP(ISTEP),
```

```
     1   PSTEPU1
         IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with -(mode 3): PSTEP(ISTEP)=',PSTEP(ISTEP),
     1   PSTEPU1
      ENDIF
C     End of (NPRINX.GT.0) condition
C
      PMINUS = PSTEP(ISTEP)
      PMINUS1= PMINUS
C
C     CALL EXIT
C
      PMXCOL1 = PMAX01M
C
      ENDIF
C     End of ((NCASES.EQ.1.AND.ILOADX.EQ.1)
C    1                    .OR.ILOADX.EQ.2.OR.ILOADX.EQ.4) condition
C
      CLAPS1(ILOADX) = PSTEP(ISTEP)
      IF (NCASES.EQ.1) THEN
         IF (PMINUS.LT.PPLUS) THEN
            M1MULTC = -1
            PSTEPU1 = PMINUS
            CLAPS1(ILOADX) = PMINUS
            PMXCOL1 = PMAX01M
         ELSE
            M1MULTC = 1
            PSTEPU1 = PPLUS
            CLAPS1(ILOADX) = PPLUS
            PMXCOL1 = PMAX01P
         ENDIF
      ENDIF
      IF (NPRINX.GT.0) THEN
         WRITE(IFILE8,'(/,A,A)')
     1 ' The following quantity is used to generate the behavioral',
     1 ' constraint condition and margin:'
         WRITE(IFILE8,'(A,A)')
     1 '                                                  PERTURBED',
     1 '  UNPERTURBED'
         IF (ILOADX.LE.2) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with mode 1: CLAPS1(ILOADX)=',
     1   CLAPS1(ILOADX),PSTEPU1
         IF (ILOADX.GT.2) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with mode 3: CLAPS1(ILOADX)=',
     1   CLAPS1(ILOADX),PSTEPU1
      ENDIF
      ENDIF
C     End of (IMODX.EQ.0) condition
```

```
C
      IF (IMODX.EQ.1) THEN
C  Find axisymmetric collapse (INDIC=0) of imperfect shell, mode 1
C
      IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
      IF (NPRINX.GT.0) WRITE(IFILE8,'(A,I2,A)')
     1' ============ Analysis No. 4 for Load Set No.',ILOADX,
     1' ============'
      IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) THEN
         IF (M1MULTC.EQ.1)  WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,+(mode 1) imperfection',
     1'      IMODX=',IMODX
         IF (M1MULTC.EQ.-1) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,-(mode 1) imperfection',
     1'      IMODX=',IMODX
      ENDIF
      IF (ILOADX.EQ.3.OR.ILOADX.EQ.4) THEN
         IF (M1MULTC.EQ.1)  WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,+(mode 3) imperfection',
     1'      IMODX=',IMODX
         IF (M1MULTC.EQ.-1) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,-(mode 3) imperfection',
     1'      IMODX=',IMODX
      ENDIF
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      CALL MOVER(WSAVEX,1,WMODEX,1,I2)
      IF (M1MULTC.EQ.-1) THEN
         DO 47 I = 1,I2
          WMODEX(I) = -WSAVEX(I)
   47    CONTINUE
      ENDIF
      INDIC = 0
      IMPERF = 1
      PMAX = PMXCOL1
      CALL BOSDEC(2,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C
C     IF (IDV.EQ.1.AND.IMODX.EQ.1) THEN
C        ITESTX = 1
C        WRITE(IFILE8,'(A,I3)')' ITESTX=',ITESTX
C     ENDIF
      CALL B4READ
      CALL B4MAIN
      IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C     CALL B4POST
```

```
      CALL GASP (DUM1,DUM2,-2,DUM3)
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (NPRINX.GT.0) THEN
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) THEN
            IF (M1MULTC.EQ.1) THEN
               WRITE(IFILE8,'(/,A,I2,A)')
     1  ' *** Output from +(mode 1) INDIC=0, collapse analysis; IMODX=',
     1         IMODX,' *********'
            ELSE
               WRITE(IFILE8,'(/,A,I2,A)')
     1  ' *** Output from -(mode 1) INDIC=0, collapse analysis; IMODX=',
     1         IMODX,' *********'
            ENDIF
         ENDIF
         IF (ILOADX.EQ.3.OR.ILOADX.EQ.4) THEN
            IF (M1MULTC.EQ.1) THEN
               WRITE(IFILE8,'(/,A,I2,A)')
     1  ' *** Output from +(mode 3) INDIC=0, collapse analysis; IMODX=',
     1         IMODX,' *********'
            ELSE
               WRITE(IFILE8,'(/,A,I2,A)')
     1  ' *** Output from -(mode 3) INDIC=0, collapse analysis; IMODX=',
     1         IMODX,' *********'
            ENDIF
         ENDIF
         WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' Pressure multiplier, P, for all load steps=',
     1   (PSTEP(I),I=1,ISTEP)
         WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' End displacement, ENDUVS, for all load steps=',
     1   (ENDUVS(I),I=1,ISTEP)
C
         WRITE(IFILE8,'(/,A,A)')
     1 '                                                  PERTURBED',
     1 '  UNPERTURBED'
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with mode 1: PSTEP(ISTEP)=',PSTEP(ISTEP),
     1   PSTEPU1
         IF (ILOADX.EQ.3.OR.ILOADX.EQ.4) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with mode 3: PSTEP(ISTEP)=',PSTEP(ISTEP),
     1   PSTEPU1
      ENDIF
C     End of (NPRINX.GT.0) condition
C
      CLAPS1(ILOADX) = PSTEP(ISTEP)
C
C     CALL EXIT
```

```
C
      ENDIF
C     End of (IMODX.EQ.1) condition
C
C
      IF (IMODX.EQ.0) THEN
C
      IF (ILOADX.EQ.1.OR.ILOADX.EQ.3) THEN
C
C  Find axisymmetric collapse (INDIC=0) of imperfect shell, +(mode 2)
      IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
      IF (NPRINX.GT.0) WRITE(IFILE8,'(A,I2,A)')
     1' ============ Analysis No. 5 for Load Set No.',ILOADX,
     1' ============'
      IF (ILOADX.EQ.1) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,+(mode 2) imperfection',
     1'      IMODX=',IMODX
      IF (ILOADX.EQ.3) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,+(mode 4) imperfection',
     1'      IMODX=',IMODX
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      CALL MOVER(WMODX2,1,WMODEX,1,I2)
      INDIC = 0
      IMPERF = 1
      PMAX = PMAX02P
      CALL BOSDEC(2,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C      Get CASE.ALL7P file for input for BIGBOSOR4...
C
C      CASE.ALL7P is an input file for BIGBOSOR4 for nonlinear collapse
C      analysis of the axisymmetrically imperfect ellipsoidal shell
C      with a +(mode 2) imperfection shape if Load set no. ILOADX=1 .
C
C      CASE.ALL7P3 is an input file for BIGBOSOR4 for nonlinear collapse
C      analysis of the axisymmetrically imperfect ellipsoidal shell
C      with a +(mode 4) imperfection shape if Load set no. ILOADX=3 .
C
      IF (ITYPEX.EQ.2) THEN
         I=INDEX(CASE,' ')
         IF (ILOADX.EQ.1) THEN
            IF(I.NE.0) THEN
               CASA7=CASE(:I-1)//'.ALL7P'
            ELSE
               CASA7=CASE//'.ALL7P'
            ENDIF
```

```
         ENDIF
         IF (ILOADX.EQ.3) THEN
            IF(I.NE.0) THEN
               CASA7=CASE(:I-1)//'.ALL7P3'
            ELSE
               CASA7=CASE//'.ALL7P3'
            ENDIF
         ENDIF
         OPEN(UNIT=67,FILE=CASA7,STATUS='UNKNOWN')
         CALL BOSDEC(2,ILOADX,INDIC,IMPERF,67,IFILE8,
     1                 npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                 WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                 THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                 PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
         CLOSE(UNIT=67)
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, axisymmetric collpse, +mode 2 imperfect=',
     1   CASA7
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, axisymmetric collpse, +mode 4 imperfect=',
     1   CASA7
       ENDIF
C
C      IF (IDV.EQ.1.AND.IMODX.EQ.1) THEN
C         ITESTX = 1
C         WRITE(IFILE8,'(A,I3)')' ITESTX=',ITESTX
C      ENDIF
       CALL B4READ
       CALL B4MAIN
       IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C      CALL B4POST
       CALL GASP (DUM1,DUM2,-2,DUM3)
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
       IF (NPRINX.GT.0) THEN
          IF (ILOADX.EQ.1) WRITE(IFILE8,'(/,A,I2,A)')
     1 ' *** Output from +(mode 2) INDIC=0, collapse analysis; IMODX=',
     1   IMODX,' *********'
          IF (ILOADX.EQ.3) WRITE(IFILE8,'(/,A,I2,A)')
     1 ' *** Output from +(mode 4) INDIC=0, collapse analysis; IMODX=',
     1   IMODX,' *********'
          WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' Pressure multiplier, P, for all load steps=',
     1   (PSTEP(I),I=1,ISTEP)
          WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' End displacement, ENDUVS, for all load steps=',
     1   (ENDUVS(I),I=1,ISTEP)
C
```

```
         PSTEPU2 = PSTEP(ISTEP)
         WRITE(IFILE8,'(/,A,A)')
     1 '                                                PERTURBED',
     1 '  UNPERTURBED'
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with +(mode 2): PSTEP(ISTEP)=',PSTEP(ISTEP),
     1   PSTEPU2
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with +(mode 4): PSTEP(ISTEP)=',PSTEP(ISTEP),
     1   PSTEPU2
      ENDIF
C     End of (NPRINX.GT.0) condition
C
      PPLUS = PSTEP(ISTEP)
      PPLUS2 = PPLUS
C
C     CALL EXIT
C
      PMXCOL2 = PMAX02P
C
      ENDIF
C     End of (ILOADX.EQ.1.OR.ILOADX.EQ.3) condition
C
      IF ((NCASES.EQ.1.AND.ILOADX.EQ.1)
     1                .OR.ILOADX.EQ.2.OR.ILOADX.EQ.4) THEN
C
C  Find axisymmetric collapse (INDIC=0) of imperfect shell, -(mode 2)
C
      IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
      IF (NPRINX.GT.0) WRITE(IFILE8,'(A,I2,A)')
     1' ============ Analysis No. 5 for Load Set No.',ILOADX,
     1' ============'
      IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,-(mode 2) imperfection',
     1'      IMODX=',IMODX
      IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,-(mode 4) imperfection',
     1'      IMODX=',IMODX
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      DO 48 I = 1,I2
         WMODEX(I) = -WMODX2(I)
   48 CONTINUE
      INDIC = 0
      IMPERF = 1
      PMAX = PMAX02M
      CALL BOSDEC(2,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
```

```
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C      Get CASE.ALL7N file for input for BIGBOSOR4...
C
C      CASE.ALL7N is an input file for BIGBOSOR4 for nonlinear collapse
C      analysis of the axisymmetrically imperfect ellipsoidal shell
C      with a -(mode 2) imperfection shape if Load set no. ILOADX=2 .
C
C      CASE.ALL7N3 is an input file for BIGBOSOR4 for nonlinear collapse
C      analysis of the axisymmetrically imperfect ellipsoidal shell
C      with a -(mode 4) imperfection shape if Load set no. ILOADX=4 .
C
       IF (ITYPEX.EQ.2) THEN
          I=INDEX(CASE,' ')
          IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) THEN
             IF(I.NE.0) THEN
                CASA7=CASE(:I-1)//'.ALL7N'
             ELSE
                CASA7=CASE//'.ALL7N'
             ENDIF
          ENDIF
          IF (ILOADX.EQ.4) THEN
             IF(I.NE.0) THEN
                CASA7=CASE(:I-1)//'.ALL7N3'
             ELSE
                CASA7=CASE//'.ALL7N3'
             ENDIF
          ENDIF
          OPEN(UNIT=67,FILE=CASA7,STATUS='UNKNOWN')
          CALL BOSDEC(2,ILOADX,INDIC,IMPERF,67,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
          CLOSE(UNIT=67)
          IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, axisymmetric collpse, -mode 2 imperfect=',
     1   CASA7
          IF (ILOADX.EQ.4) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, axisymmetric collpse, -mode 4 imperfect=',
     1   CASA7
       ENDIF
C
C      IF (IDV.EQ.1.AND.IMODX.EQ.1) THEN
C         ITESTX = 1
C         WRITE(IFILE8,'(A,I3)')' ITESTX=',ITESTX
C      ENDIF
       CALL B4READ
```

```
      CALL B4MAIN
      IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C     CALL B4POST
      CALL GASP (DUM1,DUM2,-2,DUM3)
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (NPRINX.GT.0) THEN
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(/,A,I2,A)')
     1 ' *** Output from -(mode 2) INDIC=0, collapse analysis; IMODX=',
     1   IMODX,' *********'
         IF (ILOADX.EQ.4) WRITE(IFILE8,'(/,A,I2,A)')
     1 ' *** Output from -(mode 4) INDIC=0, collapse analysis; IMODX=',
     1   IMODX,' *********'
         WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' Pressure multiplier, P, for all load steps=',
     1   (PSTEP(I),I=1,ISTEP)
         WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' End displacement, ENDUVS, for all load steps=',
     1   (ENDUVS(I),I=1,ISTEP)
C
         PSTEPU2 = PSTEP(ISTEP)
         WRITE(IFILE8,'(/,A,A)')
     1 '                                                 PERTURBED',
     1 '  UNPERTURBED'
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with -(mode 2): PSTEP(ISTEP)=',PSTEP(ISTEP),
     1   PSTEPU2
         IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with -(mode 4): PSTEP(ISTEP)=',PSTEP(ISTEP),
     1   PSTEPU2
      ENDIF
C     End of (NPRINX.GT.0) condition
C
      PMINUS = PSTEP(ISTEP)
      PMINUS2= PMINUS
C
C     CALL EXIT
C
      PMXCOL2 = PMAX02M
C
      ENDIF
C     End of ((NCASES.EQ.1.AND.ILOADX.EQ.1)
C    1                    .OR.ILOADX.EQ.2.OR.ILOADX.EQ.4) condition
C
      CLAPS2(ILOADX) = PSTEP(ISTEP)
      IF (NCASES.EQ.1) THEN
         IF (PMINUS.LT.PPLUS) THEN
            M2MULTC = -1
```

```
            PSTEPU2 = PMINUS
            CLAPS2(ILOADX) = PMINUS
            PMXCOL2 = PMAX02M
         ELSE
            M2MULTC = 1
            PSTEPU2 = PPLUS
            CLAPS2(ILOADX) = PPLUS
            PMXCOL2 = PMAX02P
         ENDIF
       ENDIF
C
       IF (NPRINX.GT.0) THEN
          WRITE(IFILE8,'(/,A,A)')
      1 ' The following quantity is used to generate the behavioral',
      1 ' constraint condition and margin:'
          WRITE(IFILE8,'(A,A)')
      1 '                                                  PERTURBED',
      1 '  UNPERTURBED'
          IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,1P,2E12.4)')
      1 ' Collapse pressure with mode 2: CLAPS2(ILOADX)=',
      1   CLAPS2(ILOADX),PSTEPU2
          IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,1P,2E12.4)')
      1 ' Collapse pressure with mode 4: CLAPS2(ILOADX)=',
      1   CLAPS2(ILOADX),PSTEPU2
       ENDIF
       ENDIF
C     End of (IMODX.EQ.0) condition
C23456789012345678901234567890123456789012345678901234567890123456789012
C
       IF (IMODX.EQ.1) THEN
C  Find axisymmetric collapse (INDIC=0) of imperfect shell, mode 2
C  or mode 4
C
       IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
       IF (NPRINX.GT.0) WRITE(IFILE8,'(A,I2,A)')
      1' ============ Analysis No. 5 for Load Set No.',ILOADX,
      1' ============'
       IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) THEN
          IF (M2MULTC.EQ.1)  WRITE(IFILE8,'(A,A,I2)')
      1' ** Start nonlinear axisymmetric collpse,+(mode 2) imperfection',
      1'      IMODX=',IMODX
          IF (M2MULTC.EQ.-1) WRITE(IFILE8,'(A,A,I2)')
      1' ** Start nonlinear axisymmetric collpse,-(mode 2) imperfection',
      1'      IMODX=',IMODX
       ENDIF
       IF (ILOADX.EQ.3.OR.ILOADX.EQ.4) THEN
          IF (M2MULTC.EQ.1)  WRITE(IFILE8,'(A,A,I2)')
      1' ** Start nonlinear axisymmetric collpse,+(mode 4) imperfection',
```

```
     1'      IMODX=',IMODX
         IF (M2MULTC.EQ.-1) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear axisymmetric collpse,-(mode 4) imperfection',
     1'      IMODX=',IMODX
      ENDIF
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      CALL MOVER(WMODX2,1,WMODEX,1,I2)
      IF (M2MULTC.EQ.-1) THEN
         DO 49 I = 1,I2
          WMODEX(I) = -WMODX2(I)
   49    CONTINUE
      ENDIF
      INDIC = 0
      IMPERF = 1
      PMAX = PMXCOL2
      CALL BOSDEC(2,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C
C     IF (IDV.EQ.1.AND.IMODX.EQ.1) THEN
C        ITESTX = 1
C        WRITE(IFILE8,'(A,I3)')' ITESTX=',ITESTX
C     ENDIF
      CALL B4READ
      CALL B4MAIN
      IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C     CALL B4POST
      CALL GASP (DUM1,DUM2,-2,DUM3)
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (NPRINX.GT.0) THEN
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) THEN
            IF (M2MULTC.EQ.1) THEN
               WRITE(IFILE8,'(/,A,I2,A)')
     1  ' *** Output from +(mode 2) INDIC=0, collapse analysis; IMODX=',
     1         IMODX,' *********'
            ELSE
               WRITE(IFILE8,'(/,A,I2,A)')
     1  ' *** Output from -(mode 2) INDIC=0, collapse analysis; IMODX=',
     1         IMODX,' *********'
            ENDIF
         ENDIF
         IF (ILOADX.EQ.3.OR.ILOADX.EQ.4) THEN
            IF (M2MULTC.EQ.1) THEN
               WRITE(IFILE8,'(/,A,I2,A)')
     1  ' *** Output from +(mode 4) INDIC=0, collapse analysis; IMODX=',
```

```
     1          IMODX,' *********'
               ELSE
                  WRITE(IFILE8,'(/,A,I2,A)')
     1  ' *** Output from -(mode 4) INDIC=0, collapse analysis; IMODX=',
     1          IMODX,' *********'
               ENDIF
            ENDIF
            WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' Pressure multiplier, P, for all load steps=',
     1   (PSTEP(I),I=1,ISTEP)
            WRITE(IFILE8,'(A,/,(1P,5E12.4))')
     1 ' End displacement, ENDUVS, for all load steps=',
     1   (ENDUVS(I),I=1,ISTEP)
C
            WRITE(IFILE8,'(/,A,A)')
     1 '                                                  PERTURBED',
     1 '  UNPERTURBED'
            IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with mode 2: PSTEP(ISTEP)=',PSTEP(ISTEP),
     1   PSTEPU2
            IF (ILOADX.EQ.3.OR.ILOADX.EQ.4) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Collapse pressure with mode 4: PSTEP(ISTEP)=',PSTEP(ISTEP),
     1   PSTEPU2
         ENDIF
C     End of (NPRINX.GT.0) condition
C
         CLAPS2(ILOADX) = PSTEP(ISTEP)
C
C     CALL EXIT
C
         ENDIF
C     End of (IMODX.EQ.1) condition
C
         IF (IMODX.EQ.0) THEN
C
C  Find nonlinear bifurcation buckling of imperfect shell, +(mode 1)
C
         IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(A,I2,A,/,A,A,I2)')
        1' ============ Analysis No. 6 for Load Set No.',ILOADX,
        1' ============',
        1' ** Start nonlinear bifurcation buckling,+(mode 1) imperfection',
        1'      IMODX=',IMODX
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(A,I2,A,/,A,A,I2)')
        1' ============ Analysis No. 6 for Load Set No.',ILOADX,
        1' ============',
        1' ** Start nonlinear bifurcation buckling,+(mode 3) imperfection',
        1'      IMODX=',IMODX
```

```
      IF (ILOADX.EQ.2) WRITE(IFILE8,'(A,I2,A,/,A,A,I2)')
     1' ============ Analysis No. 6 for Load Set No.',ILOADX,
     1' ============',
     1' ** Start nonlinear bifurcation buckling,-(mode 1) imperfection',
     1'      IMODX=',IMODX
      IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,I2,A,/,A,A,I2)')
     1' ============ Analysis No. 6 for Load Set No.',ILOADX,
     1' ============',
     1' ** Start nonlinear bifurcation buckling,-(mode 3) imperfection',
     1'      IMODX=',IMODX
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
      IF (ILOADX.EQ.1.OR.ILOADX.EQ.3) THEN
C
      CALL MOVER(WSAVEX,1,WMODEX,1,I2)
      INDIC = 1
      IMPERF = 1
      PMAX01PB = PMAX01P
      PMAX = PMAX01PB
      IF (PMAX.LT.PRESS(ILOADX)) PMAX = 0.9*PMAX01PB
      IF (PMAX.GT.0.9*CLAPS1(ILOADX)) PMAX = 0.9*CLAPS1(ILOADX)
      N0BX = 0
      NMINBX = 0
      NMAXBX = 10
      IF (ITYPEX.EQ.2) NMAXBX = 30
      INCRBX = 1
 4900 CONTINUE
      CALL BOSDEC(3,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C      Get CASE.ALL8P file for input for BIGBOSOR4...
C
C      CASE.ALL8P is an input file for BIGBOSOR4 for nonlinear bifurcation
C      buckling analysis of the axisymmetrically imperfect ellipsoidal
shell
C      with a +(mode 1) imperfection shape if Load set no. ILOADX=1 .
C
C      CASE.ALL8P3 is an input file for BIGBOSOR4 for nonlinear
bifurcation
C      buckling analysis of the axisymmetrically imperfect ellipsoidal
shell
C      with a +(mode 3) imperfection shape if Load set no. ILOADX=3 .
C
      IF (ITYPEX.EQ.2) THEN
         I=INDEX(CASE,' ')
         IF (ILOADX.EQ.1) THEN
```

```
            IF(I.NE.0) THEN
               CASA8=CASE(:I-1)//'.ALL8P'
            ELSE
               CASA8=CASE//'.ALL8P'
            ENDIF
         ENDIF
         IF (ILOADX.EQ.3) THEN
            IF(I.NE.0) THEN
               CASA8=CASE(:I-1)//'.ALL8P3'
            ELSE
               CASA8=CASE//'.ALL8P3'
            ENDIF
         ENDIF
         OPEN(UNIT=68,FILE=CASA8,STATUS='UNKNOWN')
         CALL BOSDEC(3,ILOADX,INDIC,IMPERF,68,IFILE8,
     1                 npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                 WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                 THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                 PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
         CLOSE(UNIT=68)
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, bifurcation buckling, +(mode 1) imperf.=',
     1   CASA8
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, bifurcation buckling, +(mode 3) imperf.=',
     1   CASA8
      ENDIF
C23456789012345678901234567890123456789012345678901234567890123456789012
C
      CALL B4READ
      CALL B4MAIN
      IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C     CALL B4POST
      CALL GASP (DUM1,DUM2,-2,DUM3)
C
C23456789012345678901234567890123456789012345678901234567890123456789012
      IF (NPRINX.GT.0) THEN
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
     1 ' ******* Nonlinear overall bifurcation buckling results ******',
     1 ' Overall buckling, +(mode 1) imperfection shape;',
     1 ' Applied pressure, PMAX =', PMAX
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(A,I2,A)')
     1 ' *** Output from +(mode 1) INDIC=1, buckling analysis; IMODX=',
     1   IMODX,' *********'
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
     1 ' ******* Nonlinear overall bifurcation buckling results ******',
     1 ' Overall buckling, +(mode 3) imperfection shape;',
     1 ' Applied pressure, PMAX =', PMAX
```

```
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(A,I2,A)')
     1 ' *** Output from +(mode 3) INDIC=1, buckling analysis; IMODX=',
     1   IMODX,' *********'
         WRITE(IFILE8, '(/,A,/,A,1P,E12.4,A,I5,/,A)')
     1 ' **** CRITICAL EIGENVALUE AND WAVENUMBER ****',
     1 ' EIGCRT=',EIGCRT,'; NO. OF CIRC. WAVES, NWVCRT=',NWVCRT,
     1 ' ************************************************'
         WRITE(IFILE8,'(/,A,/,A,/,A)')
     1 ' ***** EIGENVALUES AND MODE SHAPES *****',
     1'  EIGENVALUE(CIRC. WAVES)',
     1 ' ======================================='
         DO 50 I = 1,IWAVEB
           WRITE(IFILE8,'(A,1P,E12.4,A,I4,A)')
     1 '  ',EIGCOM(I),'(',NWVCOM(I),')'
   50    CONTINUE
         WRITE(IFILE8,'(A)')
     1 ' ======================================='
C
         WRITE(IFILE8, '(/,A,/,A,1P,E12.4,A,I5,/,A)')
     1 ' **** CRITICAL NEGATIVE EIGENVALUE AND WAVENUMBER ****',
     1 ' EIGCRN=',EIGCRN,'; NO. OF CIRC. WAVES, NWVCRN=',NWVCRN,
     1 ' ************************************************'
         WRITE(IFILE8,'(/,A,/,A,/,A)')
     1 ' ***** NEGATIVE EIGENVALUES AND MODE SHAPES *****',
     1'  EIGENVALUE(CIRC. WAVES)',
     1 ' ======================================='
         DO 5001 I = 1,IWAVEB
           WRITE(IFILE8,'(A,1P,E12.4,A,I4,A)')
     1 '  ',EIGNEG(I),'(',NWVNEG(I),')'
 5001    CONTINUE
         WRITE(IFILE8,'(A)')
     1 ' ======================================='
C
      ENDIF
C     End of (NPRINX.GT.0) condition
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (EIGCRN.LT.0.0) THEN
C BEG MAR 2008
         BUCTST =  PMAX + EIGCRN*PMAX/1000.
C        Disregard the negative eigenvalue if it is too large.
         IF (BUCTST.LE.0.) THEN
            EIGCRN = EIGCRT
            NWVCRN = NWVCRT
         ENDIF
C END MAR 2008
         BUCPRSP = PMAX + EIGCRN*PMAX/1000.
C BEG MAR 2008
```

```
C        IF (BUCPRSP.GT.0.) THEN
            EIGCRT = EIGCRN
            NWVCRT = NWVCRN
C        WRITE(IFILE8,'(/,A,/,A,/,A,/,A,1P,E12.4,/,A,1P,E12.4,/,A,/,A)')
C    1    ' ******* WARNING ******* WARNING ******* WARNING *******',
C    1    ' There are negative eigenvalues. The nonlinear bifurcation',
C    1    ' buckling load is less than the applied load, PMAX:',
C    1    ' Buckling load, BUCPRSP=',BUCPRSP,
C    1    ' Applied  load, PMAX   =',PMAX,
C    1    ' PMAX is being reset equal to 0.9*BUCPRSP and a new',
C    1    ' nonlinear bifurcation buckling load computed.'
C           PMAX = 0.9*BUCPRSP
C           PMAX01PB = BUCPRSP
C           GO TO 4900
C        ENDIF
C END MAR 2008
      ENDIF
      EIG1P  = EIGCRT
      NWAV1P = NWVCRT
      BUCPRSP = PMAX + EIGCRT*PMAX/1000.
      GENBK1(ILOADX) = BUCPRSP/PRESS(ILOADX)
      IF (NPRINX.GT.0) THEN
         WRITE(IFILE8,'(/,A,1P,E12.4,A,I2,A,/,A,1P,E12.4,/,/)')
     1 ' Nonlinear bifurcation buckling pressure, BUCPRSP(circ.waves)=',
     1   BUCPRSP,'(',NWVCRT,')',
     1 ' General bifurcation buckling load factor, GENBK1(ILOADX)=',
     1   GENBK1(ILOADX)
C
         BUCPRU1 = BUCPRSP
         WRITE(IFILE8,'(/,A,A)')
     1 '                                                    PERTURBED',
     1 '  UNPERTURBED'
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Nonlin. bifurcation buckling, +(mode 1):BUCPRSP=',
     1   BUCPRSP,BUCPRU1
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Nonlin. bifurcation buckling, +(mode 3):BUCPRSP=',
     1   BUCPRSP,BUCPRU1
      ENDIF
C
C BEG MAR 2008
C     WRITE(IFILE8,'(/,A,I2,1P3E14.6)')
C    1' +mode 1, ITYPEX,BUCPRSP,PPLUS1,EIGCRT=',
C    1           ITYPEX,BUCPRSP,PPLUS1,EIGCRT
C     IF (ITYPEX.EQ.2.AND.BUCPRSP.LT.0.980*PPLUS1.AND.
C    1    EIGCRT.GT.0.0.AND.(EIGCRT/1000.0).GE.0.02
C    1                 .AND.(EIGCRT/1000.0).LT.0.25) THEN
      IF (ITYPEX.EQ.2.AND.EIGCRT.GT.0.0
```

```
     1               .AND.(EIGCRT/1000.0).GE.0.02) THEN
         INDIC = -2
         N0BX = NWVCRT
         NMINBX = NWVCRT
         NMAXBX = NWVCRT
         CALL BOSDEC(3,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                 npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                 WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                 THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                 PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C
         CALL B4READ
         CALL B4MAIN
         IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C        CALL B4POST
         CALL GASP (DUM1,DUM2,-2,DUM3)
         IF (PPP.LT.BUCPRSP.AND.PPP.LT.0.98*PPLUS1.AND.IDETCT.EQ.1) THEN
            BUCPRSP = PPP
            GENBK1(ILOADX) = BUCPRSP/PRESS(ILOADX)
         ENDIF
         IF (PPP.GE.PPLUS1) WRITE(IFILE8,'(/,A,/,A,/,A,I3,A,/,A)')
     1 ' ***************** INDIC=-2 analysis yields ***************',
     1 ' SHELL COLLAPSES AXISYMMETRICALLY BEFORE NONLINEAR BIFURCA-',
     1 ' TION BUCKLING WITH N =',NWVCRT,' CIRCUMFERENTIAL WAVES.',
     1 ' ************************************************************'
         IF (IDETCT.EQ.1)
     1   WRITE(IFILE8,'(/,A,/,A,1PE12.4,/,A,1PE12.4,/,A,/,A)')
     1 ' ***************** INDIC=-2 analysis yields ***************',
     1 ' Critical pressure from INDIC=-2 analysis        =',PPP,
     1 ' Nonlinear buckling margin from INDIC=-2 analysis=',
     1   GENBK1(ILOADX)-1.0,
     1 ' NOTE: This margin may differ from that obtained with ITYPE=1',
     1 ' ************************************************************'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      ENDIF
C END MAR 2008
C
C     CALL EXIT
C
      BUCPRS = BUCPRSP
      PMAXBUC1 = PMAX01PB
      NWAV1 = NWAV1P
      EIG1  = EIG1P
C
      ENDIF
C     End of (ILOADX.EQ.1.OR.ILOADX.EQ.3) condition
C
      IF ((NCASES.EQ.1.AND.ILOADX.EQ.1)
```

```
     1                .OR.ILOADX.EQ.2.OR.ILOADX.EQ.4) THEN
C
C  Find nonlinear bifurcation buckling of imperfect shell, -(mode 1)
C  or -(mode 3)
C
      IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
      IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear bifurcation buckling,-(mode 1) imperfection',
     1'      IMODX=',IMODX
      IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear bifurcation buckling,-(mode 3) imperfection',
     1'      IMODX=',IMODX
C23456789012345678901234567890123456789012345678901234567890123456789012
      DO 51 I = 1,I2
       WMODEX(I) = -WSAVEX(I)
   51 CONTINUE
      INDIC = 1
      IMPERF = 1
      PMAX01MB = PMAX01M
      PMAX = PMAX01MB
      IF (PMAX.LT.PRESS(ILOADX)) PMAX = 0.9*PMAX01MB
      IF (PMAX.GT.0.9*CLAPS1(ILOADX)) PMAX = 0.9*CLAPS1(ILOADX)
      N0BX = 0
      NMINBX = 0
      NMAXBX = 10
      IF (ITYPEX.EQ.2) NMAXBX = 30
      INCRBX = 1
 5100 CONTINUE
      CALL BOSDEC(3,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C      Get CASE.ALL8N file for input for BIGBOSOR4...
C
C      CASE.ALL8N is an input file for BIGBOSOR4 for nonlinear bifurcation
C      buckling analysis of the axisymmetrically imperfect ellipsoidal
shell
C      with a -(mode 1) imperfection shape if Load set no. ILOADX=1 .
C
C      CASE.ALL8N3 is an input file for BIGBOSOR4 for nonlinear
bifurcation
C      buckling analysis of the axisymmetrically imperfect ellipsoidal
shell
C      with a -(mode 3) imperfection shape if Load set no. ILOADX=3 .
C
      IF (ITYPEX.EQ.2) THEN
         I=INDEX(CASE,' ')
```

```
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) THEN
            IF(I.NE.0) THEN
               CASA8=CASE(:I-1)//'.ALL8N'
            ELSE
               CASA8=CASE//'.ALL8N'
            ENDIF
         ENDIF
         IF (ILOADX.EQ.4) THEN
            IF(I.NE.0) THEN
               CASA8=CASE(:I-1)//'.ALL8N3'
            ELSE
               CASA8=CASE//'.ALL8N3'
            ENDIF
         ENDIF
         OPEN(UNIT=68,FILE=CASA8,STATUS='UNKNOWN')
         CALL BOSDEC(3,ILOADX,INDIC,IMPERF,68,IFILE8,
     1                 npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                 WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                 THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                 PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
         CLOSE(UNIT=68)
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, bifurcation buckling, -(mode 1) imperf.=',
     1   CASA8
         IF (ILOADX.EQ.4) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, bifurcation buckling, -(mode 3) imperf.=',
     1   CASA8
       ENDIF
C23456789012345678901234567890123456789012345678901234567890123456789012
C
       CALL B4READ
       CALL B4MAIN
       IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C      CALL B4POST
       CALL GASP (DUM1,DUM2,-2,DUM3)
C
C23456789012345678901234567890123456789012345678901234567890123456789012
       IF (NPRINX.GT.0) THEN
          IF (ILOADX.EQ.1.OR.ILOADX.EQ.2)
     1   WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
     1 ' ******* Nonlinear overall bifurcation buckling results ******',
     1 ' Overall buckling, -(mode 1) imperfection shape;',
     1 ' Applied pressure, PMAX =', PMAX
          IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,I2,A)')
     1 ' *** Output from -(mode 1) INDIC=1, buckling analysis; IMODX=',
     1   IMODX,' *********'
          IF (ILOADX.EQ.4)
     1   WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
```

```
     1 ' ******* Nonlinear overall bifurcation buckling results ******',
     1 ' Overall buckling, -(mode 3) imperfection shape;',
     1 ' Applied pressure, PMAX =', PMAX
         IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,I2,A)')
     1 ' *** Output from -(mode 3) INDIC=1, buckling analysis; IMODX=',
     1   IMODX,' *********'
         WRITE(IFILE8, '(/,A,/,A,1P,E12.4,A,I5,/,A)')
     1 ' **** CRITICAL EIGENVALUE AND WAVENUMBER ****',
     1 ' EIGCRT=',EIGCRT,'; NO. OF CIRC. WAVES, NWVCRT=',NWVCRT,
     1 ' ************************************************'
         WRITE(IFILE8,'(/,A,/,A,/,A)')
     1 ' ***** EIGENVALUES AND MODE SHAPES *****',
     1'  EIGENVALUE(CIRC. WAVES)',
     1 ' ======================================='
         DO 52 I = 1,IWAVEB
           WRITE(IFILE8,'(A,1P,E12.4,A,I4,A)')
     1 '  ',EIGCOM(I),'(',NWVCOM(I),')'
    52   CONTINUE
         WRITE(IFILE8,'(A)')
     1 ' ======================================='
C
         WRITE(IFILE8, '(/,A,/,A,1P,E12.4,A,I5,/,A)')
     1 ' **** CRITICAL NEGATIVE EIGENVALUE AND WAVENUMBER ****',
     1 ' EIGCRN=',EIGCRN,'; NO. OF CIRC. WAVES, NWVCRN=',NWVCRN,
     1 ' ************************************************'
         WRITE(IFILE8,'(/,A,/,A,/,A)')
     1 ' ***** NEGATIVE EIGENVALUES AND MODE SHAPES *****',
     1'  EIGENVALUE(CIRC. WAVES)',
     1 ' ======================================='
         DO 5201 I = 1,IWAVEB
           WRITE(IFILE8,'(A,1P,E12.4,A,I4,A)')
     1 '  ',EIGNEG(I),'(',NWVNEG(I),')'
 5201    CONTINUE
         WRITE(IFILE8,'(A)')
     1 ' ======================================='
C
      ENDIF
C     End of (NPRINX.GT.0) condition
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (EIGCRN.LT.0.0) THEN
C BEG MAR 2008
         BUCTST =  PMAX + EIGCRN*PMAX/1000.
C        Disregard the negative eigenvalue if it is too large.
         IF (BUCTST.LE.0.) THEN
            EIGCRN = EIGCRT
            NWVCRN = NWVCRT
         ENDIF
```

```
C END MAR 2008
         BUCPRSM = PMAX + EIGCRN*PMAX/1000.
C BEG MAR 2008
C        IF (BUCPRSM.GT.0.) THEN
            EIGCRT = EIGCRN
            NWVCRT = NWVCRN
C        WRITE(IFILE8,'(/,A,/,A,/,A,/,A,1P,E12.4,/,A,1P,E12.4,/,A,/,A)')
C    1    ' ******* WARNING ******* WARNING ******* WARNING *******',
C    1    ' There are negative eigenvalues. The nonlinear bifurcation',
C    1    ' buckling load is less than the applied load, PMAX:',
C    1    ' Buckling load, BUCPRSM=',BUCPRSM,
C    1    ' Applied  load, PMAX   =',PMAX,
C    1    ' PMAX is being reset equal to 0.9*BUCPRSM and a new',
C    1    ' nonlinear bifurcation buckling load computed.'
C           PMAX = 0.9*BUCPRSM
C           PMAX01MB = BUCPRSM
C           GO TO 5100
C        ENDIF
C END MAR 2008
      ENDIF
      EIG1M  = EIGCRT
      NWAV1M = NWVCRT
      BUCPRSM = PMAX + EIGCRT*PMAX/1000.
      GENBK1(ILOADX) = BUCPRSM/PRESS(ILOADX)
      IF (NPRINX.GT.0) THEN
         WRITE(IFILE8,'(/,A,1P,E12.4,A,I2,A,/,A,1P,E12.4,/,/)')
     1 ' Nonlinear bifurcation buckling pressure, BUCPRSM(circ.waves)=',
     1   BUCPRSM,'(',NWVCRT,')',
     1 ' General bifurcation buckling load factor, GENBK1(ILOADX)=',
     1   GENBK1(ILOADX)
C
         BUCPRU1 = BUCPRSM
         WRITE(IFILE8,'(/,A,A)')
     1 '                                                  PERTURBED',
     1 '  UNPERTURBED'
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Nonlin. bifurcation buckling, -(mode 1):BUCPRSM=',
     1   BUCPRSM,BUCPRU1
         IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Nonlin. bifurcation buckling, -(mode 3):BUCPRSM=',
     1   BUCPRSM,BUCPRU1
      ENDIF
C
C BEG MAR 2008
C     WRITE(IFILE8,'(/,A,I2,1P3E14.6)')
C    1' -mode 1, ITYPEX,BUCPRSM,PMINUS1,EIGCRT=',
C    1           ITYPEX,BUCPRSM,PMINUS1,EIGCRT
C     IF (ITYPEX.EQ.2.AND.BUCPRSM.LT.0.980*PMINUS1.AND.
```

```
C     1    EIGCRT.GT.0.0.AND.(EIGCRT/1000.0).GE.0.02
C     1                  .AND.(EIGCRT/1000.0).LT.0.25) THEN
      IF (ITYPEX.EQ.2.AND.EIGCRT.GT.0.0
     1               .AND.(EIGCRT/1000.0).GE.0.02) THEN
          INDIC = -2
          N0BX = NWVCRT
          NMINBX = NWVCRT
          NMAXBX = NWVCRT
          CALL BOSDEC(3,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C
          CALL B4READ
          CALL B4MAIN
          IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C         CALL B4POST
          CALL GASP (DUM1,DUM2,-2,DUM3)
          IF (PPP.LT.BUCPRSM.AND.PPP.LT.0.98*PMINUS1.AND.IDETCT.EQ.1)THEN
             BUCPRSM = PPP
             GENBK1(ILOADX) = BUCPRSM/PRESS(ILOADX)
          ENDIF
          IF (PPP.GE.PMINUS1) WRITE(IFILE8,'(/,A,/,A,/,A,I3,A,/,A)')
     1 ' ***************** INDIC=-2 analysis yields ***************',
     1 ' SHELL COLLAPSES AXISYMMETRICALLY BEFORE NONLINEAR BIFURCA-',
     1 ' TION BUCKLING WITH N =',NWVCRT,' CIRCUMFERENTIAL WAVES.',
     1 ' ******************************************************'
          IF (IDETCT.EQ.1)
     1   WRITE(IFILE8,'(/,A,/,A,1PE12.4,/,A,1PE12.4,/,A,/,A)')
     1 ' ***************** INDIC=-2 analysis yields ***************',
     1 ' Critical pressure from INDIC=-2 analysis        =',PPP,
     1 ' Nonlinear buckling margin from INDIC=-2 analysis=',
     1   GENBK1(ILOADX)-1.0,
     1 ' NOTE: This margin may differ from that obtained with ITYPE=1',
     1 ' ******************************************************'
C2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
      ENDIF
C END MAR 2008
C
C     CALL EXIT
C
      BUCPRS = BUCPRSM
      PMAXBUC1 = PMAX01MB
      NWAV1 = NWAV1M
      EIG1  = EIG1M
C
      ENDIF
```

```
C     End of ((NCASES.EQ.1.AND.ILOADX.EQ.1)
C    1                    .OR.ILOADX.EQ.2.OR.ILOADX.EQ.4) condition
C
      IF (NCASES.EQ.1) BUCPRS = MIN(BUCPRSP,BUCPRSM)
      BUCPRU1 = BUCPRS
      GENBK1(ILOADX) = BUCPRS/PRESS(ILOADX)
      IF (NCASES.EQ.1) THEN
         IF (BUCPRSM.LT.BUCPRSP) THEN
            M1MULTB = -1
            NWAV1 = NWAV1M
            EIG1  = EIG1M
            PMAXBUC1 = PMAX01MB
         ELSE
            M1MULTB = 1
            NWAV1 = NWAV1P
            EIG1  = EIG1P
            PMAXBUC1 = PMAX01PB
         ENDIF
      ENDIF
C23456789012345678901234567890123456789012345678901234567890123456789012
      WRITE(IFILE8,'(A,2I5,1P,E12.4)')
     1' IMODX=0: M1MULTB,NWAV1,PMAXBUC1=',M1MULTB,NWAV1,PMAXBUC1
      IF (PMAXBUC1.LT.PRESS(ILOADX)) PMAXBUC1 = 0.9*PMAXBUC1
      IF (PMAXBUC1.GT.0.9*CLAPS1(ILOADX)) PMAXBUC1 = 0.9*CLAPS1(ILOADX)
C     WRITE(IFILE8,'(/,A,2I7,1P2E12.4)')
C    1' M1MULTB,NWAV1,EIG1,PMAX=',M1MULTB,NWAV1,EIG1,PMAX
      WRITE(IFILE8,'(/,A,A)')
     1 ' The following quantity is used to generate the behavioral',
     1 ' constraint condition and margin:'
      WRITE(IFILE8,'(A,A)')
     1 '                                                  PERTURBED',
     1 '  UNPERTURBED'
      IF (ILOADX.LE.2) THEN
         IF (M1MULTB.EQ.-1) THEN
            WRITE(IFILE8,'(A,1P,2E12.4)')
     1     ' Nonlin. bifurcation buckling, -(mode 1):BUCPRS =',
     1       BUCPRS,BUCPRU1
         ELSE
            WRITE(IFILE8,'(A,1P,2E12.4)')
     1     ' Nonlin. bifurcation buckling, +(mode 1):BUCPRS =',
     1       BUCPRS,BUCPRU1
         ENDIF
      ENDIF
      IF (ILOADX.GT.2) THEN
         IF (M1MULTB.EQ.-1) THEN
            WRITE(IFILE8,'(A,1P,2E12.4)')
     1     ' Nonlin. bifurcation buckling, -(mode 3):BUCPRS =',
     1       BUCPRS,BUCPRU1
```

```
         ELSE
            WRITE(IFILE8,'(A,1P,2E12.4)')
     1     ' Nonlin. bifurcation buckling, +(mode 3):BUCPRS =',
     1       BUCPRS,BUCPRU1
         ENDIF
       ENDIF
       ENDIF
C      End of (IMODX.EQ.0) condition
C
       IF (IMODX.EQ.1) THEN
C
       IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
       IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) THEN
          IF (M1MULTB.EQ.1) THEN
             WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear bifurcation buckling,+(mode 1) imperfection',
     1'       IMODX=',IMODX
          ELSE
             WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear bifurcation buckling,-(mode 1) imperfection',
     1'       IMODX=',IMODX
          ENDIF
       ENDIF
       IF (ILOADX.EQ.3.OR.ILOADX.EQ.4) THEN
          IF (M1MULTB.EQ.1) THEN
             WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear bifurcation buckling,+(mode 3) imperfection',
     1'       IMODX=',IMODX
          ELSE
             WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear bifurcation buckling,-(mode 3) imperfection',
     1'       IMODX=',IMODX
          ENDIF
       ENDIF
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
       CALL MOVER(WSAVEX,1,WMODEX,1,I2)
       IF (M1MULTB.EQ.-1) THEN
          DO 53 I = 1,I2
           WMODEX(I) = -WSAVEX(I)
    53    CONTINUE
       ENDIF
       INDIC = 1
       IMPERF = 1
C      WRITE(IFILE8,'(A,2I5)')' after 53: IMODX,NWAV1=',IMODX,NWAV1
       N0BX =   NWAV1
       NMINBX = NWAV1
       NMAXBX = NWAV1
       INCRBX = 1
```

```
       PMAX = PMAXBUC1
       IF (M1MULTC.EQ.M1MULTB.AND.CLAPS1(ILOADX).LT.PMAX)
      1   PMAX = 0.9*CLAPS1(ILOADX)
       IF (NPRINX.GT.0) WRITE(IFILE8,'(A,2I5,1P,E12.4)')
      1' IMODX=1: M1MULTB,NWAV1,PMAX=',M1MULTB,NWAV1,PMAX
       CALL BOSDEC(3,ILOADX,INDIC,IMPERF,24,IFILE8,
      1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
      1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
      1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
      1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
       CALL B4READ
       CALL B4MAIN
       IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C      CALL B4POST
       CALL GASP (DUM1,DUM2,-2,DUM3)
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
       IF (NPRINX.GT.0) THEN
          IF (ILOADX.LE.2) THEN
             IF (M1MULTB.EQ.1) THEN
                WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
      1       ' **** Nonlinear overall bifurcation buckling results ***',
      1       ' Overall buckling, +(mode 1) imperfection shape;',
      1       ' Applied pressure, PMAX =', PMAX
                WRITE(IFILE8,'(A,I2,A)')
      1       ' Output from +(mode 1) INDIC=1, buckling analysis;IMODX=',
      1         IMODX,' ***'
             ELSE
                WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
      1       ' **** Nonlinear overall bifurcation buckling results ***',
      1       ' Overall buckling, -(mode 1) imperfection shape;',
      1       ' Applied pressure, PMAX =', PMAX
                WRITE(IFILE8,'(A,I2,A)')
      1       ' Output from -(mode 1) INDIC=1, buckling analysis;IMODX=',
      1         IMODX,' ***'
             ENDIF
          ENDIF
          IF (ILOADX.GT.2) THEN
             IF (M1MULTB.EQ.1) THEN
                WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
      1       ' **** Nonlinear overall bifurcation buckling results ***',
      1       ' Overall buckling, +(mode 3) imperfection shape;',
      1       ' Applied pressure, PMAX =', PMAX
                WRITE(IFILE8,'(A,I2,A)')
      1       ' Output from +(mode 3) INDIC=1, buckling analysis;IMODX=',
      1         IMODX,' ***'
             ELSE
                WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
```

```
     1       ' **** Nonlinear overall bifurcation buckling results ***',
     1       ' Overall buckling, -(mode 3) imperfection shape;',
     1       ' Applied pressure, PMAX =', PMAX
               WRITE(IFILE8,'(A,I2,A)')
     1       ' Output from -(mode 3) INDIC=1, buckling analysis;IMODX=',
     1         IMODX,' ***'
           ENDIF
         ENDIF
         WRITE(IFILE8, '(/,A,/,A,1P,E12.4,A,I5,/,A)')
     1 ' **** CRITICAL EIGENVALUE AND WAVENUMBER ****',
     1 ' EIGCRT=',EIGCRT,'; NO. OF CIRC. WAVES, NWVCRT=',NWVCRT,
     1 ' ************************************************'
         WRITE(IFILE8,'(/,A,/,A,/,A)')
     1 ' ***** EIGENVALUES AND MODE SHAPES *****',
     1'  EIGENVALUE(CIRC. WAVES)',
     1 ' ======================================='
         DO 54 I = 1,IWAVEB
           WRITE(IFILE8,'(A,1P,E12.4,A,I4,A)')
     1 '  ',EIGCOM(I),'(',NWVCOM(I),')'
   54    CONTINUE
         WRITE(IFILE8,'(A)')
     1 ' ======================================='
C
         WRITE(IFILE8, '(/,A,/,A,1P,E12.4,A,I5,/,A)')
     1 ' **** CRITICAL NEGATIVE EIGENVALUE AND WAVENUMBER ****',
     1 ' EIGCRN=',EIGCRN,'; NO. OF CIRC. WAVES, NWVCRN=',NWVCRN,
     1 ' ************************************************'
         WRITE(IFILE8,'(/,A,/,A,/,A)')
     1 ' ***** NEGATIVE EIGENVALUES AND MODE SHAPES *****',
     1'  EIGENVALUE(CIRC. WAVES)',
     1 ' ======================================='
         DO 5401 I = 1,IWAVEB
           WRITE(IFILE8,'(A,1P,E12.4,A,I4,A)')
     1 '  ',EIGNEG(I),'(',NWVNEG(I),')'
 5401    CONTINUE
         WRITE(IFILE8,'(A)')
     1 ' ======================================='
C
      ENDIF
C     End of (NPRINX.GT.0) condition
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (EIGCRN.LT.0.0) THEN
C BEG MAR 2008
         BUCTST =  PMAX + EIGCRN*PMAX/1000.
C        Disregard the negative eigenvalue if it is too large.
         IF (BUCTST.LE.0.) THEN
            EIGCRN = EIGCRT
```

```
            NWVCRN = NWVCRT
          ENDIF
C END MAR 2008
          EIGCRT = EIGCRN
          NWVCRT = NWVCRN
        ENDIF
        BUCPRS = PMAX + EIGCRT*PMAX/1000.
        GENBK1(ILOADX) = BUCPRS/PRESS(ILOADX)
        IF (NPRINX.GT.0) THEN
          WRITE(IFILE8,'(/,A,1P,E12.4,A,I2,A,/,A,1P,E12.4,/,/)')
     1 ' Nonlinear bifurcation buckling pressure, BUCPRS(circ.waves)=',
     1   BUCPRS,'(',NWVCRT,')',
     1 ' General bifurcation buckling load factor, GENBK1(ILOADX)=',
     1   GENBK1(ILOADX)
C
          WRITE(IFILE8,'(/,A,A)')
     1 '                                                    PERTURBED',
     1 '  UNPERTURBED'
          IF (ILOADX.LE.2) THEN
            IF (M1MULTB.EQ.1) THEN
              WRITE(IFILE8,'(A,1P,2E12.4)')
     1        ' Nonlin. bifurcation buckling, +(mode 1):BUCPRS =',
     1          BUCPRS,BUCPRU1
            ELSE
              WRITE(IFILE8,'(A,1P,2E12.4)')
     1        ' Nonlin. bifurcation buckling, -(mode 1):BUCPRS =',
     1          BUCPRS,BUCPRU1
            ENDIF
          ENDIF
          IF (ILOADX.GT.2) THEN
            IF (M1MULTB.EQ.1) THEN
              WRITE(IFILE8,'(A,1P,2E12.4)')
     1        ' Nonlin. bifurcation buckling, +(mode 3):BUCPRS =',
     1          BUCPRS,BUCPRU1
            ELSE
              WRITE(IFILE8,'(A,1P,2E12.4)')
     1        ' Nonlin. bifurcation buckling, -(mode 3):BUCPRS =',
     1          BUCPRS,BUCPRU1
            ENDIF
          ENDIF
        ENDIF
C     End of (NPRINX.GT.0) condition
C
C     CALL EXIT
C
        ENDIF
C     End of (IMODX.EQ.1) condition
C
```

```
C  Find nonlinear bifurcation buckling of imperfect shell, +(mode 2)
C
      IF (IMODX.EQ.0) THEN
C
      IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
      IF (ILOADX.EQ.1) WRITE(IFILE8,'(A,I2,A,/,A,A,I2)')
     1' ============ Analysis No. 7 for Load Set No.',ILOADX,
     1' ============',
     1' ** Start nonlinear bifurcation buckling,+(mode 2) imperfection',
     1'      IMODX=',IMODX
      IF (ILOADX.EQ.3) WRITE(IFILE8,'(A,I2,A,/,A,A,I2)')
     1' ============ Analysis No. 7 for Load Set No.',ILOADX,
     1' ============',
     1' ** Start nonlinear bifurcation buckling,+(mode 4) imperfection',
     1'      IMODX=',IMODX
      IF (ILOADX.EQ.2) WRITE(IFILE8,'(A,I2,A,/,A,A,I2)')
     1' ============ Analysis No. 7 for Load Set No.',ILOADX,
     1' ============',
     1' ** Start nonlinear bifurcation buckling,-(mode 2) imperfection',
     1'      IMODX=',IMODX
      IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,I2,A,/,A,A,I2)')
     1' ============ Analysis No. 7 for Load Set No.',ILOADX,
     1' ============',
     1' ** Start nonlinear bifurcation buckling,-(mode 4) imperfection',
     1'      IMODX=',IMODX
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
      IF (ILOADX.EQ.1.OR.ILOADX.EQ.3) THEN
C
      CALL MOVER(WMODX2,1,WMODEX,1,I2)
      INDIC = 1
      IMPERF = 1
      PMAX02PB = PMAX02P
      PMAX = PMAX02PB
      IF (PMAX.LT.PRESS(ILOADX)) PMAX = 0.9*PMAX02PB
      IF (PMAX.GT.0.9*CLAPS2(ILOADX)) PMAX = 0.9*CLAPS2(ILOADX)
      N0BX = 0
      NMINBX = 0
      NMAXBX = 10
      IF (ITYPEX.EQ.2) NMAXBX = 30
      INCRBX = 1
 5500 CONTINUE
      CALL BOSDEC(3,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C      Get CASE.ALL9P file for input for BIGBOSOR4...
```

```
C
C      CASE.ALL9P is an input file for BIGBOSOR4 for nonlinear bifurcation
C      buckling analysis of the axisymmetrically imperfect ellipsoidal
shell
C      with a +(mode 2) imperfection shape if Load set no. ILOADX=2 .
C
C      CASE.ALL9P3 is an input file for BIGBOSOR4 for nonlinear
bifurcation
C      buckling analysis of the axisymmetrically imperfect ellipsoidal
shell
C      with a +(mode 4) imperfection shape if Load set no. ILOADX=4 .
C
      IF (ITYPEX.EQ.2) THEN
         I=INDEX(CASE,' ')
         IF (ILOADX.EQ.1) THEN
            IF(I.NE.0) THEN
               CASA9=CASE(:I-1)//'.ALL9P'
            ELSE
               CASA9=CASE//'.ALL9P'
            ENDIF
         ENDIF
         IF (ILOADX.EQ.3) THEN
            IF(I.NE.0) THEN
               CASA9=CASE(:I-1)//'.ALL9P3'
            ELSE
               CASA9=CASE//'.ALL9P3'
            ENDIF
         ENDIF
         OPEN(UNIT=69,FILE=CASA9,STATUS='UNKNOWN')
         CALL BOSDEC(3,ILOADX,INDIC,IMPERF,69,IFILE8,
     1                 npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                 WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                 THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                 PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
         CLOSE(UNIT=69)
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, bifurcation buckling, +(mode 2) imperf.=',
     1   CASA9
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, bifurcation buckling, +(mode 4) imperf.=',
     1   CASA9
       ENDIF
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
       CALL B4READ
       CALL B4MAIN
       IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C      CALL B4POST
```

```
      CALL GASP (DUM1,DUM2,-2,DUM3)
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (NPRINX.GT.0) THEN
         IF (ILOADX.EQ.1) THEN
            WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
     1     ' ***** Nonlinear overall bifurcation buckling results *****',
     1     ' Overall buckling, +(mode 2) imperfection shape;',
     1     ' Applied pressure, PMAX =', PMAX
            WRITE(IFILE8,'(A,I2,A)')
     1     ' ** Output from +(mode 2) INDIC=1, buckling analysis;IMODX=',
     1       IMODX,' *********'
         ENDIF
         IF (ILOADX.EQ.3) THEN
            WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
     1     ' ***** Nonlinear overall bifurcation buckling results *****',
     1     ' Overall buckling, +(mode 4) imperfection shape;',
     1     ' Applied pressure, PMAX =', PMAX
            WRITE(IFILE8,'(A,I2,A)')
     1     ' ** Output from +(mode 4) INDIC=1, buckling analysis;IMODX=',
     1       IMODX,' *********'
         ENDIF
         WRITE(IFILE8, '(/,A,/,A,1P,E12.4,A,I5,/,A)')
     1 ' **** CRITICAL EIGENVALUE AND WAVENUMBER ****',
     1 ' EIGCRT=',EIGCRT,'; NO. OF CIRC. WAVES, NWVCRT=',NWVCRT,
     1 ' ************************************************'
         WRITE(IFILE8,'(/,A,/,A,/,A)')
     1 ' ***** EIGENVALUES AND MODE SHAPES *****',
     1'  EIGENVALUE(CIRC. WAVES)',
     1 ' ======================================='
         DO 55 I = 1,IWAVEB
           WRITE(IFILE8,'(A,1P,E12.4,A,I4,A)')
     1 '  ',EIGCOM(I),'(',NWVCOM(I),')'
   55    CONTINUE
         WRITE(IFILE8,'(A)')
     1 ' ======================================='
C
         WRITE(IFILE8, '(/,A,/,A,1P,E12.4,A,I5,/,A)')
     1 ' **** CRITICAL NEGATIVE EIGENVALUE AND WAVENUMBER ****',
     1 ' EIGCRN=',EIGCRN,'; NO. OF CIRC. WAVES, NWVCRN=',NWVCRN,
     1 ' ************************************************'
         WRITE(IFILE8,'(/,A,/,A,/,A)')
     1 ' ***** NEGATIVE EIGENVALUES AND MODE SHAPES *****',
     1'  EIGENVALUE(CIRC. WAVES)',
     1 ' ======================================='
         DO 5501 I = 1,IWAVEB
           WRITE(IFILE8,'(A,1P,E12.4,A,I4,A)')
     1 '  ',EIGNEG(I),'(',NWVNEG(I),')'
```

```
 5501    CONTINUE
         WRITE(IFILE8,'(A)')
     1 ' ======================================='
C
      ENDIF
C     End of (NPRINX.GT.0) condition
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (EIGCRN.LT.0.0) THEN
C BEG MAR 2008
         BUCTST =  PMAX + EIGCRN*PMAX/1000.
C        Disregard the negative eigenvalue if it is too large.
         IF (BUCTST.LE.0.) THEN
            EIGCRN = EIGCRT
            NWVCRN = NWVCRT
         ENDIF
C END MAR 2008
         BUCPRSP = PMAX + EIGCRN*PMAX/1000.
C BEG MAR 2008
C        IF (BUCPRSP.GT.0.) THEN
            EIGCRT = EIGCRN
            NWVCRT = NWVCRN
C        WRITE(IFILE8,'(/,A,/,A,/,A,/,A,1P,E12.4,/,A,1P,E12.4,/,A,/,A)')
C    1    ' ******* WARNING ******* WARNING ******* WARNING *******',
C    1    ' There are negative eigenvalues. The nonlinear bifurcation',
C    1    ' buckling load is less than the applied load, PMAX:',
C    1    ' Buckling load, BUCPRSP=',BUCPRSP,
C    1    ' Applied  load, PMAX   =',PMAX,
C    1    ' PMAX is being reset equal to 0.9*BUCPRSP and a new',
C    1    ' nonlinear bifurcation buckling load computed.'
C           PMAX = 0.9*BUCPRSP
C           PMAX02PB = BUCPRSP
C           GO TO 5500
C        ENDIF
C END MAR 2008
      ENDIF
      EIG2P  = EIGCRT
      NWAV2P = NWVCRT
      BUCPRSP = PMAX + EIGCRT*PMAX/1000.
      GENBK2(ILOADX) = BUCPRSP/PRESS(ILOADX)
      IF (NPRINX.GT.0) THEN
         WRITE(IFILE8,'(/,A,1P,E12.4,A,I2,A,/,A,1P,E12.4,/,/)')
     1 ' Nonlinear bifurcation buckling pressure, BUCPRSP(circ.waves)=',
     1   BUCPRSP,'(',NWVCRT,')',
     1 ' General bifurcation buckling load factor, GENBK2(ILOADX)=',
     1   GENBK2(ILOADX)
C
         BUCPRU2 = BUCPRSP
```

```
         WRITE(IFILE8,'(/,A,A)')
     1 '                                                    PERTURBED',
     1 '  UNPERTURBED'
         IF (ILOADX.EQ.1) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Nonlin. bifurcation buckling, +(mode 2):BUCPRSP=',
     1   BUCPRSP,BUCPRU2
         IF (ILOADX.EQ.3) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Nonlin. bifurcation buckling, +(mode 4):BUCPRSP=',
     1   BUCPRSP,BUCPRU2
       ENDIF
C
C BEG MAR 2008
C     WRITE(IFILE8,'(/,A,I2,1P3E14.6)')
C    1' +mode 2, ITYPEX,BUCPRSP,PPLUS2,EIGCRT=',
C    1           ITYPEX,BUCPRSP,PPLUS2,EIGCRT
C     IF (ITYPEX.EQ.2.AND.BUCPRSP.LT.0.980*PPLUS2.AND.
C    1    EIGCRT.GT.0.0.AND.(EIGCRT/1000.0).GE.0.02
C    1                .AND.(EIGCRT/1000.0).LT.0.25) THEN
       IF (ITYPEX.EQ.2.AND.EIGCRT.GT.0.0
     1              .AND.(EIGCRT/1000.0).GE.0.02) THEN
         INDIC = -2
         N0BX = NWVCRT
         NMINBX = NWVCRT
         NMAXBX = NWVCRT
         CALL BOSDEC(3,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                 npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                 WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                 THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                 PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C
         CALL B4READ
         CALL B4MAIN
         IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C        CALL B4POST
         CALL GASP (DUM1,DUM2,-2,DUM3)
         IF (PPP.LT.BUCPRSP.AND.PPP.LT.0.98*PPLUS2.AND.IDETCT.EQ.1) THEN
            BUCPRSP = PPP
            GENBK2(ILOADX) = BUCPRSP/PRESS(ILOADX)
         ENDIF
         IF (PPP.GE.PPLUS2) WRITE(IFILE8,'(/,A,/,A,/,A,I3,A,/,A)')
     1 ' ***************** INDIC=-2 analysis yields ***************',
     1 ' SHELL COLLAPSES AXISYMMETRICALLY BEFORE NONLINEAR BIFURCA-',
     1 ' TION BUCKLING WITH N =',NWVCRT,' CIRCUMFERENTIAL WAVES.',
     1 ' ************************************************************'
         IF (IDETCT.EQ.1)
     1   WRITE(IFILE8,'(/,A,/,A,1PE12.4,/,A,1PE12.4,/,A,/,A)')
     1 ' ***************** INDIC=-2 analysis yields ***************',
     1 ' Critical pressure from INDIC=-2 analysis        =',PPP,
```

```
     1 ' Nonlinear buckling margin from INDIC=-2 analysis=',
     1   GENBK2(ILOADX)-1.0,
     1 ' NOTE: This margin may differ from that obtained with ITYPE=1',
     1 ' ************************************************************'
C23456789012345678901234567890123456789012345678901234567890123456789012
      ENDIF
C END MAR 2008
C
C     CALL EXIT
C
      BUCPRS = BUCPRSP
      PMAXBUC2 = PMAX02PB
      NWAV2 = NWAV2P
      EIG2  = EIG2P
C
      ENDIF
C     End of (ILOADX.EQ.1.OR.ILOADX.EQ.3) condition
C
      IF ((NCASES.EQ.1.AND.ILOADX.EQ.1)
     1               .OR.ILOADX.EQ.2.OR.ILOADX.EQ.4) THEN
C
C  Find nonlinear bifurcation buckling of imperfect shell, -(mode 2)
C
      IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
      IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear bifurcation buckling,-(mode 2) imperfection',
     1'      IMODX=',IMODX
      IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear bifurcation buckling,-(mode 4) imperfection',
     1'      IMODX=',IMODX
C23456789012345678901234567890123456789012345678901234567890123456789012
      DO 56 I = 1,I2
       WMODEX(I) = -WMODX2(I)
   56 CONTINUE
      INDIC = 1
      IMPERF = 1
      PMAX02MB = PMAX02M
      PMAX = PMAX02MB
      IF (PMAX.LT.PRESS(ILOADX)) PMAX = 0.9*PMAX02MB
      IF (PMAX.GT.0.9*CLAPS2(ILOADX)) PMAX = 0.9*CLAPS2(ILOADX)
      N0BX = 0
      NMINBX = 0
      NMAXBX = 10
      IF (ITYPEX.EQ.2) NMAXBX = 30
      INCRBX = 1
 5600 CONTINUE
      CALL BOSDEC(3,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                 npoint,ainput,binput,LENCYL,nodes,WIMP,
```

```
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C      Get CASE.ALL9N file for input for BIGBOSOR4...
C
C      CASE.ALL9N is an input file for BIGBOSOR4 for nonlinear bifurcation
C      buckling analysis of the axisymmetrically imperfect ellipsoidal
shell
C      with a -(mode 2) imperfection shape if Load set no. ILOADX=2 .
C
C      CASE.ALL9N3 is an input file for BIGBOSOR4 for nonlinear
bifurcation
C      buckling analysis of the axisymmetrically imperfect ellipsoidal
shell
C      with a -(mode 4) imperfection shape if Load set no. ILOADX=4 .
C
      IF (ITYPEX.EQ.2) THEN
         I=INDEX(CASE,' ')
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) THEN
            IF(I.NE.0) THEN
               CASA9=CASE(:I-1)//'.ALL9N'
            ELSE
               CASA9=CASE//'.ALL9N'
            ENDIF
         ENDIF
         IF (ILOADX.EQ.4) THEN
            IF(I.NE.0) THEN
               CASA9=CASE(:I-1)//'.ALL9N3'
            ELSE
               CASA9=CASE//'.ALL9N3'
            ENDIF
         ENDIF
         OPEN(UNIT=69,FILE=CASA9,STATUS='UNKNOWN')
         CALL BOSDEC(3,ILOADX,INDIC,IMPERF,69,IFILE8,
     1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
         CLOSE(UNIT=69)
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, bifurcation buckling, -(mode 2) imperf.=',
     1   CASA9
         IF (ILOADX.EQ.4) WRITE(IFILE8,'(A)')
     1' BIGBOSOR4 input file, bifurcation buckling, -(mode 4) imperf.=',
     1   CASA9
      ENDIF
C23456789012345678901234567890123456789012345678901234567890123456789012
C
```

```
      CALL B4READ
      CALL B4MAIN
      IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C     CALL B4POST
      CALL GASP (DUM1,DUM2,-2,DUM3)
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (NPRINX.GT.0) THEN
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2)
     1   WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
     1 ' ******* Nonlinear overall bifurcation buckling results ******',
     1 ' Overall buckling, -(mode 2) imperfection shape;',
     1 ' Applied pressure, PMAX =', PMAX
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,I2,A)')
     1 ' *** Output from -(mode 2) INDIC=1, buckling analysis; IMODX=',
     1   IMODX,' *********'
         IF (ILOADX.EQ.4)
     1   WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
     1 ' ******* Nonlinear overall bifurcation buckling results ******',
     1 ' Overall buckling, -(mode 4) imperfection shape;',
     1 ' Applied pressure, PMAX =', PMAX
         IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,I2,A)')
     1 ' *** Output from -(mode 4) INDIC=1, buckling analysis; IMODX=',
     1   IMODX,' *********'
         WRITE(IFILE8, '(/,A,/,A,1P,E12.4,A,I5,/,A)')
     1 ' **** CRITICAL EIGENVALUE AND WAVENUMBER ****',
     1 ' EIGCRT=',EIGCRT,'; NO. OF CIRC. WAVES, NWVCRT=',NWVCRT,
     1 ' ************************************************'
         WRITE(IFILE8,'(/,A,/,A,/,A)')
     1 ' ***** EIGENVALUES AND MODE SHAPES *****',
     1'  EIGENVALUE(CIRC. WAVES)',
     1 ' ======================================='
         DO 57 I = 1,IWAVEB
           WRITE(IFILE8,'(A,1P,E12.4,A,I4,A)')
     1 '  ',EIGCOM(I),'(',NWVCOM(I),')'
   57    CONTINUE
         WRITE(IFILE8,'(A)')
     1 ' ======================================='
C
         WRITE(IFILE8, '(/,A,/,A,1P,E12.4,A,I5,/,A)')
     1 ' **** CRITICAL NEGATIVE EIGENVALUE AND WAVENUMBER ****',
     1 ' EIGCRN=',EIGCRN,'; NO. OF CIRC. WAVES, NWVCRN=',NWVCRN,
     1 ' ************************************************'
         WRITE(IFILE8,'(/,A,/,A,/,A)')
     1 ' ***** NEGATIVE EIGENVALUES AND MODE SHAPES *****',
     1'  EIGENVALUE(CIRC. WAVES)',
     1 ' ======================================='
         DO 5701 I = 1,IWAVEB
```

```
            WRITE(IFILE8,'(A,1P,E12.4,A,I4,A)')
     1 '  ',EIGNEG(I),'(',NWVNEG(I),')'
 5701    CONTINUE
         WRITE(IFILE8,'(A)')
     1 ' ======================================='
C
      ENDIF
C     End of (NPRINX.GT.0) condition
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (EIGCRN.LT.0.0) THEN
C BEG MAR 2008
         BUCTST =  PMAX + EIGCRN*PMAX/1000.
C        Disregard the negative eigenvalue if it is too large.
         IF (BUCTST.LE.0.) THEN
            EIGCRN = EIGCRT
            NWVCRN = NWVCRT
         ENDIF
C END MAR 2008
         BUCPRSM = PMAX + EIGCRN*PMAX/1000.
C BEG MAR 2008
C        IF (BUCPRSM.GT.0.) THEN
            EIGCRT = EIGCRN
            NWVCRT = NWVCRN
C        WRITE(IFILE8,'(/,A,/,A,/,A,/,A,1P,E12.4,/,A,1P,E12.4,/,A,/,A)')
C    1    ' ******* WARNING ******* WARNING ******* WARNING *******',
C    1    ' There are negative eigenvalues. The nonlinear bifurcation',
C    1    ' buckling load is less than the applied load, PMAX:',
C    1    ' Buckling load, BUCPRSM=',BUCPRSM,
C    1    ' Applied  load, PMAX   =',PMAX,
C    1    ' PMAX is being reset equal to 0.9*BUCPRSM and a new',
C    1    ' nonlinear bifurcation buckling load computed.'
C           PMAX = 0.9*BUCPRSM
C           PMAX02MB = BUCPRSM
C           GO TO 5600
C        ENDIF
C END MAR 2008
      ENDIF
      EIG2M  = EIGCRT
      NWAV2M = NWVCRT
      BUCPRSM = PMAX + EIGCRT*PMAX/1000.
      GENBK2(ILOADX) = BUCPRSM/PRESS(ILOADX)
      IF (NPRINX.GT.0) THEN
         WRITE(IFILE8,'(/,A,1P,E12.4,A,I2,A,/,A,1P,E12.4,/,/)')
     1 ' Nonlinear bifurcation buckling pressure, BUCPRSM(circ.waves)=',
     1   BUCPRSM,'(',NWVCRT,')',
     1 ' General bifurcation buckling load factor, GENBK2(ILOADX)=',
     1   GENBK2(ILOADX)
```

```
C
         BUCPRU2 = BUCPRSM
         WRITE(IFILE8,'(/,A,A)')
     1 '                                                    PERTURBED',
     1 '  UNPERTURBED'
         IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Nonlin. bifurcation buckling, -(mode 2):BUCPRSM=',
     1   BUCPRSM,BUCPRU2
         IF (ILOADX.EQ.4) WRITE(IFILE8,'(A,1P,2E12.4)')
     1 ' Nonlin. bifurcation buckling, -(mode 4):BUCPRSM=',
     1   BUCPRSM,BUCPRU2
      ENDIF
C
C BEG MAR 2008
C     WRITE(IFILE8,'(/,A,I2,1P3E14.6)')
C    1' -mode 2, ITYPEX,BUCPRSM,PMINUS2,EIGCRT=',
C    1           ITYPEX,BUCPRSM,PMINUS2,EIGCRT
C     IF (ITYPEX.EQ.2.AND.BUCPRSM.LT.0.980*PMINUS2.AND.
C    1    EIGCRT.GT.0.0.AND.(EIGCRT/1000.0).GE.0.02
C    1                 .AND.(EIGCRT/1000.0).LT.0.25) THEN
      IF (ITYPEX.EQ.2.AND.EIGCRT.GT.0.0
     1               .AND.(EIGCRT/1000.0).GE.0.02) THEN
         INDIC = -2
         N0BX = NWVCRT
         NMINBX = NWVCRT
         NMAXBX = NWVCRT
         CALL BOSDEC(3,ILOADX,INDIC,IMPERF,24,IFILE8,
     1                 npoint,ainput,binput,LENCYL,nodes,WIMP,
     1                 WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
     1                 THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
     1                 PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
C
         CALL B4READ
         CALL B4MAIN
         IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C        CALL B4POST
         CALL GASP (DUM1,DUM2,-2,DUM3)
         IF (PPP.LT.BUCPRSM.AND.PPP.LT.0.98*PMINUS2.AND.IDETCT.EQ.1)THEN
            BUCPRSM = PPP
            GENBK2(ILOADX) = BUCPRSM/PRESS(ILOADX)
         ENDIF
         IF (PPP.GE.PMINUS2) WRITE(IFILE8,'(/,A,/,A,/,A,I3,A,/,A)')
     1 ' ***************** INDIC=-2 analysis yields ***************',
     1 ' SHELL COLLAPSES AXISYMMETRICALLY BEFORE NONLINEAR BIFURCA-',
     1 ' TION BUCKLING WITH N =',NWVCRT,' CIRCUMFERENTIAL WAVES.',
     1 ' ******************************************************'
         IF (IDETCT.EQ.1)
     1   WRITE(IFILE8,'(/,A,/,A,1PE12.4,/,A,1PE12.4,/,A,/,A)')
```

```
     1 ' *************** INDIC=-2 analysis yields **************',
     1 ' Critical pressure from INDIC=-2 analysis        =',PPP,
     1 ' Nonlinear buckling margin from INDIC=-2 analysis=',
     1   GENBK2(ILOADX)-1.0,
     1 ' NOTE: This margin may differ from that obtained with ITYPE=1',
     1 ' ********************************************************'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      ENDIF
C END MAR 2008
C
C     CALL EXIT
C
      BUCPRS = BUCPRSM
      PMAXBUC2 = PMAX02MB
      NWAV2 = NWAV2M
      EIG2  = EIG2M
C
      ENDIF
C     End of ((NCASES.EQ.1.AND.ILOADX.EQ.1)
C    1                    .OR.ILOADX.EQ.2.OR.ILOADX.EQ.4) condition
C
      IF (NCASES.EQ.1) BUCPRS = MIN(BUCPRSP,BUCPRSM)
      BUCPRU2 = BUCPRS
      GENBK2(ILOADX) = BUCPRS/PRESS(ILOADX)
      IF (NCASES.EQ.1) THEN
         IF (BUCPRSM.LT.BUCPRSP) THEN
            M2MULTB = -1
            NWAV2 = NWAV2M
            EIG2  = EIG2M
            PMAXBUC2 = PMAX02MB
         ELSE
            M2MULTB = 1
            NWAV2 = NWAV2P
            EIG2  = EIG2P
            PMAXBUC2 = PMAX02PB
         ENDIF
      ENDIF
C
      WRITE(IFILE8,'(A,2I5,1P,E12.4)')
     1' IMODX=0: M2MULTB,NWAV2,PMAXBUC2=',M2MULTB,NWAV2,PMAXBUC2
      IF (PMAXBUC2.LT.PRESS(ILOADX)) PMAXBUC2 = 0.9*PMAXBUC2
      IF (PMAXBUC2.GT.0.9*CLAPS2(ILOADX)) PMAXBUC2 = 0.9*CLAPS2(ILOADX)
C     WRITE(IFILE8,'(/,A,2I7,1P2E12.4)')
C    1' M2MULTB,NWAV2,EIG2,PMAX=',M2MULTB,NWAV2,EIG2,PMAX
      WRITE(IFILE8,'(/,A,A)')
     1 ' The following quantity is used to generate the behavioral',
     1 ' constraint condition and margin:'
      WRITE(IFILE8,'(A,A)')
```

```
     1 '                                                    PERTURBED',
     1 '  UNPERTURBED'
      IF (ILOADX.LE.2) THEN
         IF (M2MULTB.EQ.-1) THEN
            WRITE(IFILE8,'(A,1P,2E12.4)')
     1     ' Nonlin. bifurcation buckling, -(mode 2):BUCPRS =',
     1       BUCPRS,BUCPRU2
         ELSE
            WRITE(IFILE8,'(A,1P,2E12.4)')
     1     ' Nonlin. bifurcation buckling, +(mode 2):BUCPRS =',
     1       BUCPRS,BUCPRU2
         ENDIF
      ENDIF
      IF (ILOADX.GT.2) THEN
         IF (M2MULTB.EQ.-1) THEN
            WRITE(IFILE8,'(A,1P,2E12.4)')
     1     ' Nonlin. bifurcation buckling, -(mode 4):BUCPRS =',
     1       BUCPRS,BUCPRU2
         ELSE
            WRITE(IFILE8,'(A,1P,2E12.4)')
     1     ' Nonlin. bifurcation buckling, +(mode 4):BUCPRS =',
     1       BUCPRS,BUCPRU2
         ENDIF
      ENDIF
      ENDIF
C     End of (IMODX.EQ.0) condition
C
      IF (IMODX.EQ.1) THEN
C
      IF (NPRINX.GT.0) WRITE(IFILE8,'(/,/,/)')
      IF (ILOADX.EQ.1.OR.ILOADX.EQ.2) THEN
         IF (M2MULTB.EQ.1) THEN
            WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear bifurcation buckling,+(mode 2) imperfection',
     1'       IMODX=',IMODX
         ELSE
            WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear bifurcation buckling,-(mode 2) imperfection',
     1'       IMODX=',IMODX
         ENDIF
      ENDIF
      IF (ILOADX.EQ.3.OR.ILOADX.EQ.4) THEN
         IF (M2MULTB.EQ.1) THEN
            WRITE(IFILE8,'(A,A,I2)')
     1' ** Start nonlinear bifurcation buckling,+(mode 4) imperfection',
     1'       IMODX=',IMODX
         ELSE
            WRITE(IFILE8,'(A,A,I2)')
```

```
     1' ** Start nonlinear bifurcation buckling,-(mode 4) imperfection',
     1'      IMODX=',IMODX
         ENDIF
       ENDIF
C23456789012345678901234567890123456789012345678901234567890123456789012
       CALL MOVER(WMODX2,1,WMODEX,1,I2)
       IF (M2MULTB.EQ.-1) THEN
          DO 58 I = 1,I2
           WMODEX(I) = -WMODX2(I)
   58     CONTINUE
       ENDIF
       INDIC = 1
       IMPERF = 1
       N0BX =   NWAV2
       NMINBX = NWAV2
       NMAXBX = NWAV2
       INCRBX = 1
       PMAX = PMAXBUC2
       IF (M2MULTC.EQ.M2MULTB.AND.CLAPS2(ILOADX).LT.PMAX)
      1   PMAX = 0.9*CLAPS2(ILOADX)
       IF (NPRINX.GT.0) WRITE(IFILE8,'(A,2I5,1P,E12.4)')
      1' IMODX=1: M2MULTB,NWAV2,PMAX=',M2MULTB,NWAV2,PMAX
       CALL BOSDEC(3,ILOADX,INDIC,IMPERF,24,IFILE8,
      1                  npoint,ainput,binput,LENCYL,nodes,WIMP,
      1                  WMODEX,xinput,xlimit,EMATL,NUMATL,DNMATL,
      1                  THKSKN,HIGHST,SPACNG,THSTIF,THKCYL,
      1                  PRESS,PMAX,N0BX,NMINBX,NMAXBX,INCRBX)
       CALL B4READ
       CALL B4MAIN
       IF (NPRINX.GT.0) WRITE(IFILE8,'(A)') WRDCOL
C      CALL B4POST
       CALL GASP (DUM1,DUM2,-2,DUM3)
C
C23456789012345678901234567890123456789012345678901234567890123456789012
       IF (NPRINX.GT.0) THEN
          IF (ILOADX.LE.2) THEN
             IF (M2MULTB.EQ.1) THEN
                WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
      1       ' **** Nonlinear overall bifurcation buckling results ***',
      1       ' Overall buckling, +(mode 2) imperfection shape;',
      1       ' Applied pressure, PMAX =', PMAX
                WRITE(IFILE8,'(A,I2,A)')
      1       ' Output from +(mode 2) INDIC=1, buckling analysis;IMODX=',
      1         IMODX,' ***'
             ELSE
                WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
      1       ' **** Nonlinear overall bifurcation buckling results ***',
      1       ' Overall buckling, -(mode 2) imperfection shape;',
```

```
     1       ' Applied pressure, PMAX =', PMAX
               WRITE(IFILE8,'(A,I2,A)')
     1       ' Output from -(mode 2) INDIC=1, buckling analysis;IMODX=',
     1         IMODX,' ***'
            ENDIF
          ENDIF
          IF (ILOADX.GT.2) THEN
            IF (M2MULTB.EQ.1) THEN
               WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
     1       ' **** Nonlinear overall bifurcation buckling results ***',
     1       ' Overall buckling, +(mode 4) imperfection shape;',
     1       ' Applied pressure, PMAX =', PMAX
               WRITE(IFILE8,'(A,I2,A)')
     1       ' Output from +(mode 4) INDIC=1, buckling analysis;IMODX=',
     1         IMODX,' ***'
            ELSE
               WRITE(IFILE8,'(/,A,/,A,A,1P,E12.4)')
     1       ' **** Nonlinear overall bifurcation buckling results ***',
     1       ' Overall buckling, -(mode 4) imperfection shape;',
     1       ' Applied pressure, PMAX =', PMAX
               WRITE(IFILE8,'(A,I2,A)')
     1       ' Output from -(mode 4) INDIC=1, buckling analysis;IMODX=',
     1         IMODX,' ***'
            ENDIF
          ENDIF
          WRITE(IFILE8, '(/,A,/,A,1P,E12.4,A,I5,/,A)')
     1 ' **** CRITICAL EIGENVALUE AND WAVENUMBER ****',
     1 ' EIGCRT=',EIGCRT,'; NO. OF CIRC. WAVES, NWVCRT=',NWVCRT,
     1 ' ************************************************'
          WRITE(IFILE8,'(/,A,/,A,/,A)')
     1 ' ***** EIGENVALUES AND MODE SHAPES *****',
     1'  EIGENVALUE(CIRC. WAVES)',
     1 ' ======================================'
          DO 59 I = 1,IWAVEB
            WRITE(IFILE8,'(A,1P,E12.4,A,I4,A)')
     1 '  ',EIGCOM(I),'(',NWVCOM(I),')'
    59    CONTINUE
          WRITE(IFILE8,'(A)')
     1 ' ======================================'
C
          WRITE(IFILE8, '(/,A,/,A,1P,E12.4,A,I5,/,A)')
     1 ' **** CRITICAL NEGATIVE EIGENVALUE AND WAVENUMBER ****',
     1 ' EIGCRN=',EIGCRN,'; NO. OF CIRC. WAVES, NWVCRN=',NWVCRN,
     1 ' ************************************************'
          WRITE(IFILE8,'(/,A,/,A,/,A)')
     1 ' ***** NEGATIVE EIGENVALUES AND MODE SHAPES *****',
     1'  EIGENVALUE(CIRC. WAVES)',
     1 ' ======================================'
```

```
         DO 5901 I = 1,IWAVEB
           WRITE(IFILE8,'(A,1P,E12.4,A,I4,A)')
     1 '  ',EIGNEG(I),'(',NWVNEG(I),')'
 5901    CONTINUE
         WRITE(IFILE8,'(A)')
     1 ' ======================================='
C
      ENDIF
C     End of (NPRINX.GT.0) condition
C
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
      IF (EIGCRN.LT.0.0) THEN
C BEG MAR 2008
         BUCTST =  PMAX + EIGCRN*PMAX/1000.
C        Disregard the negative eigenvalue if it is too large.
         IF (BUCTST.LE.0.) THEN
            EIGCRN = EIGCRT
            NWVCRN = NWVCRT
         ENDIF
C END MAR 2008
         EIGCRT = EIGCRN
         NWVCRT = NWVCRN
      ENDIF
      BUCPRS = PMAX + EIGCRT*PMAX/1000.
      GENBK2(ILOADX) = BUCPRS/PRESS(ILOADX)
      IF (NPRINX.GT.0) THEN
         WRITE(IFILE8,'(/,A,1P,E12.4,A,I2,A,/,A,1P,E12.4,/,/)')
     1 ' Nonlinear bifurcation buckling pressure, BUCPRS(circ.waves)=',
     1   BUCPRS,'(',NWVCRT,')',
     1 ' General bifurcation buckling load factor, GENBK2(ILOADX)=',
     1   GENBK2(ILOADX)
C
         WRITE(IFILE8,'(/,A,A)')
     1 '                                                  PERTURBED',
     1 '  UNPERTURBED'
         IF (ILOADX.LE.2) THEN
            IF (M2MULTB.EQ.1) THEN
               WRITE(IFILE8,'(A,1P,2E12.4)')
     1          ' Nonlin. bifurcation buckling, +(mode 2):BUCPRS =',
     1          BUCPRS,BUCPRU2
            ELSE
               WRITE(IFILE8,'(A,1P,2E12.4)')
     1          ' Nonlin. bifurcation buckling, -(mode 2):BUCPRS =',
     1          BUCPRS,BUCPRU2
            ENDIF
         ENDIF
         IF (ILOADX.GT.2) THEN
            IF (M2MULTB.EQ.1) THEN
```

```
                WRITE(IFILE8,'(A,1P,2E12.4)')
     1         ' Nonlin. bifurcation buckling, +(mode 4):BUCPRS =',
     1           BUCPRS,BUCPRU2
              ELSE
                WRITE(IFILE8,'(A,1P,2E12.4)')
     1         ' Nonlin. bifurcation buckling, -(mode 4):BUCPRS =',
     1           BUCPRS,BUCPRU2
              ENDIF
           ENDIF
         ENDIF
C      End of (NPRINX.GT.0) condition
C
C      IF (IDV.EQ.2) CALL EXIT
C
       ENDIF
C      End of (IMODX.EQ.1) condition
C
       IF (NPRINX.GT.0) WRITE(IFILE8,'(/)')
       WRITE(IFILE8,'(A,I2,A,/,/)')
      1' ********* End of all analysis. IMODX=',IMODX,' **************'
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
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 CLAPS1:
C  CLAPS1 = collapse pressure with imperfection mode 1
C
       PHRASE =
      1 'collapse pressure with imperfection mode 1'
       CALL BLANKX(PHRASE,IENDP4)
       IF (IBEHV(1  ).EQ.0) CALL BEHX1
      1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
      1 'collapse pressure with imperfection mode 1')
       IF (CLAPS1(ILOADX  ).EQ.0.)  CLAPS1(ILOADX  ) = 1.E+10
       IF (CLAPS1A(ILOADX  ).EQ.0.)  CLAPS1A(ILOADX  ) = 1.0
       IF (CLAPS1F(ILOADX  ).EQ.0.)  CLAPS1F(ILOADX  ) = 1.0
       KCONX = KCONX + 1
       CARX(KCONX) =CLAPS1(ILOADX  )
       WORDCX= '(CLAPS1('//CIX//')/CLAPS1A('//CIX//
      1  ')) / CLAPS1F('//CIX//')'
       CALL CONX(CLAPS1(ILOADX  ),CLAPS1A(ILOADX  ),CLAPS1F(ILOADX  )
      1,'collapse pressure with imperfection mode 1',
      1 'allowable pressure for axisymmetric collapse',
      1 'factor of safety for axisymmetric collapse',
      1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
      1 WORDMX,PCWORD,CPLOTX,ICARX)
```

```
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  collapse pressure with imperfection mode 1: '
         IENDP4 =46
         CODNAM ='CLAPS1('//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
  130 CONTINUE
  131 CONTINUE
C
C  Behavior and constraints generated next for GENBK1:
C  GENBK1 = general buckling load factor, mode 1
C
      PHRASE =
     1 'general buckling load factor, mode 1'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(2  ).EQ.0) CALL BEHX2
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'general buckling load factor, mode 1')
      IF (GENBK1(ILOADX  ).EQ.0.)  GENBK1(ILOADX  ) = 1.E+10
      IF (GENBK1A(ILOADX  ).EQ.0.)  GENBK1A(ILOADX  ) = 1.0
      IF (GENBK1F(ILOADX  ).EQ.0.)  GENBK1F(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =GENBK1(ILOADX  )
      WORDCX= '(GENBK1('//CIX//')/GENBK1A('//CIX//
     1  ')) / GENBK1F('//CIX//')'
      CALL CONX(GENBK1(ILOADX  ),GENBK1A(ILOADX  ),GENBK1F(ILOADX  )
     1,'general buckling load factor, mode 1',
     1 'allowable general buckling load factor (use 1.0)',
     1 'factor of safety for general buckling',
     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, mode 1: '
         IENDP4 =40
         CODNAM ='GENBK1('//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
  145 CONTINUE
  146 CONTINUE
C
```

```
C  Behavior and constraints generated next for SKNBK1:
C  SKNBK1 = buckling load of skin
C
      IF (JSKNBK1.EQ.0) GO TO 166
      IF (NPRINX.GT.0) THEN
         IF (JSKNBK1.GT.1) THEN
            WRITE(IFILE8,'(1X,A)')'  '
            WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
            WRITE(IFILE8,'(1X,A)')
     1      'number of regions for computing behavior'
         ENDIF
      ENDIF
      DO 165  J=1,JSKNBK1
      CALL CONVR2(J,CJX)
      PHRASE =
     1 'buckling load of skin'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(3  ).EQ.0) CALL BEHX3
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
     1 'buckling load of skin')
      IF (SKNBK1(ILOADX,J).EQ.0.)  SKNBK1(ILOADX,J) = 1.E+10
      IF (SKNBK1A(ILOADX,J).EQ.0.)  SKNBK1A(ILOADX,J) = 1.0
      IF (SKNBK1F(ILOADX,J).EQ.0.)  SKNBK1F(ILOADX,J) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =SKNBK1(ILOADX,J)
      WORDCX= '(SKNBK1('//CIX//','//CJX//')/SKNBK1A('//CIX//','//CJX//
     1  ')) / SKNBK1F('//CIX//','//CJX//')'
      CALL CONX(SKNBK1(ILOADX,J),SKNBK1A(ILOADX,J),SKNBK1F(ILOADX,J)
     1,'buckling load of skin',
     1 'allowable buckling load factor',
     1 'factor of safety for skin buckling',
     1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  buckling load of skin: '
         IENDP4 =25
         CODNAM ='SKNBK1('//CIX//','//CJX//')'
         MLET4 =6 + 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
  165 CONTINUE
  166 CONTINUE
C
C  Behavior and constraints generated next for STFBK1:
C  STFBK1 = buckling load factor, isogrid member, mode 1
```

```
C
      IF (JSKNBK1.EQ.0) GO TO 181
      IF (NPRINX.GT.0) THEN
         IF (JSKNBK1.GT.1) THEN
            WRITE(IFILE8,'(1X,A)')'  '
            WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
            WRITE(IFILE8,'(1X,A)')
     1      'number of regions for computing behavior'
         ENDIF
      ENDIF
      DO 180  J=1,JSKNBK1
      CALL CONVR2(J,CJX)
      PHRASE =
     1 'buckling load factor, isogrid member, mode 1'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(4  ).EQ.0) CALL BEHX4
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
     1 'buckling load factor, isogrid member, mode 1')
      IF (STFBK1(ILOADX,J).EQ.0.)  STFBK1(ILOADX,J) = 1.E+10
      IF (STFBK1A(ILOADX,J).EQ.0.)  STFBK1A(ILOADX,J) = 1.0
      IF (STFBK1F(ILOADX,J).EQ.0.)  STFBK1F(ILOADX,J) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =STFBK1(ILOADX,J)
      WORDCX= '(STFBK1('//CIX//','//CJX//')/STFBK1A('//CIX//','//CJX//
     1  ')) / STFBK1F('//CIX//','//CJX//')'
      CALL CONX(STFBK1(ILOADX,J),STFBK1A(ILOADX,J),STFBK1F(ILOADX,J)
     1,'buckling load factor, isogrid member, mode 1',
     1 'allowable for isogrid stiffener buckling (Use 1.)',
     1 'factor of safety for isogrid stiffener buckling',
     1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  buckling load factor, isogrid member, mode 1: '
         IENDP4 =48
         CODNAM ='STFBK1('//CIX//','//CJX//')'
         MLET4 =6 + 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
  180 CONTINUE
  181 CONTINUE
C
C  Behavior and constraints generated next for SKNST1:
C  SKNST1 = maximum stress in the shell skin, mode 1
C
      IF (JSKNBK1.EQ.0) GO TO 196
```

```
      IF (NPRINX.GT.0) THEN
         IF (JSKNBK1.GT.1) THEN
            WRITE(IFILE8,'(1X,A)')'  '
            WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
            WRITE(IFILE8,'(1X,A)')
     1      'number of regions for computing behavior'
         ENDIF
      ENDIF
      DO 195  J=1,JSKNBK1
      CALL CONVR2(J,CJX)
      PHRASE =
     1 'maximum stress in the shell skin, mode 1'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(5  ).EQ.0) CALL BEHX5
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
     1 'maximum stress in the shell skin, mode 1')
      IF (SKNST1(ILOADX,J).EQ.0.)  SKNST1(ILOADX,J) = 1.E-10
      IF (SKNST1A(ILOADX,J).EQ.0.)  SKNST1A(ILOADX,J) = 1.0
      IF (SKNST1F(ILOADX,J).EQ.0.)  SKNST1F(ILOADX,J) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =SKNST1(ILOADX,J)
      WORDCX= '(SKNST1A('//CIX//','//CJX//')/SKNST1('//CIX//','//CJX//
     1  ')) / SKNST1F('//CIX//','//CJX//')'
      CALL CONX(SKNST1(ILOADX,J),SKNST1A(ILOADX,J),SKNST1F(ILOADX,J)
     1,'maximum stress in the shell skin, mode 1',
     1 'allowable stress for the shell skin',
     1 'factor of safety for skin stress',
     1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  maximum stress in the shell skin, mode 1: '
         IENDP4 =44
         CODNAM ='SKNST1('//CIX//','//CJX//')'
         MLET4 =6 + 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
  195 CONTINUE
  196 CONTINUE
C
C  Behavior and constraints generated next for STFST1:
C  STFST1 = maximum stress in isogrid stiffener, mode 1
C
      IF (JSKNBK1.EQ.0) GO TO 211
      IF (NPRINX.GT.0) THEN
         IF (JSKNBK1.GT.1) THEN
```

```
            WRITE(IFILE8,'(1X,A)')'  '
            WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
            WRITE(IFILE8,'(1X,A)')
     1      'number of regions for computing behavior'
          ENDIF
         ENDIF
         DO 210  J=1,JSKNBK1
         CALL CONVR2(J,CJX)
         PHRASE =
        1 'maximum stress in isogrid stiffener, mode 1'
         CALL BLANKX(PHRASE,IENDP4)
         IF (IBEHV(6  ).EQ.0) CALL BEHX6
        1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
        1 'maximum stress in isogrid stiffener, mode 1')
         IF (STFST1(ILOADX,J).EQ.0.)  STFST1(ILOADX,J) = 1.E-10
         IF (STFST1A(ILOADX,J).EQ.0.)  STFST1A(ILOADX,J) = 1.0
         IF (STFST1F(ILOADX,J).EQ.0.)  STFST1F(ILOADX,J) = 1.0
         KCONX = KCONX + 1
         CARX(KCONX) =STFST1(ILOADX,J)
         WORDCX= '(STFST1A('//CIX//','//CJX//')/STFST1('//CIX//','//CJX//
        1  ')) / STFST1F('//CIX//','//CJX//')'
         CALL CONX(STFST1(ILOADX,J),STFST1A(ILOADX,J),STFST1F(ILOADX,J)
        1,'maximum stress in isogrid stiffener, mode 1',
        1 'allowable stress in isogrid stiffeners',
        1 'factor of safety for stress in isogrid member',
        1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
        1 WORDMX,PCWORD,CPLOTX,ICARX)
         IF (IMODX.EQ.0) THEN
            CODPHR =
        1 '  maximum stress in isogrid stiffener, mode 1: '
            IENDP4 =47
            CODNAM ='STFST1('//CIX//','//CJX//')'
            MLET4 =6 + 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
  210 CONTINUE
  211 CONTINUE
C
C  Behavior and constraints generated next for WAPEX1:
C  WAPEX1 = normal (axial) displacement at apex, mode 1
C
         PHRASE =
        1 'normal (axial) displacement at apex, mode 1'
         CALL BLANKX(PHRASE,IENDP4)
         IF (IBEHV(7  ).EQ.0) CALL BEHX7
        1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
```

```
     1 'normal (axial) displacement at apex, mode 1')
      IF (WAPEX1(ILOADX  ).EQ.0.)  WAPEX1(ILOADX  ) = 1.E-10
      IF (WAPEX1A(ILOADX  ).EQ.0.)  WAPEX1A(ILOADX  ) = 1.0
      IF (WAPEX1F(ILOADX  ).EQ.0.)  WAPEX1F(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =WAPEX1(ILOADX  )
      WORDCX= '(WAPEX1A('//CIX//')/WAPEX1('//CIX//
     1  ')) / WAPEX1F('//CIX//')'
      CALL CONX(WAPEX1(ILOADX  ),WAPEX1A(ILOADX  ),WAPEX1F(ILOADX  )
     1,'normal (axial) displacement at apex, mode 1',
     1 'allowable normal (axial) displacement at apex',
     1 'factor of safety for WAPEX',
     1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  normal (axial) displacement at apex, mode 1: '
         IENDP4 =47
         CODNAM ='WAPEX1('//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
   225 CONTINUE
   226 CONTINUE
C
C  Behavior and constraints generated next for CLAPS2:
C  CLAPS2 = collapse pressure with imperfection mode 2
C
      PHRASE =
     1 'collapse pressure with imperfection mode 2'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(8  ).EQ.0) CALL BEHX8
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'collapse pressure with imperfection mode 2')
      IF (CLAPS2(ILOADX  ).EQ.0.)  CLAPS2(ILOADX  ) = 1.E+10
      IF (CLAPS2A(ILOADX  ).EQ.0.)  CLAPS2A(ILOADX  ) = 1.0
      IF (CLAPS2F(ILOADX  ).EQ.0.)  CLAPS2F(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =CLAPS2(ILOADX  )
      WORDCX= '(CLAPS2('//CIX//')/CLAPS2A('//CIX//
     1  ')) / CLAPS2F('//CIX//')'
      CALL CONX(CLAPS2(ILOADX  ),CLAPS2A(ILOADX  ),CLAPS2F(ILOADX  )
     1,'collapse pressure with imperfection mode 2',
     1 'allowable pressure for axisymmetric collapse',
     1 'factor of safety for axisymmetric collapse',
     1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
```

```
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  collapse pressure with imperfection mode 2: '
         IENDP4 =46
         CODNAM ='CLAPS2('//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
  240 CONTINUE
  241 CONTINUE
C
C  Behavior and constraints generated next for GENBK2:
C  GENBK2 = general buckling load factor, mode 2
C
      PHRASE =
     1 'general buckling load factor, mode 2'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(9  ).EQ.0) CALL BEHX9
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'general buckling load factor, mode 2')
      IF (GENBK2(ILOADX  ).EQ.0.)  GENBK2(ILOADX  ) = 1.E+10
      IF (GENBK2A(ILOADX  ).EQ.0.)  GENBK2A(ILOADX  ) = 1.0
      IF (GENBK2F(ILOADX  ).EQ.0.)  GENBK2F(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =GENBK2(ILOADX  )
      WORDCX= '(GENBK2('//CIX//')/GENBK2A('//CIX//
     1  ')) / GENBK2F('//CIX//')'
      CALL CONX(GENBK2(ILOADX  ),GENBK2A(ILOADX  ),GENBK2F(ILOADX  )
     1,'general buckling load factor, mode 2',
     1 'allowable general buckling load factor (use 1.0)',
     1 'factor of safety for general buckling',
     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, mode 2: '
         IENDP4 =40
         CODNAM ='GENBK2('//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
  255 CONTINUE
  256 CONTINUE
```

```
C
C  Behavior and constraints generated next for SKNBK2:
C  SKNBK2 = local skin buckling load factor, mode 2
C
      IF (JSKNBK2.EQ.0) GO TO 276
      IF (NPRINX.GT.0) THEN
         IF (JSKNBK2.GT.1) THEN
            WRITE(IFILE8,'(1X,A)')'  '
            WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
            WRITE(IFILE8,'(1X,A)')
     1      'number of regions for computing behavior'
         ENDIF
      ENDIF
      DO 275  J=1,JSKNBK2
      CALL CONVR2(J,CJX)
      PHRASE =
     1 'local skin buckling load factor, mode 2'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(10 ).EQ.0) CALL BEHX10
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
     1 'local skin buckling load factor, mode 2')
      IF (SKNBK2(ILOADX,J).EQ.0.)  SKNBK2(ILOADX,J) = 1.E+10
      IF (SKNBK2A(ILOADX,J).EQ.0.)  SKNBK2A(ILOADX,J) = 1.0
      IF (SKNBK2F(ILOADX,J).EQ.0.)  SKNBK2F(ILOADX,J) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =SKNBK2(ILOADX,J)
      WORDCX= '(SKNBK2('//CIX//','//CJX//')/SKNBK2A('//CIX//','//CJX//
     1  ')) / SKNBK2F('//CIX//','//CJX//')'
      CALL CONX(SKNBK2(ILOADX,J),SKNBK2A(ILOADX,J),SKNBK2F(ILOADX,J)
     1,'local skin buckling load factor, mode 2',
     1 'allowable skin buckling load factor (use 1.0)',
     1 'factor of safety for local skin buckling',
     1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  local skin buckling load factor, mode 2: '
         IENDP4 =43
         CODNAM ='SKNBK2('//CIX//','//CJX//')'
         MLET4 =6 + 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
  275 CONTINUE
  276 CONTINUE
C
C  Behavior and constraints generated next for STFBK2:
```

```
C  STFBK2 = buckling load factor for isogrid member
C
      IF (JSKNBK2.EQ.0) GO TO 291
      IF (NPRINX.GT.0) THEN
         IF (JSKNBK2.GT.1) THEN
            WRITE(IFILE8,'(1X,A)')'  '
            WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
            WRITE(IFILE8,'(1X,A)')
     1      'number of regions for computing behavior'
         ENDIF
      ENDIF
      DO 290  J=1,JSKNBK2
      CALL CONVR2(J,CJX)
      PHRASE =
     1 'buckling load factor for isogrid member'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(11 ).EQ.0) CALL BEHX11
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
     1 'buckling load factor for isogrid member')
      IF (STFBK2(ILOADX,J).EQ.0.)  STFBK2(ILOADX,J) = 1.E+10
      IF (STFBK2A(ILOADX,J).EQ.0.)  STFBK2A(ILOADX,J) = 1.0
      IF (STFBK2F(ILOADX,J).EQ.0.)  STFBK2F(ILOADX,J) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =STFBK2(ILOADX,J)
      WORDCX= '(STFBK2('//CIX//','//CJX//')/STFBK2A('//CIX//','//CJX//
     1  ')) / STFBK2F('//CIX//','//CJX//')'
      CALL CONX(STFBK2(ILOADX,J),STFBK2A(ILOADX,J),STFBK2F(ILOADX,J)
     1,'buckling load factor for isogrid member',
     1 'allowable for isogrid stiffener buckling (Use 1.)',
     1 'factor of safety for isogrid stiffener buckling',
     1 2,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  buckling load factor for isogrid member: '
         IENDP4 =43
         CODNAM ='STFBK2('//CIX//','//CJX//')'
         MLET4 =6 + 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
  290 CONTINUE
  291 CONTINUE
C
C  Behavior and constraints generated next for SKNST2:
C  SKNST2 = maximum stress in the shell skin, mode 2
C
```

```
      IF (JSKNBK2.EQ.0) GO TO 306
      IF (NPRINX.GT.0) THEN
         IF (JSKNBK2.GT.1) THEN
            WRITE(IFILE8,'(1X,A)')'  '
            WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
            WRITE(IFILE8,'(1X,A)')
     1      'number of regions for computing behavior'
         ENDIF
      ENDIF
      DO 305  J=1,JSKNBK2
      CALL CONVR2(J,CJX)
      PHRASE =
     1 'maximum stress in the shell skin, mode 2'
      CALL BLANKX(PHRASE,IENDP4)
      IF (IBEHV(12 ).EQ.0) CALL BEHX12
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
     1 'maximum stress in the shell skin, mode 2')
      IF (SKNST2(ILOADX,J).EQ.0.)  SKNST2(ILOADX,J) = 1.E-10
      IF (SKNST2A(ILOADX,J).EQ.0.)  SKNST2A(ILOADX,J) = 1.0
      IF (SKNST2F(ILOADX,J).EQ.0.)  SKNST2F(ILOADX,J) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =SKNST2(ILOADX,J)
      WORDCX= '(SKNST2A('//CIX//','//CJX//')/SKNST2('//CIX//','//CJX//
     1  ')) / SKNST2F('//CIX//','//CJX//')'
      CALL CONX(SKNST2(ILOADX,J),SKNST2A(ILOADX,J),SKNST2F(ILOADX,J)
     1,'maximum stress in the shell skin, mode 2',
     1 'allowable stress for the shell skin',
     1 'factor of safety for skin stress',
     1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  maximum stress in the shell skin, mode 2: '
         IENDP4 =44
         CODNAM ='SKNST2('//CIX//','//CJX//')'
         MLET4 =6 + 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
  305 CONTINUE
  306 CONTINUE
C
C  Behavior and constraints generated next for STFST2:
C  STFST2 = maximum stress in isogrid stiffener, mode 2
C
      IF (JSKNBK2.EQ.0) GO TO 321
      IF (NPRINX.GT.0) THEN
```

```
         IF (JSKNBK2.GT.1) THEN
            WRITE(IFILE8,'(1X,A)')'  '
            WRITE(IFILE8,'(1X,A,$)')' BEHAVIOR OVER J = '
            WRITE(IFILE8,'(1X,A)')
     1      'number of regions for computing behavior'
         ENDIF
       ENDIF
       DO 320  J=1,JSKNBK2
       CALL CONVR2(J,CJX)
       PHRASE =
     1 'maximum stress in isogrid stiffener, mode 2'
       CALL BLANKX(PHRASE,IENDP4)
       IF (IBEHV(13 ).EQ.0) CALL BEHX13
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX,J,
     1 'maximum stress in isogrid stiffener, mode 2')
       IF (STFST2(ILOADX,J).EQ.0.)  STFST2(ILOADX,J) = 1.E-10
       IF (STFST2A(ILOADX,J).EQ.0.)  STFST2A(ILOADX,J) = 1.0
       IF (STFST2F(ILOADX,J).EQ.0.)  STFST2F(ILOADX,J) = 1.0
       KCONX = KCONX + 1
       CARX(KCONX) =STFST2(ILOADX,J)
       WORDCX= '(STFST2A('//CIX//','//CJX//')/STFST2('//CIX//','//CJX//
     1  ')) / STFST2F('//CIX//','//CJX//')'
       CALL CONX(STFST2(ILOADX,J),STFST2A(ILOADX,J),STFST2F(ILOADX,J)
     1,'maximum stress in isogrid stiffener, mode 2',
     1 'allowable stress in isogrid stiffeners',
     1 'factor of safety for stress in isogrid member',
     1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
       IF (IMODX.EQ.0) THEN
          CODPHR =
     1 '  maximum stress in isogrid stiffener, mode 2: '
          IENDP4 =47
          CODNAM ='STFST2('//CIX//','//CJX//')'
          MLET4 =6 + 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
  320 CONTINUE
  321 CONTINUE
C
C  Behavior and constraints generated next for WAPEX2:
C  WAPEX2 = normal (axial) displacement at apex, mode 2
C
       PHRASE =
     1 'normal (axial) displacement at apex, mode 2'
       CALL BLANKX(PHRASE,IENDP4)
       IF (IBEHV(14 ).EQ.0) CALL BEHX14
```

```
     1 (IFILE8,NPRINX,IMODX,IFAST,ILOADX  ,
     1 'normal (axial) displacement at apex, mode 2')
      IF (WAPEX2(ILOADX  ).EQ.0.)  WAPEX2(ILOADX  ) = 1.E-10
      IF (WAPEX2A(ILOADX  ).EQ.0.)  WAPEX2A(ILOADX  ) = 1.0
      IF (WAPEX2F(ILOADX  ).EQ.0.)  WAPEX2F(ILOADX  ) = 1.0
      KCONX = KCONX + 1
      CARX(KCONX) =WAPEX2(ILOADX  )
      WORDCX= '(WAPEX2A('//CIX//')/WAPEX2('//CIX//
     1  ')) / WAPEX2F('//CIX//')'
      CALL CONX(WAPEX2(ILOADX  ),WAPEX2A(ILOADX  ),WAPEX2F(ILOADX  )
     1,'normal (axial) displacement at apex, mode 2',
     1 'allowable normal (axial) displacement at apex',
     1 'factor of safety for WAPEX',
     1 3,INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,WORDCX,
     1 WORDMX,PCWORD,CPLOTX,ICARX)
      IF (IMODX.EQ.0) THEN
         CODPHR =
     1 '  normal (axial) displacement at apex, mode 2: '
         IENDP4 =47
         CODNAM ='WAPEX2('//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
  335 CONTINUE
  336 CONTINUE
C
C  NEXT, EVALUATE THE OBJECTIVE, OBJGEN:
      IF (ILOADX.EQ.1) THEN
         PHRASE ='weight of the equivalent ellipsoidal head'
         CALL BLANKX(PHRASE,IENDP4)
         CALL OBJECT(IFILE8,NPRINX,IMODX,OBJGEN,
     1    'weight of the equivalent ellipsoidal head')
      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/FV01/xinput(21),Ixinpu
      REAL xinput
      COMMON/FV02/ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      REAL ainput,binput,xlimit,SPACNG,THSTIF,THKCYL,RADCYL
      COMMON/FV05/THKSKN(21),HIGHST(21)
      REAL THKSKN,HIGHST
      COMMON/FV16/PRESS(20)
      REAL PRESS
      COMMON/FV19/CLAPS1(20),CLAPS1A(20),CLAPS1F(20)
      REAL CLAPS1,CLAPS1A,CLAPS1F
      COMMON/FV22/GENBK1(20),GENBK1A(20),GENBK1F(20)
      REAL GENBK1,GENBK1A,GENBK1F
      COMMON/FV25/SKNBK1(20,10),JSKNBK1,SKNBK1A(20,10),SKNBK1F(20,10)
      REAL SKNBK1,SKNBK1A,SKNBK1F
      COMMON/FV28/STFBK1(20,10),STFBK1A(20,10),STFBK1F(20,10)
      REAL STFBK1,STFBK1A,STFBK1F
      COMMON/FV31/SKNST1(20,10),SKNST1A(20,10),SKNST1F(20,10)
      REAL SKNST1,SKNST1A,SKNST1F
      COMMON/FV34/STFST1(20,10),STFST1A(20,10),STFST1F(20,10)
      REAL STFST1,STFST1A,STFST1F
      COMMON/FV37/WAPEX1(20),WAPEX1A(20),WAPEX1F(20)
      REAL WAPEX1,WAPEX1A,WAPEX1F
      COMMON/FV40/CLAPS2(20),CLAPS2A(20),CLAPS2F(20)
      REAL CLAPS2,CLAPS2A,CLAPS2F
      COMMON/FV43/GENBK2(20),GENBK2A(20),GENBK2F(20)
      REAL GENBK2,GENBK2A,GENBK2F
      COMMON/FV46/SKNBK2(20,10),JSKNBK2,SKNBK2A(20,10),SKNBK2F(20,10)
      REAL SKNBK2,SKNBK2A,SKNBK2F
      COMMON/FV49/STFBK2(20,10),STFBK2A(20,10),STFBK2F(20,10)
```

```
      REAL STFBK2,STFBK2A,STFBK2F
      COMMON/FV52/SKNST2(20,10),SKNST2A(20,10),SKNST2F(20,10)
      REAL SKNST2,SKNST2A,SKNST2F
      COMMON/FV55/STFST2(20,10),STFST2A(20,10),STFST2F(20,10)
      REAL STFST2,STFST2A,STFST2F
      COMMON/FV58/WAPEX2(20),WAPEX2A(20),WAPEX2F(20)
      REAL WAPEX2,WAPEX2A,WAPEX2F
      COMMON/IV01/npoint,nodes,IMODE
      INTEGER npoint,nodes,IMODE
      COMMON/FV11/LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
      REAL LENCYL,WIMP,EMATL,NUMATL,DNMATL,WEIGHT
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
C=Deck    wall
c=purpose User-written WALL subroutine
c=version May 2002
c=This particular version is for an isogrid-stiffened
c=torispherical or ellipsoidal head optimized by GENOPT.
c=The isogrid stiffeners are internal and smeared.
c=The shell skin is layer 2; the internal isogrid is layer1.
c=The skin thickness and isogrid height vary in the meridional
c=direction only. (Meridional direction=XYs(1) coordinate.)

c#include "keydefs.h"

#if   _usage_
*
*     calling sequence: call WALL ( iunit, ielt, kelt, XYZg, XYs,
```

```
*                                     zeta,  ecz,  ilin, iplas )
*
*     Input Arguments:
*     ================
*     iunit = unit number
*     ielt  = local element number (in unit iunit)
*     kelt  = element type code
*     XYZg  = {x,y,z} global coordinates
*     XYs   = {s,t} surface coordinates (shell unit, only)
*
*     Output Arguments:
*     =================
*     zeta  = zeta (see M-5 or T-3 for details)
*     ecz   = eccentricity (see M-5 or T-3 for details)
*     ilin  = nonlinearity flag
*     iplas = plasticity flag
*
#endif

************************************************************************
            subroutine WALTST( iunit, ielt, kelt, XYZg, XYs,
     &                               zeta,  ecz,  ilin, iplas )
************************************************************************

c     _implicit_none_

      Integer    iunit
      Integer    ielt
      Integer    kelt
      Integer    ilin
      Integer    iplas
      Real       XYZg(3)
      Real       XYs(2)
      Real       zeta
      Real       ecz

      Integer         maxLAY
      PARAMETER     ( maxLAY = 100 )

      Integer         maxSM
      PARAMETER     ( maxSM  = 6 )

      Integer         nit,  not
      common /nitnot/ nit,  not

      Integer         itaw, kwall, nlay, nlip, nsmrs
      common /WALLX / itaw, kwall, nlay, nlip, nsmrs
```

```
 Integer         matL (maxLAY)
 Real            tL   (maxLAY)
 Real            zetL (maxLAY)
 Integer         lsoL (maxLAY)
 Real            e1L  (maxLAY)
 Real            u12L (maxLAY)
 Real            gL   (maxLAY)
 Real            rhoL (maxLAY)
 Real            a1L  (maxLAY)
 Real            e2L  (maxLAY)
 Real            a2L  (maxLAY)
 common /WALL1 / matL, tL,   zetL, lsoL, e1L, u12L,
&                gL,   rhoL, a1L,  e2L,  a2L

 Integer         matF, matM
 Real            ttL   (maxLAY)
 Real            xxL   (maxLAY)
 Real            zetwL (maxLAY)
 Real            oL    (maxLAY)
 Real            eF,   uF,   rhoF,  alF
 Real            eM,   uM,   rhoM,  alM
 common /WALL2 / matF, matM,
&                ttL,  xxL,  zetwL, oL,
&                eF,   uF,   rhoF,  alF,
&                eM,   uM,   rhoM,  alM

 Integer         matC, matS
 Real            ct,   cc,   ch,   cd,   cb
 Real            ts,   phi,  anc
 Real            eC,   uC,   rhoC, alC
 Real            eS,   uS,   rhoS, alS
 common /WALL3 / matC, matS,
&                ct,   cc,   ch,   cd,   cb,
&                ts,   phi,  anc,
&                eC,   uC,   rhoC, alC,
&                eS,   uS,   rhoS, alS

 Integer         ta,   mat,  itvs, idumt
 Real            ccc,        cts
 common /WALL4 / ta,   mat,  itvs, idumt,
&                ccc(6,6),   cts(2,2)

 Integer         icroSM (maxSM)
 Real            spaSM  (maxSM)
 Real            zetSM  (maxSM)
 Real            xsiSM  (maxSM)
 Real            eczSM  (maxSM)
 common /SMEAR / icroSM, spaSM, zetSM, xsiSM, eczSM
```

```
C
      Real SPACNG,EMATL,DNMATL,PHDIFF,XDIFF,RATIO,TDIFF,HDIFF,TATX,HATX
      Real THSKIN,THKSTF,HEIGHT,PHORIG,SARCLT
      Integer I5,NSEG,ISEG,JSEG,I5I,I,IMORE,IMORE1
      COMMON/ISEGX1/PHORIG(100,30),SARCLT(100,30)
      COMMON/ISEGX2/THSKIN(100,30),THKSTF(100,30),HEIGHT(100,30)
      COMMON/ISEGX3/I5(30)
      COMMON/ISEGX4/SPACNG,EMATL,NUMATL,DNMATL
      REAL NUMATL
      CHARACTER*38 WORD1,WORD2,WORD3,WORD4,WORD5,WORD6,WORD7,WORD8
      CHARACTER*2 WORD3B
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
      character       filnam*33
      integer         iw,       ios,      itime
      data            iw    / 61 /
      data            itime / -1 /

c-----------------------------------------------------------------------
c     1st time enter, open the wall thickness file (iw)
c     read the data therein
c     and fill common blocks ISEGX1, ISEGX2, ISEGX3
c-----------------------------------------------------------------------
      if ( itime .lt. 0 ) then

        filnam  =  'WALLTHICK.STAGS'

        open ( unit=iw, name=filnam, access='SEQUENTIAL',
     $         form='FORMATTED', iostat=ios)

        if (ios .ne. 0) then
          write(not,3000) iw, filnam, ios
 3000     format (/,'*****ERROR in routine WALL(---) *****',
     $            /,'tried to open file:  iw = ',I4,'   name = ',A,
     $            /,'error return (iostat) = ',I12,/)
          call exit
        endif
c-----------------------------------------------------------------------
C  Retrieve angle, PHORIG and arc length SARCLT (X-coordinates),
C  shell skin thickness THSKIN, stringer thickness, THKSTF, and
C  stringer height, HEIGHT
C
        WORD1 = '     Number of shell segments (units)='
        WORD2 = '   Isogrid spacing,modulus,nu,density='
        WORD3 = '                Nodal points in Segment'
        WORD3B= '= '
        WORD4 = '                  Angle (X-coordinate)='
        WORD5 = ' Meridional arc length (X-coordinate)='
```

```
        WORD6 = '                  Shell skin thickness='
        WORD7 = '        Stringer (or isogrid)  height='
        WORD8 = '     Stringer (or isogrid)  thickness='
        READ(iw,'(/,A38,I4)') WORD1,NSEG
        READ(iw,'(/,A38,1P,4E14.6)')
     1   WORD2,SPACNG,EMATL,NUMATL,DNMATL
        DO 3 ISEG = 1,NSEG
           READ(iw,'(/,A38,I3,A2,I4)') WORD3,JSEG,WORD3B,I5I
           I5(ISEG) = I5I
           READ(iw,'(/,A38,/(1P5E14.6))') WORD4,(PHORIG(I,ISEG),I=1,I5I)
           READ(iw,'(/,A38,/(1P5E14.6))') WORD5,(SARCLT(I,ISEG),I=1,I5I)
           READ(iw,'(/,A38,/(1P5E14.6))') WORD6,(THSKIN(I,ISEG),I=1,I5I)
           READ(iw,'(/,A38,/(1P5E14.6))') WORD7,(HEIGHT(I,ISEG),I=1,I5I)
           READ(iw,'(/,A38,/(1P5E14.6))') WORD8,(THKSTF(I,ISEG),I=1,I5I)
    3   CONTINUE
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
C
c  Test SUBROUTINE WALL (remove the following statements later)
        rewind iw
        WRITE(not,'(/,A38,I4)')
     1  '     Number of shell segments (units)=',NSEG
        WRITE(not,'(/,A38,1P,4E14.6)')
     1  '   Isogrid spacing,modulus,nu,density=',
     1      SPACNG,EMATL,NUMATL,DNMATL
        DO 20 ISEG = 1,NSEG
           I5I = I5(ISEG)
           WRITE(not,'(/,A38,I3,A2,I4)')
     1'                Nodal points in Segment',ISEG,' =',I5I
           WRITE(not,'(/,A38,/(1P5E14.6))')
     1'                  Angle (X-coordinate)=', (PHORIG(I,ISEG),I=1,I5I)
           WRITE(not,'(/,A38,/(1P5E14.6))')
     1' Meridional arc length (X-coordinate)=', (SARCLT(I,ISEG),I=1,I5I)
           WRITE(not,'(/,A38,/(1P5E14.6))')
     1'                  Shell skin thickness=', (THSKIN(I,ISEG),I=1,I5I)
           WRITE(not,'(/,A38,/(1P5E14.6))')
     1'        Stringer (or isogrid)  height=', (HEIGHT(I,ISEG),I=1,I5I)
           WRITE(not,'(/,A38,/(1P5E14.6))')
     1'     Stringer (or isogrid)  thickness=', (THKSTF(I,ISEG),I=1,I5I)
   20   CONTINUE
C234567890123456789012345678901234567890123456789012345678901234567890123456789012
c
        CLOSE(UNIT=iw)
C
        itime  =  0

      endif
C
c  Fill common block WALLX:
```

```
      itaw = 0
      kwall = 1
      nlay  = 2
      nlip = 0
c     nlip = 5
      nsmrs = 0
c
c  Fill "A" output except ecz (which is a function of X)
      zeta = 0.
      ilin = 0
c     iplas = 1
      iplas = 0
c
c  Find thickness, stiffener height at shell coordinate, X:
c  thickness at X = TATX;  stiffener height at X = HATX
      I5I = I5(iunit)
      DO 10 I = 2,I5I
         IF (XYs(1).LT.PHORIG(I,iunit)) THEN
            IMORE = I
            GO TO 11
         ENDIF
   10 CONTINUE
   11 CONTINUE
      IMORE1 = IMORE - 1
      PHDIFF = PHORIG(IMORE,iunit) - PHORIG(IMORE1,iunit)
      XDIFF  = XYs(1) - PHORIG(IMORE1,iunit)
      RATIO  = XDIFF/PHDIFF
      TDIFF  = THSKIN(IMORE,iunit) - THSKIN(IMORE1,iunit)
      HDIFF  = HEIGHT(IMORE,iunit) - HEIGHT(IMORE1,iunit)
      TATX   = THSKIN(IMORE1,iunit) + RATIO*TDIFF
      HATX   = HEIGHT(IMORE1,iunit) + RATIO*HDIFF
c
c   Find ecz
      ecz = (TATX + HATX)/2. - HATX
c
c   Fill common block WALL1
      matL(1) = 0
      matL(2) = 0
c     matL(1) = 2
c     matL(2) = 1
      tL(2)   = TATX
      tL(1)   = HATX
      zetL(1) = 0.
      zetL(2) = 0.
      lsoL(1) = 0
      lsoL(2) = 0
      e1L(2)  = EMATL
      e1L(1)  = EMATL*THKSTF(1,iunit)/SPACNG
```

```
      u12L(2) = NUMATL
      u12L(1) = 1./3.
      gL(2)   = EMATL/(2.*(1.+NUMATL))
      gL(1)   = e1L(1)/(2.*(1.+u12L(1)))
      rhoL(2) = DNMATL
      rhoL(1) = DNMATL*THKSTF(1,iunit)/SPACNG
      a1L(1)  = 0.
      a1L(2)  = 0.
      e2L(1)  = e1L(1)
      e2L(2)  = e1L(2)
      a2L(1)  = 0.
      a2L(2)  = 0.
c
      WRITE(6,'(/,A,1P,3E12.4)')' TATX,HATX,ecz=',TATX,HATX,ecz
      WRITE(6,'(/,A,/,1P,6E12.4)')
     1' e1L(1),e1L(2),u12L(1),u12L(2),gL(1),gL(2)=',
     1  e1L(1),e1L(2),u12L(1),u12L(2),gL(1),gL(2)
      WRITE(6,'(/,A,/,1P,6E12.4)')
     1' tL(1),tL(2),rhoL(1),rhoL(2),e2L(1),e2L(2)=',
     1  tL(1),tL(2),rhoL(1),rhoL(2),e2L(1),e2L(2)
c
      return
c
c     write (not,900)
c     stop

c 900 format (//' SUBROUTINE WALL HAS NOT BEEN PROVIDED.')

      end
=================================================================
```