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

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

C=DECK BEHAVIOR.NEW

C This library contains the skeletons of

C subroutines called SUBROUTINE BEHXn, n = 1,

C 2, 3, . . . that will yield predictions

C of behavioral responses of various systems

C to environments (loads).

C

C You may complete the subroutines by writing

C algorithms that yield the responses,

C each of which plays a part in constraining

C the design to a feasible region. Examples

C of responses are: stress, buckling, drag,

C vibration, deformation, clearances, etc.

C

C A skeleton routine called SUBROUTINE OBJECT

C is also provided for any objective function

C (e.g. weight, deformation, conductivity)

C you may wish to create.

C

C A skeleton routine called SUBROUTINE USRCON

C is also provided for any user-written

C constraint condition you may wish to write:

C This is an INEQUALITY condition that

C involves any program variables. However,

C note that this kind of thing is done

C automatically in the program DECIDE, so

C try DECIDE first to see if your particular

C constraint conditions can be accommodated

C more easily there.

C

C Please note that you do not have to modify

C BEHAVIOR.NEW in any way, but may instead

C prefer to insert your subroutines into the

C skeletal libraries ADDCODEn.NEW, n=1,2,...

C and appropriate common blocks, dimension

C and type statements and calls to these

C subroutines in the library STRUCT.NEW.

C This strategy is best if your FORTRAN

C input to GENOPT contains quite a bit

C of software previously written by

C yourself or others, and/or the generation

C of behavioral constraints is more easily

C accomplished via another architecture

C than that provided for in the

C BEHAVIOR.NEW library. (See instructions

C in the libraries ADDCODEn.NEW and

C STRUCT.NEW for this procedure.)

C

C The two test cases provided with GENOPT

C provide examples of each method:

C PLATE (test case 1): use of BEHAVIOR.NEW

C PANEL (test case 2): use of ADDCODEn.NEW

C and STRUCT.NEW.

C

C SEVEN ROLES THAT VARIABLES IN THIS SYSTEM OF PROGRAMS PLAY

C

C A variable can have one of the following roles:

C

C 1 = a possible decision variable for optimization,

C typically a dimension of a structure.

C 2 = a constant parameter (cannot vary as design evolves),

C typically a control integer or material property,

C but not a load, allowable, or factor of safety,

C which are asked for later.

C 3 = a parameter characterizing the environment, such

C as a load component or a temperature.

C 4 = a quantity that describes the response of the

C structure, (e.g. stress, buckling load, frequency)

C 5 = an allowable, such as maximum allowable stress,

C minimum allowable frequency, etc.

C 6 = a factor of safety

C 7 = the quantity that is to be minimized or maximized,

C called the "objective function" (e.g. weight).

C ===========================================================

C

C NAMES, DEFINITIONS, AND ROLES OF THE VARIABLES:

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

C OPTIMIZATION PROGRAM FOR A PARTICULAR CLASS OF PROBLEMS.

C THE NAME YOU HAVE CHOSEN FOR THIS CLASS OF PROBLEMS IS: balloon

C "GENOPT" (GENeral OPTimization) was written during 1987-1988

C by Dr. David Bushnell, Dept. 93-30, Bldg. 251, (415)424-3237

C Lockheed Missiles and Space Co., 3251 Hanover St.,

C Palo Alto, California, USA 94304

C The optimizer used in GENOPT is called ADS, and was

C written by G. Vanderplaats [3]. It is based on the method

C of feasible directions [4].

C ABSTRACT

C "GENOPT" has the following purposes and properties:

C 1. Any relatively simple analysis is "automatically"

C converted into an optimization of whatever system

C can be analyzed with fixed properties. Please note

C that GENOPT is not intended to be used for problems

C that require elaborate data-base management systems

C or large numbers of degrees of freedom.

C 2. The optimization problems need not be in fields nor

C jargon familiar to me, the developer of GENOPT.

C Although all of the example cases (See the cases

C in the directories under genopt/case)

C are in the field of structural analysis, GENOPT is

C not limited to that field.

C 3. GENOPT is a program that writes other programs. These

C programs, WHEN AUGMENTED BY USER-SUPPLIED CODING,

C form a program system that should be user-friendly in

C the GENOPT-user"s field. In this instance the user

C of GENOPT must later supply FORTRAN coding that

C calculates behavior in the problem class called "balloon".

C 4. Input data and textual material are elicited from

C the user of GENOPT in a general enough way so that

C he or she may employ whatever data, definitions, and

C "help" paragraphs will make subsequent use of the

C program system thus generated easy by those less

C familiar with the class of problems "balloon" than

C the GENOPT user.

C 5. The program system generated by GENOPT has the same

C general architecture as previous programs written for

C specific applications by the developer [7 - 16]. That

C is, the command set is:

C BEGIN (User supplies starting design, loads,

C control integers, material properties,

C etc. in an interactive-help mode.)

C DECIDE (User chooses decision and linked

C variables and inequality constraints

C that are not based on behavior.)

C MAINSETUP (User chooses output option, whether

C to perform analysis of a fixed design

C or to optimize, and number of design

C iterations.)

C OPTIMIZE (The program system performs, in a batch

C mode, the work specified in MAINSETUP.)

C SUPEROPT (Program tries to find the GLOBAL optimum

C design as described in Ref.[11] listed

C below (Many OPTIMIZEs in one run.)

C CHANGE (User changes certain parameters)

C CHOOSEPLOT (User selects which quantities to plot

C vs. design iterations.)

C DIPLOT (User generates plots)

C CLEANSPEC (User cleans out unwanted files.)

C A typical runstream is:

C GENOPTLOG (activate command set)

C BEGIN (provide starting design, loads, etc.)

C DECIDE (choose decision variables and bounds)

C MAINSETUP (choose print option and analysis type)

C OPTIMIZE (launch batch run for n design iterations)

C OPTIMIZE (launch batch run for n design iterations)

C OPTIMIZE (launch batch run for n design iterations)

C OPTIMIZE (launch batch run for n design iterations)

C OPTIMIZE (launch batch run for n design iterations)

C CHANGE (change some variables for new starting pt)

C OPTIMIZE (launch batch run for n design iterations)

C OPTIMIZE (launch batch run for n design iterations)

C OPTIMIZE (launch batch run for n design iterations)

C OPTIMIZE (launch batch run for n design iterations)

C OPTIMIZE (launch batch run for n design iterations)

C CHOOSEPLOT (choose which variables to plot)

C DIPLOT (plot variables v. iterations)

C CHOOSEPLOT (choose additional variables to plot)

C DIPLOT (plot more variables v design iterations)

C CLEANSPEC (delete extraneous files for specific case)

C IMPORTANT: YOU MUST ALWAYS GIVE THE COMMAND "OPTIMIZE"

C SEVERAL TIMES IN SUCCESSION IN ORDER TO OBTAIN

C CONVERGENCE! AN EXPLANATION OF WHY YOU MUST DO

C THIS IS GIVEN ON P 580-582 OF THE PAPER "PANDA2,

C PROGRAM FOR MINIMUM WEIGHT DESIGN OF STIFFENED,

C COMPOSITE LOCALLY BUCKLED PANELS", Computers and

C Structures, Vol. 25, No. 4, pp 469-605 (1987).

C Due to introduction of a "global" optimizer, SUPEROPT,

C described in Ref.[11], you can now use the runstream

C BEGIN (provide starting design, loads, etc.)

C DECIDE (choose decision variables and bounds)

C MAINSETUP (choose print option and analysis type)

C SUPEROPT (launch batch run for "global" optimization)

C CHOOSEPLOT (choose which variables to plot)

C DIPLOT (plot variables v. iterations)

C "Global" is in quotes because SUPEROPT does its best to find

C a true global optimum design. The user is strongly urged to

C execute SUPEROPT/CHOOSEPLOT several times in succession in

C order to determine an optimum that is essentially just as

C good as the theoretical true global optimum. Each execution

C of the series,

C SUPEROPT

C CHOOSEPLOT

C does the following:

C 1. SUPEROPT executes many sets of the two processors,

C OPTIMIZE and AUTOCHANGE (AUTOCHANGE gets a new random

C "starting" design), in which each set does the following:

C OPTIMIZE (perform k design iterations)

C OPTIMIZE (perform k design iterations)

C OPTIMIZE (perform k design iterations)

C OPTIMIZE (perform k design iterations)

C OPTIMIZE (perform k design iterations)

C AUTOCHANGE (get new starting design randomly)

C SUPEROPT keeps repeating the above sequence until the

C total number of design iterations reaches about 270.

C The number of OPTIMIZEs per AUTOCHANGE is user-provided.

C 2. CHOOSEPLOT allows the user to plot stuff and resets the

C total number of design iterations from SUPEROPT to zero.

C After each execution of SUPEROPT the user MUST execute

C CHOOSEPLOT: before the next execution of SUPEROPT the

C total number of design iterations MUST be reset to zero.

C REFERENCES

C [1] Bushnell, D., "GENOPT--A program that writes

C user-friendly optimization code", International

C Journal of Solids and Structures, Vol. 26, No. 9/10,

C pp. 1173-1210, 1990. The same paper is contained in a

C bound volume of papers from the International Journal of

C Solids and Structures published in memory of Professor

C Charles D. Babcock, formerly with the California Institute

C of Technology.

C [2] Bushnell, D., "Automated optimum design of shells of

C revolution with application to ring-stiffened cylindrical

C shells with wavy walls", AIAA paper 2000-1663, 41st

C AIAA Structures Meeting, Atlanta, GA, April 2000. Also see

C Lockheed Martin report, same title, LMMS P525674, November

C 1999

C [2b] Bushnell, D., "Minimum weight design of imperfect

C isogrid-stiffened ellipsoidal shells under uniform external

C pressure", AIAA paper 2009-2702, 50th AIAA Structures

C Meeting, Palm Springs, CA, May 4-7, 2009

C [2c] Bushnell, D., "Use of GENOPT and a BIGBOSOR4 "huge"

C torus" model to optimize a typical weld land and weld land

C edge stringers in a previously optimized internally

C stiffened cylindrical shell without weld lands",

C unpublished report to NASA Langley Research, May 15, 2009

C [2d] Bushnell, D., "Use of GENOPT and BIGBOSOR4 to obtain

C optimum designs of a cylindrical shell with a composite

C truss-core sandwich wall", unpublished report to NASA

C Langley Research Center, Hampton, VA, June 20, 2009

C [2e] Bushnell, D., "Use of GENOPT and BIGBOSOR4 to obtain

C an optimum design of a deep submergence tank", unpublished

C report to the DOER company, Alameda, CA, June 30, 2009

C [3] Vanderplaats, G. N., "ADS--a FORTRAN program for

C automated design synthesis, Version 2.01", Engineering

C Design Optimization, Inc, Santa Barbara, CA, January, 1987

C [4] Vanderplaats, G. N. and Sugimoto, H., "A general-purpose

C optimization program for engineering design", Computers

C and Structures, Vol. 24, pp 13-21, 1986

C [5] Bushnell, D., "BOSOR4: Program for stress, stability,

C and vibration of complex, branched shells of revolution",

C in STRUCTURAL ANALYSIS SYSTEMS, Vol. 2, edited by A.

C Niku-Lari, pp. 25-54, (1986)

C [6] Bushnell, D., "BOSOR5: Program for buckling of complex,

C branched shells of revolution including large deflections,

C plasticity and creep," in STRUCTURAL ANALYSIS SYSTEMS, Vol.

C 2, edited by A. Niku-Lari, pp. 55-67, (1986)

C [7] Bushnell, D., "PANDA2--program for minimum weight

C design of stiffened, composite, locally buckled panels",

C COMPUTERS AND STRUCTURES, vol. 25, No. 4, pp 469-605, 1987

C [8] Bushnell, D., "Improved optimum design of dewar

C supports", COMPUTERS and STRUCTURES, Vol. 29, No. 1,

C pp. 1-56 (1988)

C [9] Bushnell, D., "SPHERE - Program for minimum weight

C design of isogrid-stiffened spherical shells under uniform

C external pressure", Lockheed Report F372046, January, 1990

C [10] Bushnell, D.,"Optimum design of imperf.isogrid-stiffened

C ellipsoidal shells...", written and placed in the file

C ..genopt/case/torisph/sdm50.report.pdf

C [11] Bushnell, D., "Recent enhancements to PANDA2", AIAA

C paper 96-1337-CP, Proc. 37th AIAA SDM Meeting, April 1996

C pp. 126-182, in particular, pp. 127-130

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

C [13] Bushnell, D., the case ..genopt/case/torisph, Ref.[2b]

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

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

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

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

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

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

C [20] Bushnell, D., the case ..genopt/case/sphere

C==============================================================

C TABLE 1 "GENOPT" COMMANDS

C==============================================================

C HELPG (get information on GENOPT.)

C GENTEXT (GENOPT user generate a prompt file, program

C fragments [see TABLE 5], programs [see

C TABLE 4]., and this and other files

C [see TABLE 5 and the rest of this file.])

C GENPROGRAMS (GENOPT user generate absolute elements:

C BEGIN.EXE, DECIDE.EXE, MAINSETUP.EXE,

C OPTIMIZE.EXE, CHANGE.EXE, STORE.EXE,

C CHOOSEPLOT.EXE, DIPLOT.EXE.)

C BEGIN (end user provide starting data.)

C DECIDE (end user choose decision variables, bounds,

C linked variables,inequality constraints.)

C MAINSETUP (end user set up strategy parameters.)

C OPTIMIZE (end user perform optimization, batch mode.)

C SUPEROPT (Program tries to find the GLOBAL optimum

C design as described in Ref.[11] listed

C above (Many OPTIMIZEs in one run.)

C CHANGE (end user change some parameters.)

C CHOOSEPLOT (end user choose which variables to plot v.

C design iterations.)

C DIPLOT (end user obtain plots.)

C INSERT (GENOPT user add parameters to the problem.)

C CLEANGEN (GENOPT user cleanup your GENeric files.)

C CLEANSPEC (end user cleanup your SPECific case files)

C Please consult the following sources for more

C information about GENOPT:

C 1. GENOPT.STORY and HOWTO.RUN and GENOPT.NEWS

C 2. Sample cases: (in the directory, genopt/case)

C 3. NAME.DEF file, where NAME is the name chosen by

C the GENOPT-user for a class of problems. (In this

C case NAME = balloon)

C 4. GENOPT.HLP file (type HELPG)

C=============================================================

C==============================================================

C TABLE 2 GLOSSARY OF VARIABLES USED IN "balloon"

C==============================================================

C ARRAY NUMBER OF PROMPT

C ? (ROWS,COLS) ROLE NUMBER NAME DEFINITION OF VARIABLE

C (balloon.PRO)

C==============================================================

C n ( 0, 0) 2 10 LENGTH = length of the cylindrical shell

C n ( 0, 0) 2 15 RADIUS = inner radius of the cylindrical b

C n ( 0, 0) 2 20 NMODUL = number of modules over 90 degrees

C n ( 0, 0) 2 30 IEMOD1 = material number in EMOD1(IEMOD1)

C y ( 10, 0) 2 35 EMOD1 = elastic modulus, meridional direc

C y ( 10, 0) 2 40 EMOD2 = elastic modulus, circumferential

C y ( 10, 0) 2 45 G12 = in-plane shear modulus

C y ( 10, 0) 2 50 G13 = out-of-plane (s,z) shear modulus

C y ( 10, 0) 2 55 G23 = out-of-plane (y,z) shear modulus

C y ( 10, 0) 2 60 NU = Poisson ratio

C y ( 10, 0) 2 65 ALPHA1 = meridional coef. thermal expansio

C y ( 10, 0) 2 70 ALPHA2 = circumf.coef.thermal expansion

C y ( 10, 0) 2 75 TEMPER = delta-T from fabrication temperat

C y ( 10, 0) 2 80 DENSTY = weight density of material

C n ( 0, 0) 1 90 HEIGHT = height from inner to outer membra

C n ( 0, 0) 1 95 RINNER = radius of curvature of inner memb

C n ( 0, 0) 1 100 ROUTER = radius of curvature of outer memb

C n ( 0, 0) 1 105 TINNER = thickness of the inner curved mem

C n ( 0, 0) 1 110 TOUTER = thickness of the outer curved mem

C n ( 0, 0) 1 115 TFINNR = thickness of inner truss-core seg

C n ( 0, 0) 1 120 TFOUTR = thickness of the outer truss segm

C n ( 0, 0) 1 125 TFWEBS = thickness of each truss-core web

C n ( 0, 0) 2 135 NCASES = Number of load cases (number of e

C y ( 20, 0) 3 140 PINNER = pressure inside the inner membran

C y ( 20, 0) 3 145 PMIDDL = pressure between inner and outer

C y ( 20, 0) 3 150 POUTER = pressure outside the outer membra

C y ( 20, 0) 4 160 GENBUK = general buckling load factor

C y ( 20, 0) 5 165 GENBUKA = allowable for general buckling lo

C y ( 20, 0) 6 170 GENBUKF = general buckling factor of safety

C n ( 0, 0) 2 175 JSTRM1 = stress component number in STRM1(

C y ( 20, 5) 4 180 STRM1 = stress component in material 1

C y ( 20, 5) 5 185 STRM1A = allowable stress in material 1

C y ( 20, 5) 6 190 STRM1F = factor of safety for stress in ma

C y ( 20, 5) 4 195 STRM2 = stress component in material 2

C y ( 20, 5) 5 200 STRM2A = allowable for stress in material

C y ( 20, 5) 6 205 STRM2F = factor of safety for stress in ma

C y ( 20, 5) 4 210 STRM3 = stress component in material 3

C y ( 20, 5) 5 215 STRM3A = allowable for stress in material

C y ( 20, 5) 6 220 STRM3F = factor of safety for stress in ma

C n ( 0, 0) 7 230 WEIGHT = weight/length of the balloon

C

C=DECK BEHX1

SUBROUTINE BEHX1

1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,PHRASE)

C

C PURPOSE: OBTAIN general buckling load factor

C

C YOU MUST WRITE CODE THAT, USING

C THE VARIABLES IN THE LABELLED

C COMMON BLOCKS AS INPUT, ULTIMATELY

C YIELDS THE RESPONSE VARIABLE FOR

C THE ith LOAD CASE, ILOADX:

C

C GENBUK(ILOADX)

C

C AS OUTPUT. THE ith CASE REFERS

C TO ith ENVIRONMENT (e.g. load com-

C bination).

C

C DEFINITIONS OF INPUT DATA:

C IMODX = DESIGN CONTROL INTEGER:

C IMODX = 0 MEANS BASELINE DESIGN

C IMODX = 1 MEANS PERTURBED DESIGN

C IFAST = 0 MEANS FEW SHORTCUTS FOR PERTURBED DESIGNS

C IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS

C IFILE = FILE FOR OUTPUT LIST:

C NPRINX= OUTPUT CONTROL INTEGER:

C NPRINX=0 MEANS SMALLEST AMOUNT

C NPRINX=1 MEANS MEDIUM AMOUNT

C NPRINX=2 MEANS LOTS OF OUTPUT

C

C ILOADX = ith LOADING COMBINATION

C PHRASE = general buckling load factor

C

C OUTPUT:

C

C GENBUK(ILOADX)

C

CHARACTER\*80 PHRASE

C INSERT ADDITIONAL COMMON BLOCKS:

COMMON/FV03/EMOD1(10),IEMOD1

REAL EMOD1

COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)

REAL EMOD2,G12,G13,G23,NU,ALPHA1

COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER

REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER

COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)

REAL ALPHA2,TEMPER,DENSTY

COMMON/FV21/PINNER(20)

REAL PINNER

COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)

REAL GENBUK,GENBUKA,GENBUKF

COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )

REAL STRM1,STRM1A,STRM1F

COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )

REAL STRM2,STRM2A,STRM2F

COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )

REAL STRM3,STRM3A,STRM3F

COMMON/IV01/NMODUL

INTEGER NMODUL

COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT

REAL TFINNR,TFOUTR,TFWEBS,WEIGHT

COMMON/FV22/PMIDDL(20),POUTER(20)

REAL PMIDDL,POUTER

C

C INSERT SUBROUTINE STATEMENTS HERE.

**C**

**COMMON/ITRYX/ITRY**

**DOUBLE PRECISION FTOTX**

**COMMON/FPREBX/FMAXST(200),FTOTX(20000)**

**COMMON/IFPREB/IFTOTS**

**COMMON/SEGS/NSEGB4,M2B4,I5B4(295),I2B4,I2GB4**

**COMMON/IFRHX/IFBB4,RHFIX(198),**

**1 KKKK,MNUMB,ISWTCH,KNTB4,IFTOT,INDSIG,IFIXB4**

**COMMON/FLNFLO/FLINNR,FLOUTR**

**COMMON/WRDCLX/WRDCOL**

**CHARACTER\*45 WRDCOL**

**COMMON/ITERS/ITER**

**COMMON/ITERS2/ITRSTP(200)**

**COMMON/MEMSTR/STRS1F(1,5),STRS2F(1,5),STRS1V(1,5),STRS2V(1,5)**

**DIMENSION THK(5)**

**COMMON/ERROR/ERR**

**COMMON/N1N2FX/N1FIX(100,100),N2FIX(100,100)**

**COMMON/N1N2VR/N1VAR(100,100),N2VAR(100,100)**

**REAL N1FIX,N2FIX,N1VAR,N2VAR**

**COMMON/N2DIFX/N2DIFF(6)**

**REAL N2DIFF**

**COMMON/FINNER/C44FIN,DELTAT,DELT,NODSEG,MSEGS**

**COMMON/NUMSEG/NSEGS**

**COMMON/INSTAB/INDIC**

**COMMON/EIGB4M/EIGCOM(200),EIGNEG(200),EIGCRN**

**COMMON/WVEB4M/NWVCOM(200),NWVNEG(200),IWAVEB,NWVCRN**

**COMMON/EIGBUK/EIGCRT**

**COMMON/NWVBUK/NWVCRT**

**COMMON/BUCKN/N0BX,NMINBX,NMAXBX,INCRBX**

**COMMON/BUCKN0/N0B,NMAXB**

**COMMON/RBEGX/RBIG0,RBIGL,RBIGG**

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

**COMMON/EIGALL/EIG0,EIG1,EIG2,EIG3,EIG4**

**COMMON/WAVALL/NWAV0,NWAV1,NWAV2,NWAV3,NWAV4**

**COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX**

**common/caseblock/CASE**

**CHARACTER\*28 CASE**

**CHARACTER\*35 CASA,CASA2**

**C**

**PI = 3.1415927**

**C**

**IF (NMODUL.GT.36) THEN**

**I=INDEX(CASE,' ')**

**NLET = I - 1**

**IF (I.EQ.0) NLET = 28**

**WRITE(IFILE,'(/,A,/,A,/,A,A,A,/,A)')**

**1' \*\*\*\*\*\*\*\*\*\*\*\*\* RUN ABORT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*',**

**1' Too many modules. NMODUL must be less than 36',**

**1' Reduce NMODUL in the file, ',CASE(1:NLET),'.BEG.',**

**1' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

**CALL ERREX**

**ENDIF**

**C**

**IF (IMODX.EQ.0) ERR = 0.**

**IF (IMODX.EQ.1) ERR = 0.01**

**C**

**RAVE = RADIUS/PI**

**RBIGG = RAVE**

**C**

**C Obtain nonlinear equilibrium for Load Set B by itself.**

**C Use 10 load steps to assure convergence.**

**C**

**INDIC = 0**

**WRDCOL = ' '**

**IFTOTS = 0**

**ITRY = 1**

**NSTEPS = 10**

**CALL MOVER(0.,0,FTOTX,1,40000)**

**CALL BOSDEC(0,24,ILOADX,INDIC)**

**C**

**IF (0.5\*FLINNR.GT.RINNER) THEN**

**WRITE(IFILE,'(/,A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\* ABORT \*\*\*\*\*\*\*\*\*\*'**

**WRITE(IFILE,'(A)') ' 0.5 x FLINNR is greater than RINNER'**

**WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')**

**1 ' 0.5 x FLINNR =',0.5\*FLINNR,'; RINNER =',RINNER,'; IMODX=',IMODX**

**WRITE(IFILE,'(A)') ' Put a higher lower bound on RINNER.'**

**WRITE(IFILE,'(A)') ' The run is now aborting.'**

**WRITE(IFILE,'(A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

**CALL ERREX**

**ENDIF**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**C**

**IF (0.5\*FLOUTR.GT.ROUTER) THEN**

**WRITE(IFILE,'(/,A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\* ABORT \*\*\*\*\*\*\*\*\*\*'**

**WRITE(IFILE,'(A)') ' 0.5 x FLOUTR is greater than ROUTER'**

**WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')**

**1 ' 0.5 x FLOUTR =',0.5\*FLOUTR,'; ROUTER =',ROUTER,'; IMODX=',IMODX**

**WRITE(IFILE,'(A)') ' Put a higher lower bound on ROUTER.'**

**WRITE(IFILE,'(A)') ' The run is now aborting.'**

**WRITE(IFILE,'(A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

**CALL ERREX**

**ENDIF**

**C**

**C IF (ITYPEX.EQ.2) THEN**

**C Get CASE.BEHX2 file for input for BIGBOSOR4...**

**C CASE.BEHX2 is an input file for BIGBOSOR4 for behavior no. 2:**

**C pre-buckling state of the balloon.**

**C I=INDEX(CASE,' ')**

**C IF(I.NE.0) THEN**

**C CASA=CASE(:I-1)//'.BEHX2'**

**C ELSE**

**C CASA=CASE//'.BEHX2'**

**C ENDIF**

**C OPEN(UNIT=61,FILE=CASA2,STATUS='UNKNOWN')**

**C CALL BOSDEC(1,61,ILOADX,INDIC)**

**C CLOSE(UNIT=61)**

**C WRITE(IFILE,'(/,/,A,A,/,A)')**

**C 1 ' BIGBOSOR4 input file for:',**

**C 1 ' pre-buckling state of the balloon',**

**C 1 CASA2**

**C ENDIF**

**C**

**CALL B4READ**

**CALL B4MAIN**

**C**

**ILETW = INDEX(WRDCOL,'SHELL COLLAPSES AXISYMMETRICALLY')**

**IF (ILETW.NE.0) THEN**

**WRITE(IFILE,'(/,A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\* ABORT \*\*\*\*\*\*\*\*\*\*'**

**WRITE(IFILE,'(A,A,/,1P,5E14.6)')**

**1 ' Decision variable candidates,',**

**1 ' HEIGHT,RINNER,ROUTER,TINNER,TOUTER=',**

**1 HEIGHT,RINNER,ROUTER,TINNER,TOUTER**

**WRITE(IFILE,'(A,1P,3E12.4)')**

**1 ' TFINNR,TFOUTR,TFWEBS=',TFINNR,TFOUTR,TFWEBS**

**WRITE(IFILE,'(A)') ' SHELL COLLAPSES AXISYMMETRICALLY'**

**WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')**

**1 ' 0.5 x FLINNR =',0.5\*FLINNR,'; RINNER =',RINNER,'; IMODX=',IMODX**

**WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')**

**1 ' 0.5 x FLOUTR =',0.5\*FLOUTR,'; ROUTER =',ROUTER,'; IMODX=',IMODX**

**WRITE(IFILE,'(A)') ' Put a higher lower bound on RINNER or'**

**WRITE(IFILE,'(A)') ' put a higher lower bound on ROUTER.'**

**WRITE(IFILE,'(A)') ' The run is now aborting.'**

**WRITE(IFILE,'(A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

**CALL ERREX**

**ENDIF**

**C**

**ILETW = INDEX(WRDCOL,'INITIAL LOADS TOO HIGH FOR THIS STRUCT')**

**IF (ILETW.NE.0) THEN**

**WRITE(IFILE,'(A,A,/,1P,5E14.6)')**

**1 ' Decision variable candidates,',**

**1 ' HEIGHT,RINNER,ROUTER,TINNER,TOUTER=',**

**1 HEIGHT,RINNER,ROUTER,TINNER,TOUTER**

**WRITE(IFILE,'(A,1P,3E12.4)')**

**1 ' TFINNR,TFOUTR,TFWEBS=',TFINNR,TFOUTR,TFWEBS**

**WRITE(IFILE,'(/,A)')**

**1 ' \*\*\*\*\*\* CHANGE FROM 10 TO 1 LOAD STEPS \*\*\*\*\*\*'**

**WRITE(IFILE,'(A)') ' INITIAL LOADS TOO HIGH FOR THIS STRUCT'**

**WRITE(IFILE,'(A,I2)') ' Changing from 10 to 1 steps: IMODX=',**

**1 IMODX**

**WRITE(IFILE,'(A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

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

**WRDCOL = ' '**

**INDIC = 0**

**IFTOTS = 0**

**ITRY = 2**

**NSTEPS = 1**

**CALL MOVER(0.,0,FTOTX,1,40000)**

**CALL BOSDEC(0,24,ILOADX,INDIC)**

**CALL B4READ**

**CALL B4MAIN**

**ITRY = 1**

**ENDIF**

**C**

**ILETW = INDEX(WRDCOL,'SHELL COLLAPSES AXISYMMETRICALLY')**

**IF (ILETW.NE.0) THEN**

**WRITE(IFILE,'(A,A,/,1P,5E14.6)')**

**1 ' Decision variable candidates,',**

**1 ' HEIGHT,RINNER,ROUTER,TINNER,TOUTER=',**

**1 HEIGHT,RINNER,ROUTER,TINNER,TOUTER**

**WRITE(IFILE,'(A,1P,3E12.4)')**

**1 ' TFINNR,TFOUTR,TFWEBS=',TFINNR,TFOUTR,TFWEBS**

**WRITE(IFILE,'(/,A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\* ABORT \*\*\*\*\*\*\*\*\*\*'**

**WRITE(IFILE,'(A)') ' SHELL COLLAPSES AXISYMMETRICALLY'**

**WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')**

**1 ' 0.5 x FLINNR =',0.5\*FLINNR,'; RINNER =',RINNER,'; IMODX=',IMODX**

**WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')**

**1 ' 0.5 x FLOUTR =',0.5\*FLOUTR,'; ROUTER =',ROUTER,'; IMODX=',IMODX**

**WRITE(IFILE,'(A)') ' Put a higher lower bound on RINNER or'**

**WRITE(IFILE,'(A)') ' put a higher lower bound on ROUTER.'**

**WRITE(IFILE,'(A)') ' The run is now aborting.'**

**WRITE(IFILE,'(A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

**CALL ERREX**

**ENDIF**

**C**

**ILETW = INDEX(WRDCOL,'INITIAL LOADS TOO HIGH FOR THIS STRUCT')**

**IF (ILETW.NE.0) THEN**

**WRITE(IFILE,'(A,A,/,1P,5E14.6)')**

**1 ' Decision variable candidates,',**

**1 ' HEIGHT,RINNER,ROUTER,TINNER,TOUTER=',**

**1 HEIGHT,RINNER,ROUTER,TINNER,TOUTER**

**WRITE(IFILE,'(A,1P,3E12.4)')**

**1 ' TFINNR,TFOUTR,TFWEBS=',TFINNR,TFOUTR,TFWEBS**

**WRITE(IFILE,'(/,A)')**

**1 ' \*\*\*\*\*\* CHANGE FROM 1 TO 50 LOAD STEPS \*\*\*\*\*\*'**

**WRITE(IFILE,'(A)') ' INITIAL LOADS TOO HIGH FOR THIS STRUCT'**

**WRITE(IFILE,'(A,I2)') ' Changing from 1 to 50 steps: IMODX=',**

**1 IMODX**

**WRITE(IFILE,'(A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

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

**WRDCOL = ' '**

**INDIC = 0**

**IFTOTS = 0**

**ITRY = 3**

**NSTEPS = 50**

**CALL MOVER(0.,0,FTOTX,1,40000)**

**CALL BOSDEC(0,24,ILOADX,INDIC)**

**CALL B4READ**

**CALL B4MAIN**

**ITRY = 1**

**ENDIF**

**C**

**ILETW = INDEX(WRDCOL,'SHELL COLLAPSES AXISYMMETRICALLY')**

**IF (ILETW.NE.0) THEN**

**WRITE(IFILE,'(A,A,/,1P,5E14.6)')**

**1 ' Decision variable candidates,',**

**1 ' HEIGHT,RINNER,ROUTER,TINNER,TOUTER=',**

**1 HEIGHT,RINNER,ROUTER,TINNER,TOUTER**

**WRITE(IFILE,'(A,1P,3E12.4)')**

**1 ' TFINNR,TFOUTR,TFWEBS=',TFINNR,TFOUTR,TFWEBS**

**WRITE(IFILE,'(/,A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\* ABORT \*\*\*\*\*\*\*\*\*\*'**

**WRITE(IFILE,'(A)') ' SHELL COLLAPSES AXISYMMETRICALLY'**

**WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')**

**1 ' 0.5 x FLINNR =',0.5\*FLINNR,'; RINNER =',RINNER,'; IMODX=',IMODX**

**WRITE(IFILE,'(A,1P,E12.4,A,1P,E12.4,A,I2)')**

**1 ' 0.5 x FLOUTR =',0.5\*FLOUTR,'; ROUTER =',ROUTER,'; IMODX=',IMODX**

**WRITE(IFILE,'(A)') ' Put a higher lower bound on RINNER or'**

**WRITE(IFILE,'(A)') ' put a higher lower bound on ROUTER.'**

**WRITE(IFILE,'(A)') ' The run is now aborting.'**

**WRITE(IFILE,'(A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

**CALL ERREX**

**ENDIF**

**C**

**ILETW = INDEX(WRDCOL,'INITIAL LOADS TOO HIGH FOR THIS STRUCT')**

**IF (ILETW.NE.0) THEN**

**WRITE(IFILE,'(A,A,/,1P,5E14.6)')**

**1 ' Decision variable candidates,',**

**1 ' HEIGHT,RINNER,ROUTER,TINNER,TOUTER=',**

**1 HEIGHT,RINNER,ROUTER,TINNER,TOUTER**

**WRITE(IFILE,'(A,1P,3E12.4)')**

**1 ' TFINNR,TFOUTR,TFWEBS=',TFINNR,TFOUTR,TFWEBS**

**WRITE(IFILE,'(/,A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\* ABORT \*\*\*\*\*\*\*\*\*\*'**

**WRITE(IFILE,'(A)') ' INITIAL LOADS TOO HIGH FOR THIS STRUCT'**

**WRITE(IFILE,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')**

**1 ' This is an unrecoverable error because we have already',**

**1 ' tried and failed to obtain nonlinear pre-buckling convergence',**

**1 ' by changing from a nonlinear solution with 10 load steps to',**

**1 ' a nonlinear solution with 1 load step and then changing from',**

**1 ' 1 load step to 50 load steps:three tries. That strategy just',**

**1 ' failed. You may well have performed enough design iterations',**

**1 ' to have a good optimum design now. Look near the end of the',**

**1 ' \*.OPP file at the "FEASIBLE" and "ALMOST FEASIBLE" designs.'**

**WRITE(IFILE,'(A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')**

**1 ' If you are not satisfied that you have performed enough',**

**1 ' design iterations, then look at the thicknesses of the',**

**1 ' various segments. If any thicknesses seem too small, then',**

**1 ' increase them and also increase the corresponding lower',**

**1 ' bounds of them. Another thing you can try that has worked',**

**1 ' for Bushnell is to look near the end of the \*.OPM file for',**

**1 ' the last successfully obtained design. Use the GENOPT',**

**1 ' processor, CHANGE, to reset the values of the decision',**

**1 ' variables to those of the last successfully obtained design',**

**1 ' and then launch a new execution of SUPEROPT, probably',**

**1 ' leaving the lower bounds unchanged, or perhaps also changing',**

**1 ' them if you wish (before launching SUPEROPT, of course).'**

**WRITE(IFILE,'(A,I2)') ' The run is now aborting: IMODX=',IMODX**

**WRITE(IFILE,'(A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

**CALL ERREX**

**ENDIF**

**C**

**IF (IMODX.EQ.0) THEN**

**WRITE(IFILE,'(/,A,/,A,/,A,1P,E12.4,A,1P,E12.4,A,1P,E12.4,/,A)')**

**1 ' Newton iterations required to solve the nonlinear',**

**1 ' axisymmetric pre-buckling equilibrium state for the',**

**1 ' "fixed" loads, PINNER=',PINNER(ILOADX),**

**1 ', PMIDDL=',PMIDDL(ILOADX),', DELTAT=',DELTAT,**

**1 ' LOAD STEP Newton iterations Maximum displacement'**

**DO 3 I = 1,NSTEPS**

**WRITE(IFILE,'(I6,I15,1P,E25.6)') I,ITRSTP(I),FMAXST(I)**

**3 CONTINUE**

**ENDIF**

**C**

**WRITE(IFILE,'(A,A,/,A,/,A,I2)') ' WRDCOL=',WRDCOL,**

**1' IMODX=0 for current design,',**

**1' IMODX=1 for perturbed design: IMODX=',IMODX**

**C**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**C WRITE(IFILE,'(/,A,2I10)')' IFTOT, M2B4 =', IFTOT,M2B4**

**IFTOTS = IFTOT**

**M22B4 = 2\*M2B4**

**CALL GASP(FTOTX,M22B4,3,IFTOT)**

**C WRITE(IFILE,'(/,A,/,(1P,10E10.2))')**

**C 1' Nonlinear solution for Load Set B by itself, FTOTX=',**

**C 1 (FTOTX(I),I=1,M2B4)**

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

**C**

**INDIC = 1**

**N0B = 1**

**NMAXB = 1**

**WRDCOL = ' '**

**CALL BOSDEC(1,24,ILOADX,INDIC)**

**C**

**IF (ITYPEX.EQ.2) THEN**

**C Get CASE.BEHX1 file for input for BIGBOSOR4...**

**C CASE.BEHX1 is an input file for BIGBOSOR4 for behavior no. 1:**

**C general buckling load**

**I=INDEX(CASE,' ')**

**IF(I.NE.0) THEN**

**CASA=CASE(:I-1)//'.BEHX1'**

**ELSE**

**CASA=CASE//'.BEHX1'**

**ENDIF**

**OPEN(UNIT=61,FILE=CASA,STATUS='UNKNOWN')**

**CALL BOSDEC(1,61,ILOADX,INDIC)**

**CLOSE(UNIT=61)**

**WRITE(IFILE,'(/,/,A,A,/,A)')**

**1 ' BIGBOSOR4 input file for:',**

**1 ' general buckling load',**

**1 CASA**

**ENDIF**

**C**

**CALL B4READ**

**IF (IMODX.EQ.0) THEN**

**N0BX = N0B**

**NMINBX = N0B**

**NMAXBX = NMAXB**

**INCRBX = 1**

**ELSE**

**NOBX = NWAV1**

**NMINBX = NWAV1**

**NMAXBX = NWAV1**

**INCRBX = 1**

**ENDIF**

**REWIND IFILE9**

**CALL STOCM1(IFILE9)**

**CALL STOCM2(IFILE9)**

**CALL B4MAIN**

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

**IF (IMODX.EQ.0) THEN**

**EIG2 = EIGCRT**

**NWAV1= NWVCRT**

**ENDIF**

**C**

**ILETW = INDEX(WRDCOL,'SHELL COLLAPSES AXISYMMETRICALLY')**

**IF (ILETW.NE.0) THEN**

**WRITE(IFILE,'(/,A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\* ABORT \*\*\*\*\*\*\*\*\*\*'**

**WRITE(IFILE,'(A)') ' SHELL COLLAPSES AXISYMMETRICALLY'**

**WRITE(IFILE,'(A,I2)') ' The run is now aborting: IMODX=',IMODX**

**WRITE(IFILE,'(A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

**CALL ERREX**

**ENDIF**

**C**

**ILETW = INDEX(WRDCOL,'INITIAL LOADS TOO HIGH FOR THIS STRUCT')**

**IF (ILETW.NE.0) THEN**

**WRITE(IFILE,'(/,A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\* ABORT \*\*\*\*\*\*\*\*\*\*'**

**WRITE(IFILE,'(A)') ' INITIAL LOADS TOO HIGH FOR THIS STRUCT'**

**WRITE(IFILE,'(A,I2)') ' The run is now aborting: IMODX=',IMODX**

**WRITE(IFILE,'(A)') ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

**CALL ERREX**

**ENDIF**

**C**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**C**

**C Compute the membrane stresses**

**THK(1) = TFOUTR**

**THK(2) = TOUTER**

**THK(3) = TFINNR**

**THK(4) = TINNER**

**THK(5) = TFWEBS**

**C**

**DO 5 J = 1,5**

**STRS1F(1,J) = N1FIX(1,J)/THK(J)**

**STRS2F(1,J) = N2FIX(1,J)/THK(J)**

**STRS1V(1,J) = N1VAR(1,J)/THK(J)**

**STRS2V(1,J) = N2VAR(1,J)/THK(J)**

**5 CONTINUE**

**C**

**IF (IMODX.EQ.0) THEN**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**WRITE(IFILE,'(/,A,/,A,/,A,/,A,/,A,1P,E12.4,A,/,A,/,A,1P,E12.4)')**

**1' Changes in temperature required to create 2 total axial loads:',**

**1' ',**

**1' 1. Change in temperature required to create the axial thermal',**

**1' strain that generates the axial tension due to closing the',**

**1' two ends of the pressurized volume (PMIDDL=',**

**1 PMIDDL(ILOADX),')',**

**1' between the inner and outer walls of the balloon in',**

**1' Load Step No. 1: DELTAT=',**

**1 DELTAT**

**WRITE(IFILE,'(/,A,/,A,/,A,1P,1E12.4,A,1P,E12.4,/,/)')**

**1' 2. Change in temperature required to simulate the Poisson',**

**1' axial expansion caused by the application of the outer',**

**1' pressure, POUTER =',**

**1 POUTER(ILOADX),' in Load Step No. 2: DELT=',**

**1 DELT**

**WRITE(IFILE,'(/,A)')**

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

**DO 10 I = 1,IWAVEB**

**WRITE(IFILE,'(A,1P,E12.4,A,I6,A)')**

**1 ' ',EIGCOM(I),'(',NWVCOM(I),')'**

**10 CONTINUE**

**WRITE(IFILE,'(A,1P,E12.4)')**

**1' Critical buckling load factor, GENBUK=',EIGCRT**

**WRITE(IFILE,'(A,I5)')**

**1' Critical number of axial half-waves, NWVCRT=',NWVCRT**

**C**

**DO 30 J = 1,6**

**DO 20 I = 1,1**

**N2DIFF(J) = N2VAR(I,J) - N2FIX(I,J)**

**20 CONTINUE**

**30 CONTINUE**

**WRITE(IFILE,'(/,A,/,A,/,A,/,A,1P,6E12.4,/,A,/,A,/,A,/,A,/,A)')**

**1' Differences in the resultants along the axis of the prismatic',**

**1' balloon for each segment, J, of the first module:',**

**1' [N2VAR(J) for the total load] - [N2FIX(J) for the fixed load]=',**

**1' N2DIFF(J),J=1,6)=',(N2DIFF(J),J=1,6),**

**1' N2VAR(J) (total load) are the resultants from Load Step No. 2.',**

**1' N2FIX(J) (fixed load) are the resultants from Load Step No. 1.',**

**1' NOTE: The stresses used as behavioral constraints are',**

**1' computed from N2VAR(J)/thickness(J). These stresses are',**

**1' lower than those computed from N2FIX(J)/thickness(J).'**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**C**

**IF (NPRINX.GE.2) THEN**

**WRITE(IFILE,'(/,A,/,A,A,/,A,A)')**

**1 ' PREBUCKLING STRESS RESULTANTS IN THE FIRST MODULE',**

**1 ' "fixed" from Load Step No. 1',**

**1 ' total from Load Step No. 2',**

**1 ' Seg.J Node I N1FIX(I,J) N2FIX(I,J)',**

**1 ' N1VAR(I,J) N2VAR(I,J)'**

**DO 60 J = 1,6**

**DO 50 I = 1,1**

**WRITE(IFILE,'(I5,I7,1P,4E13.4)')**

**1 J,I,N1FIX(I,J),N2FIX(I,J),N1VAR(I,J),N2VAR(I,J)**

**50 CONTINUE**

**60 CONTINUE**

**C**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**WRITE(IFILE,'(/,A,A,/,A,A,/,A,A,/,A,A)')**

**1 ' PREBUCKLING MEMBRANE STRESSES IN THE FIRST MODULE COMPUTED',**

**1 ' FROM',**

**1 ' N1FIX/thickness, N2FIX/thickness, N1VAR/thickness,',**

**1 ' N2VAR/thickness:',**

**1 ' "fixed" from Load Step No. 1',**

**1 ' total from Load Step No. 2',**

**1 ' Seg.J Node I STRS1F(I,J) STRS2F(I,J)',**

**1 ' STRS1V(I,J) STRS2V(I,J)'**

**DO 90 J = 1,5**

**DO 80 I = 1,1**

**WRITE(IFILE,'(I5,I7,1P,4E13.4)')**

**1 J,I,STRS1F(I,J),STRS2F(I,J),STRS1V(I,J),STRS2V(I,J)**

**80 CONTINUE**

**90 CONTINUE**

**C**

**WRITE(IFILE,'(/,A)')**

**1 ' Behavior number, General buckling load factor:'**

**ENDIF**

**C**

**WRITE(IFILE,**

**1 '(/,A,/,A,/,A,I3,/,A,1P,E12.4,/,A,/,A,/,A,I3,/,A,1P,E12.4,/)')**

**1' Newton iterations required to solve the nonlinear',**

**1' axisymmetric pre-buckling equilibrium state for the',**

**1' "fixed" loads (PINNER, PMIDDL, DELTAT): ITER=',ITRSTP(1),**

**1' Maximum displacement, FMAX=',FMAXST(1),**

**1' Newton iterations required to solve the nonlinear',**

**1' axisymmetric pre-buckling equilibrium state for the',**

**1' total loads (PINNER, PMIDDL, DELTAT, POUTER): ITER=',ITRSTP(2),**

**1' Maximum displacement, FMAX=',FMAXST(2)**

**C**

**ENDIF**

**C**

**IF (EIGCRT.LE.0.0) THEN**

**WRITE(IFILE,'(/,A,/,A,/,A,/,A,/,A,/,A,/,A,/,A)')**

**1 ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* RUN ABORT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*',**

**1 ' Divergence or failure of convergence of nonlinear',**

**1 ' pre-buckling solution either at Load Step No. 1 (fixed',**

**1 ' loads: PINNER, PMIDDL, DELTAT) or at Load Step No. 2',**

**1 ' (total loads: PINNER, PMIDDL, DELTAT, POUTER)',**

**1 ' Probably you should increase either RINNER or ROUTER or',**

**1 ' both RINNER and ROUTER.',**

**1 ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* RUN ABORT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

**CALL ERREX**

**ENDIF**

**C**

**GENBUK(ILOADX) = EIGCRT**

**C**

RETURN

END

C

C

C

C=DECK BEHX2

SUBROUTINE BEHX2

1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)

C

C PURPOSE: OBTAIN stress component in material 1

C

C YOU MUST WRITE CODE THAT, USING

C THE VARIABLES IN THE LABELLED

C COMMON BLOCKS AS INPUT, ULTIMATELY

C YIELDS THE RESPONSE VARIABLE FOR

C THE ith LOAD CASE, ILOADX:

C

C STRM1(ILOADX,JCOL)

C

C AS OUTPUT. THE ith CASE REFERS

C TO ith ENVIRONMENT (e.g. load com-

C bination).

C THE jth COLUMN (JCOL)

C INDEX IS DEFINED AS FOLLOWS:

C stress component number

C

C DEFINITIONS OF INPUT DATA:

C IMODX = DESIGN CONTROL INTEGER:

C IMODX = 0 MEANS BASELINE DESIGN

C IMODX = 1 MEANS PERTURBED DESIGN

C IFAST = 0 MEANS FEW SHORTCUTS FOR PERTURBED DESIGNS

C IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS

C IFILE = FILE FOR OUTPUT LIST:

C NPRINX= OUTPUT CONTROL INTEGER:

C NPRINX=0 MEANS SMALLEST AMOUNT

C NPRINX=1 MEANS MEDIUM AMOUNT

C NPRINX=2 MEANS LOTS OF OUTPUT

C

C ILOADX = ith LOADING COMBINATION

C JCOL = jth column of STRM1

C JCOL = stress component number

C PHRASE = stress component in material 1

C

C OUTPUT:

C

C STRM1(ILOADX,JCOL)

C

CHARACTER\*80 PHRASE

C INSERT ADDITIONAL COMMON BLOCKS:

COMMON/FV03/EMOD1(10),IEMOD1

REAL EMOD1

COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)

REAL EMOD2,G12,G13,G23,NU,ALPHA1

COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER

REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER

COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)

REAL ALPHA2,TEMPER,DENSTY

COMMON/FV21/PINNER(20)

REAL PINNER

COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)

REAL GENBUK,GENBUKA,GENBUKF

COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )

REAL STRM1,STRM1A,STRM1F

COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )

REAL STRM2,STRM2A,STRM2F

COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )

REAL STRM3,STRM3A,STRM3F

COMMON/IV01/NMODUL

INTEGER NMODUL

COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT

REAL TFINNR,TFOUTR,TFWEBS,WEIGHT

COMMON/FV22/PMIDDL(20),POUTER(20)

REAL PMIDDL,POUTER

C

C

C INSERT SUBROUTINE STATEMENTS HERE.

C

**COMMON/MEMSTR/STRS1F(1,5),STRS2F(1,5),STRS1V(1,5),STRS2V(1,5)**

**COMMON/FINNER/C44FIN,DELTAT,DELT,NODSEG,MSEGS**

**COMMON/ERROR/ERR**

**COMMON/NUMPAR/IPARX,IVARX,IALLOW,ICONSX,NDECX,NLINKX,NESCAP,ITYPEX**

**COMMON/LAMSTR/STRRAT(99)**

**COMMON/WRDSTR/STRWRD(99)**

**CHARACTER\*80 STRWRD**

**DIMENSION ILET1(15,6),ILET2(15,3)**

**common/caseblock/CASE**

**CHARACTER\*28 CASE**

**CHARACTER\*35 CASA**

**COMMON/STRCMP/STRC1(20,5),STRC2(20,5),STRC3(20,5)**

**C**

**C IF (JCOL.GT.1) GO TO 500**

**C**

**C NOTE IMPORTANT CHANGE:**

**C**

**C October 7, 2010: Use the MEMBRANE stresses computed in**

**C SUBROUTINE BEHX1 because the**

**C meridional curvature change, KAPPA1, from BIGBOSOR4 is**

**C sometimes much too large in the immediate neighborhoods**

**C of the ends of the shell segments, generating maximum**

**C stress components that are much too high in this particular**

**C case that involves a balloon-like (membrane) structure,**

**C This is especially true for the outer and inner curved**

**C membranes, that is, segments 2 (outer) and 4 (inner)**

**C of each module of the multi-module model. For example,**

**C here is some BIGBOSOR4 output for Segment 4 (inner**

**C curved membrane) for Load Step No. 1 (applied loads**

**C are PINNER, PMIDDL, and DELTAT):**

**C**

**C AXISYMMETRIC PRESTRESS DISTRIBUTION FOR SEGMENT 4**

**C POINT EPSILON 1 EPSILON 2 KAPPA 1 KAPPA 2**

**C MERID. CIRCUMF. MERID. CIRCUMF.**

**C STRAIN STRAIN CHANGES IN CURVATURE**

**C 1 1.070E-01 7.019E-15 5.784E-01 1.795E-08**

**C 2 1.062E-01 -1.695E-09 -1.015E+01 -4.307E-10**

**C 3 1.059E-01 1.154E-08 2.981E+00 -8.949E-09**

**C 4 1.074E-01 1.021E-08 -1.251E+00 -2.625E-09**

**C 5 1.077E-01 1.483E-08 6.715E-01 -5.319E-09**

**C 6 1.085E-01 1.647E-08 -2.629E-01 -3.690E-09**

**C 7 1.090E-01 1.910E-08 1.910E-01 -4.184E-09**

**C 8 1.096E-01 2.095E-08 -3.076E-02 -3.533E-09**

**C 9 1.100E-01 2.283E-08 7.706E-02 -3.398E-09**

**C 10 1.104E-01 2.441E-08 2.394E-02 -2.950E-09**

**C**

**C Note that the meridional change in curvature, KAPPA 1,**

**C is very large at nodal points 2, 3, 4, especially at**

**C nodal point 2. This very local edge effect gives rise to**

**C artificially high local bending meridional strain, which**

**C probably does not exist in a balloon (membrane**

**C pressure-stabilized "shell" structure). The extreme**

**C fiber meridional strain from BIGBOSOR4 is given by:**

**C EPS1 = EPSILON1 + THICK\*KAPPA1/2.**

**C in which EPSILON1 is the reference (middle) surface**

**C meridional strain and THICK is the thickness of the**

**C shell segment.**

**C**

**C Because of this spurious and extremely high meridional**

**C bending strain predicted by BIGBOSOR4 (which has difficulty**

**C predicting accurate bending stresses in membrane-like structures**

**C but which works well for shell structures with "finite"**

**C bending stiffness), the previous FORTRAN statement:**

**C IF (JCOL.GT.1) GO TO 500**

**C has been commented out and replaced by the following**

**C statement, "GO TO 500". Because of this important change**

**C the file, \*.BEHX2, is no longer created and you can**

**C therefore no longer obtain plots of the pre-buckled states**

**C at Load Steps 1 and 2 unless you remove the "C" in column**

**C 1 of the statement, "IF (JCOL.GT.1) GO TO 500", and insert**

**C a "C" in column 1 of the following statement, "GO TO 500".**

**C and then re-compile via the GENOPT command, "genprograms".**

**C**

**GO TO 500**

**C**

**IF (IMODX.EQ.0) ERR = 0.**

**IF (IMODX.EQ.1) ERR = 0.01**

**C**

**INDIC = 0**

**RAVE = RADIUS/PI**

**RBIGG = RAVE**

**C**

**CALL BOSDEC(2,24,ILOADX,INDIC)**

**C**

**IF (ITYPEX.EQ.2) THEN**

**C Get CASE.BEHX2 file for input for BIGBOSOR4...**

**C CASE.BEHX2 is an input file for BIGBOSOR4 for behavior no. 2:**

**C STRM1(ILOADX,JCOL), JCOL=1,5: stress components in material 1**

**C**

**C NOTE: Also computed in SUBROUTINE BEHX2 are the following:**

**C STRM2(ILOADX,JCOL), JCOL=1,5: stress components in material 2**

**C STRM3(ILOADX,JCOL), JCOL=1,5: stress components in material 3**

**C**

**C STRM2(ILOADX,JCOL) is available in SUBROUTINE BEHX3 because**

**C it is in a labelled common block.**

**C STRM3(ILOADX,JCOL) is available in SUBROUTINE BEHX4 because**

**C it is in a labelled common block.**

**C**

**I=INDEX(CASE,' ')**

**IF(I.NE.0) THEN**

**CASA=CASE(:I-1)//'.BEHX2'**

**ELSE**

**CASA=CASE//'.BEHX2'**

**ENDIF**

**OPEN(UNIT=61,FILE=CASA,STATUS='UNKNOWN')**

**CALL BOSDEC(2,61,ILOADX,INDIC)**

**CLOSE(UNIT=61)**

**WRITE(IFILE,'(/,/,A,A,/,A)')**

**1 ' BIGBOSOR4 input file for:',**

**1 ' stress components in materials 1,2,3',**

**1 CASA**

**ENDIF**

**C**

**CALL B4READ**

**C**

**CALL B4MAIN**

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

**C**

**C With INDIC = 0,**

**C BIGBOSOR4 generates stress constraints for laminated composite**

**C material in the following form (in this case all shell segments**

**C have only one layer, and the balloon is in tension everywhere.:**

**C**

**C\*\*\*\*\*\* (ALLOWABLE STRESS)/(ACTUAL STRESS) \*\*\*\*\*\*\*\***

**C 1 3.1045E+00 fiber tension : matl=1 , A , seg=50, node=32, layer=1 ,z=0.01 ;FS=1.**

**C 2 1.7557E+00 transv tension: matl=1 , A , seg=92, node=33, layer=1 ,z=0.01 ;FS=1.**

**C 3 5.1415E+02 fiber tension : matl=2 , A , seg=1 , node=33, layer=1 ,z=1. ;FS=1.**

**C 4 1.7557E+02 transv tension: matl=2 , A , seg=91, node=33, layer=1 ,z=1. ;FS=1.**

**C 5 4.9549E+00 fiber tension : matl=3 , A , seg=11, node=1 , layer=1 ,z=-0.01 ;FS=1.**

**C 6 1.7557E+00 transv tension: matl=3 , A , seg=90, node=33, layer=1 ,z=0.01 ;FS=1.**

**C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**C**

**C or, for an isotropic material:**

**C**

**C\*\*\*\*\*\* (ALLOWABLE STRESS)/(ACTUAL STRESS) \*\*\*\*\*\*\*\***

**C 1 1.5325E+00 effect. stress: matl=1 , A , seg=52, node=32, layer=1 ,z=0.01 ;FS=1.**

**C 2 1.0960E+00 effect. stress: matl=2 , A , seg=81, node=32, layer=1 ,z=0.01 ;FS=1.**

**C 3 1.9372E+00 effect. stress: matl=3 , A , seg=84, node=33, layer=1 ,z=0.01 ;FS=1.**

**C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**C**

**ICONST = 5\*3**

**C**

**DO 10 J = 1,5**

**STRC1(ILOADX,J) = 0.**

**STRC2(ILOADX,J) = 0.**

**STRC3(ILOADX,J) = 0.**

**10 CONTINUE**

**C**

**DO 50 I = 1,ICONST**

**IF (STRRAT(I).EQ.0.) THEN**

**ICONS2 = ICONST - 1**

**GO TO 60**

**ENDIF**

**IF (ITYPEX.EQ.2.AND.IMODX.EQ.0) THEN**

**IF (I.EQ.1) WRITE(IFILE,'(/,A,A)')**

**1 ' Maximum stress components in the entire structure',**

**1 ' at the last load step (from BIGBOSOR4):'**

**WRITE(IFILE,40) I,STRRAT(I),STRWRD(I)(1:64)**

**ENDIF**

**40 FORMAT(I3,1P,E12.4,1X,A)**

**50 CONTINUE**

**60 CONTINUE**

**C**

**DO 100 I = 1,ICONS2**

**C**

**ILET1(I,1) = INDEX(STRWRD(I),'fiber tension')**

**ILET1(I,2) = INDEX(STRWRD(I),'fiber compres')**

**ILET1(I,3) = INDEX(STRWRD(I),'transv tension')**

**ILET1(I,4) = INDEX(STRWRD(I),'transv compres')**

**ILET1(I,5) = INDEX(STRWRD(I),'in-plane shear')**

**ILET1(I,6) = INDEX(STRWRD(I),'effect. stress')**

**C**

**ILET2(I,1) = INDEX(STRWRD(I),'matl=1')**

**ILET2(I,2) = INDEX(STRWRD(I),'matl=2')**

**ILET2(I,3) = INDEX(STRWRD(I),'matl=3')**

**100 CONTINUE**

**C**

**DO 200 I = 1,ICONS2**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**IF (ILET1(I,1).NE.0) THEN**

**IF (ILET2(I,1).NE.0) STRC1(ILOADX,1)=STRM1A(ILOADX,1)/STRRAT(I)**

**IF (ILET2(I,2).NE.0) STRC2(ILOADX,1)=STRM2A(ILOADX,1)/STRRAT(I)**

**IF (ILET2(I,3).NE.0) STRC3(ILOADX,1)=STRM3A(ILOADX,1)/STRRAT(I)**

**ENDIF**

**C**

**IF (ILET1(I,2).NE.0) THEN**

**IF (ILET2(I,1).NE.0) STRC1(ILOADX,2)=STRM1A(ILOADX,2)/STRRAT(I)**

**IF (ILET2(I,2).NE.0) STRC2(ILOADX,2)=STRM2A(ILOADX,2)/STRRAT(I)**

**IF (ILET2(I,3).NE.0) STRC3(ILOADX,2)=STRM3A(ILOADX,2)/STRRAT(I)**

**ENDIF**

**C**

**IF (ILET1(I,3).NE.0) THEN**

**IF (ILET2(I,1).NE.0) STRC1(ILOADX,3)=STRM1A(ILOADX,3)/STRRAT(I)**

**IF (ILET2(I,2).NE.0) STRC2(ILOADX,3)=STRM2A(ILOADX,3)/STRRAT(I)**

**IF (ILET2(I,3).NE.0) STRC3(ILOADX,3)=STRM3A(ILOADX,3)/STRRAT(I)**

**ENDIF**

**C**

**IF (ILET1(I,4).NE.0) THEN**

**IF (ILET2(I,1).NE.0) STRC1(ILOADX,4)=STRM1A(ILOADX,4)/STRRAT(I)**

**IF (ILET2(I,2).NE.0) STRC2(ILOADX,4)=STRM2A(ILOADX,4)/STRRAT(I)**

**IF (ILET2(I,3).NE.0) STRC3(ILOADX,4)=STRM3A(ILOADX,4)/STRRAT(I)**

**ENDIF**

**C**

**IF (ILET1(I,5).NE.0) THEN**

**IF (ILET2(I,1).NE.0) STRC1(ILOADX,5)=STRM1A(ILOADX,5)/STRRAT(I)**

**IF (ILET2(I,2).NE.0) STRC2(ILOADX,5)=STRM2A(ILOADX,5)/STRRAT(I)**

**IF (ILET2(I,3).NE.0) STRC3(ILOADX,5)=STRM3A(ILOADX,5)/STRRAT(I)**

**ENDIF**

**C**

**IF (ILET1(I,6).NE.0) THEN**

**IF (ILET2(I,1).NE.0) STRC1(ILOADX,1)=STRM1A(ILOADX,1)/STRRAT(I)**

**IF (ILET2(I,2).NE.0) STRC2(ILOADX,1)=STRM2A(ILOADX,1)/STRRAT(I)**

**IF (ILET2(I,3).NE.0) STRC3(ILOADX,1)=STRM3A(ILOADX,1)/STRRAT(I)**

**ENDIF**

**C**

**200 CONTINUE**

**C**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**IF (IMODX.EQ.0) THEN**

**WRITE(IFILE,'(/,A,A,/,A,A,/,A)')**

**1 ' FIVE STRESS COMPONENTS (including bending) FOR MATL i,',**

**1 ' STRCi(ILOADX,J), J=1,5:',**

**1 ' fiber tension fiber compres ',**

**1 ' transv tension transv compres in-plane shear',**

**1 ' or effect.stress'**

**WRITE(IFILE,'(A,/,1P,5E15.4)')**

**1 ' Material 1 stress: STRC1(ILOADX,J),J=1,5)=',**

**1 (STRC1(ILOADX,J),J=1,5)**

**WRITE(IFILE,'(A,/,1P,5E15.4)')**

**1 ' Material 2 stress: STRC2(ILOADX,J),J=1,5)=',**

**1 (STRC2(ILOADX,J),J=1,5)**

**WRITE(IFILE,'(A,/,1P,5E15.4,/)')**

**1 ' Material 3 stress: STRC3(ILOADX,J),J=1,5)=',**

**1 (STRC3(ILOADX,J),J=1,5)**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**WRITE(IFILE,'(/,A,/,A,/,A,/,A,/,A,1P,E12.4,A,/,A,/,A,1P,E12.4)')**

**1' Changes in temperature required to create 2 total axial loads:',**

**1' ',**

**1' 1. Change in temperature required to create the axial thermal',**

**1' strain that generates the axial tension due to closing the',**

**1' two ends of the pressurized volume (PMIDDL=',**

**1 PMIDDL(ILOADX),')',**

**1' between the inner and outer walls of the balloon in',**

**1' Load Step No. 1: DELTAT=',**

**1 DELTAT**

**WRITE(IFILE,'(/,A,/,A,/,A,1P,1E12.4,A,1P,E12.4,/,/)')**

**1' 2. Change in temperature required to simulate the Poisson',**

**1' axial expansion caused by the application of the outer',**

**1' pressure, POUTER =',**

**1 POUTER(ILOADX),' in Load Step No. 2: DELT=',**

**1 DELT**

**WRITE(IFILE,'(A)')**

**1' BEHAVIOR OVER J = stress component number'**

**ENDIF**

**C**

**500 CONTINUE**

**C**

**C NOTE: the quantities, STRS1V and STRS2V, are computed**

**C in SUBROUTINE BEHX1.**

**C**

**STRC1(ILOADX,1) = MAX(STRS1V(1,2),STRS1V(1,4))**

**STRC1(ILOADX,2) = 0.**

**STRC1(ILOADX,3) = MAX(STRS2V(1,2),STRS2V(1,4))**

**STRC1(ILOADX,4) = 0.**

**STRC1(ILOADX,5) = 0.**

**C**

**STRM1(ILOADX,JCOL) = STRC1(ILOADX,JCOL)**

**C**

RETURN

END

C

C

C

C

C=DECK BEHX3

SUBROUTINE BEHX3

1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)

C

C PURPOSE: OBTAIN stress component in material 2

C

C YOU MUST WRITE CODE THAT, USING

C THE VARIABLES IN THE LABELLED

C COMMON BLOCKS AS INPUT, ULTIMATELY

C YIELDS THE RESPONSE VARIABLE FOR

C THE ith LOAD CASE, ILOADX:

C

C STRM2(ILOADX,JCOL)

C

C AS OUTPUT. THE ith CASE REFERS

C TO ith ENVIRONMENT (e.g. load com-

C bination).

C THE jth COLUMN (JCOL)

C INDEX IS DEFINED AS FOLLOWS:

C stress component number

C

C DEFINITIONS OF INPUT DATA:

C IMODX = DESIGN CONTROL INTEGER:

C IMODX = 0 MEANS BASELINE DESIGN

C IMODX = 1 MEANS PERTURBED DESIGN

C IFAST = 0 MEANS FEW SHORTCUTS FOR PERTURBED DESIGNS

C IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS

C IFILE = FILE FOR OUTPUT LIST:

C NPRINX= OUTPUT CONTROL INTEGER:

C NPRINX=0 MEANS SMALLEST AMOUNT

C NPRINX=1 MEANS MEDIUM AMOUNT

C NPRINX=2 MEANS LOTS OF OUTPUT

C

C ILOADX = ith LOADING COMBINATION

C JCOL = jth column of STRM2

C JCOL = stress component number

C PHRASE = stress component in material 2

C

C OUTPUT:

C

C STRM2(ILOADX,JCOL)

C

CHARACTER\*80 PHRASE

C INSERT ADDITIONAL COMMON BLOCKS:

COMMON/FV03/EMOD1(10),IEMOD1

REAL EMOD1

COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)

REAL EMOD2,G12,G13,G23,NU,ALPHA1

COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER

REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER

COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)

REAL ALPHA2,TEMPER,DENSTY

COMMON/FV21/PINNER(20)

REAL PINNER

COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)

REAL GENBUK,GENBUKA,GENBUKF

COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )

REAL STRM1,STRM1A,STRM1F

COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )

REAL STRM2,STRM2A,STRM2F

COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )

REAL STRM3,STRM3A,STRM3F

COMMON/IV01/NMODUL

INTEGER NMODUL

COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT

REAL TFINNR,TFOUTR,TFWEBS,WEIGHT

COMMON/FV22/PMIDDL(20),POUTER(20)

REAL PMIDDL,POUTER

C

C

C INSERT SUBROUTINE STATEMENTS HERE.

C

**COMMON/MEMSTR/STRS1F(1,5),STRS2F(1,5),STRS1V(1,5),STRS2V(1,5)**

**COMMON/STRCMP/STRC1(20,5),STRC2(20,5),STRC3(20,5)**

**C**

**C Do not need any calculations added here because**

**C STRC2(ILOADX,JCOL), JCOL = 1,2,3,4,5**

**C are computed in SUBROUTINE BEHX2**

**C**

**C NOTE IMPORTANT CHANGE:**

**C**

**C October 7, 2010: Use the MEMBRANE stresses because the**

**C meridional curvature change, KAPPA1, from BIGBOSOR4 is**

**C sometimes much too large in the immediate neighborhoods**

**C of the ends of the shell segments, generating maximum**

**C stress components that are much too high in this particular**

**C case that involves a balloon-like (membrane) structure,**

**C This is especially true for the outer and inner curved**

**C membranes, that is, segments 2 (outer) and 4 (inner)**

**C of each module of the multi-module model. For example,**

**C here is some BIGBOSOR4 output for Segment 4 (inner**

**C curved membrane) for Load Step No. 1:**

**C**

**C AXISYMMETRIC PRESTRESS DISTRIBUTION FOR SEGMENT 4**

**C POINT EPSILON 1 EPSILON 2 KAPPA 1 KAPPA 2**

**C MERID. CIRCUMF. MERID. CIRCUMF.**

**C STRAIN STRAIN CHANGES IN CURVATURE**

**C 1 1.070E-01 7.019E-15 5.784E-01 1.795E-08**

**C 2 1.062E-01 -1.695E-09 -1.015E+01 -4.307E-10**

**C 3 1.059E-01 1.154E-08 2.981E+00 -8.949E-09**

**C 4 1.074E-01 1.021E-08 -1.251E+00 -2.625E-09**

**C 5 1.077E-01 1.483E-08 6.715E-01 -5.319E-09**

**C 6 1.085E-01 1.647E-08 -2.629E-01 -3.690E-09**

**C 7 1.090E-01 1.910E-08 1.910E-01 -4.184E-09**

**C 8 1.096E-01 2.095E-08 -3.076E-02 -3.533E-09**

**C 9 1.100E-01 2.283E-08 7.706E-02 -3.398E-09**

**C 10 1.104E-01 2.441E-08 2.394E-02 -2.950E-09**

**C**

**C**

**C NOTE: the quantities, STRS1V and STRS2V, are computed**

**C in SUBROUTINE BEHX1.**

**C**

**STRC2(ILOADX,1) = MAX(STRS1V(1,1),STRS1V(1,3))**

**STRC2(ILOADX,2) = 0.**

**STRC2(ILOADX,3) = MAX(STRS2V(1,1),STRS2V(1,3))**

**STRC2(ILOADX,4) = 0.**

**STRC2(ILOADX,5) = 0.**

**C**

**STRM2(ILOADX,JCOL) = STRC2(ILOADX,JCOL)**

**C**

C

RETURN

END

C

C

C

C

C=DECK BEHX4

SUBROUTINE BEHX4

1 (IFILE,NPRINX,IMODX,IFAST,ILOADX,JCOL,PHRASE)

C

C PURPOSE: OBTAIN stress component in material 3

C

C YOU MUST WRITE CODE THAT, USING

C THE VARIABLES IN THE LABELLED

C COMMON BLOCKS AS INPUT, ULTIMATELY

C YIELDS THE RESPONSE VARIABLE FOR

C THE ith LOAD CASE, ILOADX:

C

C STRM3(ILOADX,JCOL)

C

C AS OUTPUT. THE ith CASE REFERS

C TO ith ENVIRONMENT (e.g. load com-

C bination).

C THE jth COLUMN (JCOL)

C INDEX IS DEFINED AS FOLLOWS:

C stress component number

C

C DEFINITIONS OF INPUT DATA:

C IMODX = DESIGN CONTROL INTEGER:

C IMODX = 0 MEANS BASELINE DESIGN

C IMODX = 1 MEANS PERTURBED DESIGN

C IFAST = 0 MEANS FEW SHORTCUTS FOR PERTURBED DESIGNS

C IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS

C IFILE = FILE FOR OUTPUT LIST:

C NPRINX= OUTPUT CONTROL INTEGER:

C NPRINX=0 MEANS SMALLEST AMOUNT

C NPRINX=1 MEANS MEDIUM AMOUNT

C NPRINX=2 MEANS LOTS OF OUTPUT

C

C ILOADX = ith LOADING COMBINATION

C JCOL = jth column of STRM3

C JCOL = stress component number

C PHRASE = stress component in material 3

C

C OUTPUT:

C

C STRM3(ILOADX,JCOL)

C

CHARACTER\*80 PHRASE

C INSERT ADDITIONAL COMMON BLOCKS:

COMMON/FV03/EMOD1(10),IEMOD1

REAL EMOD1

COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)

REAL EMOD2,G12,G13,G23,NU,ALPHA1

COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER

REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER

COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)

REAL ALPHA2,TEMPER,DENSTY

COMMON/FV21/PINNER(20)

REAL PINNER

COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)

REAL GENBUK,GENBUKA,GENBUKF

COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )

REAL STRM1,STRM1A,STRM1F

COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )

REAL STRM2,STRM2A,STRM2F

COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )

REAL STRM3,STRM3A,STRM3F

COMMON/IV01/NMODUL

INTEGER NMODUL

COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT

REAL TFINNR,TFOUTR,TFWEBS,WEIGHT

COMMON/FV22/PMIDDL(20),POUTER(20)

REAL PMIDDL,POUTER

C

C

C INSERT SUBROUTINE STATEMENTS HERE.

C

**COMMON/MEMSTR/STRS1F(1,5),STRS2F(1,5),STRS1V(1,5),STRS2V(1,5)**

**COMMON/STRCMP/STRC1(20,5),STRC2(20,5),STRC3(20,5)**

**C**

**C Do not need any calculations added here because**

**C STRC3(ILOADX,JCOL), JCOL = 1,2,3,4,5**

**C are computed in SUBROUTINE BEHX2**

**C**

**C NOTE IMPORTANT CHANGE:**

**C**

**C October 7, 2010: Use the MEMBRANE stresses because the**

**C meridional curvature change, KAPPA1, from BIGBOSOR4 is**

**C sometimes much too large in the immediate neighborhoods**

**C of the ends of the shell segments, generating maximum**

**C stress components that are much too high in this particular**

**C case that involves a balloon-like (membrane) structure,**

**C This is especially true for the outer and inner curved**

**C membranes, that is, segments 2 (outer) and 4 (inner)**

**C of each module of the multi-module model. For example,**

**C here is some BIGBOSOR4 output for Segment 4 (inner**

**C curved membrane) for Load Step No. 1:**

**C**

**C AXISYMMETRIC PRESTRESS DISTRIBUTION FOR SEGMENT 4**

**C POINT EPSILON 1 EPSILON 2 KAPPA 1 KAPPA 2**

**C MERID. CIRCUMF. MERID. CIRCUMF.**

**C STRAIN STRAIN CHANGES IN CURVATURE**

**C 1 1.070E-01 7.019E-15 5.784E-01 1.795E-08**

**C 2 1.062E-01 -1.695E-09 -1.015E+01 -4.307E-10**

**C 3 1.059E-01 1.154E-08 2.981E+00 -8.949E-09**

**C 4 1.074E-01 1.021E-08 -1.251E+00 -2.625E-09**

**C 5 1.077E-01 1.483E-08 6.715E-01 -5.319E-09**

**C 6 1.085E-01 1.647E-08 -2.629E-01 -3.690E-09**

**C 7 1.090E-01 1.910E-08 1.910E-01 -4.184E-09**

**C 8 1.096E-01 2.095E-08 -3.076E-02 -3.533E-09**

**C 9 1.100E-01 2.283E-08 7.706E-02 -3.398E-09**

**C 10 1.104E-01 2.441E-08 2.394E-02 -2.950E-09**

**C**

**C**

**C NOTE: the quantities, STRS1V and STRS2V, are computed**

**C in SUBROUTINE BEHX1.**

**C**

**STRC3(ILOADX,1) = STRS1V(1,5)**

**STRC3(ILOADX,2) = 0.**

**STRC3(ILOADX,3) = STRS2V(1,5)**

**STRC3(ILOADX,4) = 0.**

**STRC3(ILOADX,5) = 0.**

**C**

**STRM3(ILOADX,JCOL) = STRC3(ILOADX,JCOL)**

**C**

C

RETURN

END

C

C

C

C

C=DECK USRCON

SUBROUTINE USRCON(INUMTT,IMODX,CONMAX,ICONSX,IPOINC,CONSTX,

1 WORDCX,WORDMX,PCWORD,CPLOTX,ICARX,IFILEX)

C PURPOSE: GENERATE USER-WRITTEN

C INEQUALITY CONSTRAINT CONDITION

C USING ANY COMBINATION OF PROGRAM

C VARIABLES.

C YOU MUST WRITE CODE THAT, USING

C THE VARIABLES IN THE LABELLED

C COMMON BLOCKS AS INPUT, ULTIMATELY

C YIELDS A CONSTRAINT CONDITION,

C CALLED "CONX" IN THIS ROUTINE.

DIMENSION WORDCX(\*),WORDMX(\*),IPOINC(\*),CONSTX(\*)

DIMENSION PCWORD(\*),CPLOTX(\*)

CHARACTER\*80 WORDCX,WORDMX,PCWORD

C INSERT ADDITIONAL COMMON BLOCKS:

COMMON/FV03/EMOD1(10),IEMOD1

REAL EMOD1

COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)

REAL EMOD2,G12,G13,G23,NU,ALPHA1

COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER

REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER

COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)

REAL ALPHA2,TEMPER,DENSTY

COMMON/FV21/PINNER(20)

REAL PINNER

COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)

REAL GENBUK,GENBUKA,GENBUKF

COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )

REAL STRM1,STRM1A,STRM1F

COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )

REAL STRM2,STRM2A,STRM2F

COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )

REAL STRM3,STRM3A,STRM3F

COMMON/IV01/NMODUL

INTEGER NMODUL

COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT

REAL TFINNR,TFOUTR,TFWEBS,WEIGHT

COMMON/FV22/PMIDDL(20),POUTER(20)

REAL PMIDDL,POUTER

C

CONX = 0.0

C

C INSERT USER-WRITTEN STATEMENTS

C HERE. THE CONSTRAINT CONDITION

C THAT YOU CALCULATE IS CALLED "CONX"

C

IF (CONX.EQ.0.0) RETURN

IF (CONX.LT.0.0) THEN

WRITE(IFILEX,\*)' CONX MUST BE GREATER THAN ZERO.'

CALL EXIT

ENDIF

C

C DO NOT CHANGE THE FOLLOWING STATEMENTS, EXCEPT WORDC

C

ICARX = ICARX + 1

INUMTT = INUMTT + 1

WORDCX(ICARX) = ' USER: PROVIDE THIS.'

CPLOTX(ICARX) = CONX - 1.

CALL BLANKX(WORDCX(ICARX),IENDP)

PCWORD(ICARX) = WORDCX(ICARX)(1:IENDP)//' -1'

IF (IMODX.EQ.0.AND.CONX.GT.CONMAX) GO TO 200

IF (IMODX.EQ.1.AND.IPOINC(INUMTT).EQ.0) GO TO 200

ICONSX = ICONSX + 1

IF (IMODX.EQ.0) IPOINC(INUMTT) = 1

CONSTX(ICONSX) = CONX

WORDMX(ICONSX) = WORDCX(ICARX)(1:IENDP)//' -1'

200 CONTINUE

C END OF USRCON

C

C

RETURN

END

C

C

C

C=DECK USRLNK

SUBROUTINE USRLNK(VARI,I,VARIAB)

C Purpose: generate user-written

C linking conditions using any

C combination of decision variables.

C You must write conde that, using

C the variables in the subroutine

C argument VARIAB as input, ultimately

C yield a value for the linked variable

C VARI.

C

C VARI is the Ith entry of the array

C VARIAB. You have decided that this

C is to be a linked variable with user

C defined linking. It is linked to

C the decision variables in the array

C VARIAB.

C An example will provide the simplest

C explanation of this:

C Let"s say that the 5th decision

C variable candidate (I=5) is linked

C to the decision variable candidates

C 2 and 7. (You used DECIDE to select

C these as decision variables.

C In this case VARI is equal to

C VARIAB(I). You then write your

C linking equation in the form

C VARI=f(VARIAB(2),VARIAB(7)).

C Use the index I in an IF statement if

C you have more than one user-defined

C linked variable.

C

C

REAL VARI,VARIAB(50)

INTEGER I

C

C INSERT USER-WRITTEN DECLARATION

C STATEMENTS HERE.

C

C INSERT USER-WRITTEN

C STATEMENTS HERE.

C

C

C END OF USRLNK

RETURN

END

C=DECK OBJECT

SUBROUTINE OBJECT(IFILE,NPRINX,IMODX,OBJGEN,PHRASE)

C PURPOSE:weight/length of the balloon

C

C YOU MUST WRITE CODE THAT, USING

C THE VARIABLES IN THE LABELLED

C COMMON BLOCKS AS INPUT, ULTIMATELY

C YIELDS THE OBJECTIVE FUNCTION

C WEIGHT

C AS OUTPUT. MAKE SURE TO INCLUDE AT

C THE END OF THE SUBROUTINE, THE

C STATEMENT: OBJGEN = WEIGHT

C

C

C DEFINITIONS OF INPUT DATA:

C IMODX = DESIGN CONTROL INTEGER:

C IMODX = 0 MEANS BASELINE DESIGN

C IMODX = 1 MEANS PERTURBED DESIGN

C IFAST = 0 MEANS FEW SHORTCUTS FOR PERTURBED DESIGNS

C IFAST = 1 MEANS MORE SHORTCUTS FOR PERTURBED DESIGNS

C IFILE = FILE FOR OUTPUT LIST:

C NPRINX= OUTPUT CONTROL INTEGER:

C NPRINX=0 MEANS SMALLEST AMOUNT

C NPRINX=1 MEANS MEDIUM AMOUNT

C NPRINX=2 MEANS LOTS OF OUTPUT

C

C DEFINITION OF PHRASE:

C PHRASE = weight/length of the balloon

C

CHARACTER\*80 PHRASE

C INSERT ADDITIONAL COMMON BLOCKS:

COMMON/FV03/EMOD1(10),IEMOD1

REAL EMOD1

COMMON/FV04/EMOD2(10),G12(10),G13(10),G23(10),NU(10),ALPHA1(10)

REAL EMOD2,G12,G13,G23,NU,ALPHA1

COMMON/FV01/LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER

REAL LENGTH,RADIUS,HEIGHT,RINNER,ROUTER,TINNER,TOUTER

COMMON/FV10/ALPHA2(10),TEMPER(10),DENSTY(10)

REAL ALPHA2,TEMPER,DENSTY

COMMON/FV21/PINNER(20)

REAL PINNER

COMMON/FV26/GENBUK(20),GENBUKA(20),GENBUKF(20)

REAL GENBUK,GENBUKA,GENBUKF

COMMON/FV29/STRM1(20,5 ),JSTRM1 ,STRM1A(20,5 ),STRM1F(20,5 )

REAL STRM1,STRM1A,STRM1F

COMMON/FV32/STRM2(20,5 ),STRM2A(20,5 ),STRM2F(20,5 )

REAL STRM2,STRM2A,STRM2F

COMMON/FV35/STRM3(20,5 ),STRM3A(20,5 ),STRM3F(20,5 )

REAL STRM3,STRM3A,STRM3F

COMMON/IV01/NMODUL

INTEGER NMODUL

COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT

REAL TFINNR,TFOUTR,TFWEBS,WEIGHT

COMMON/FV22/PMIDDL(20),POUTER(20)

REAL PMIDDL,POUTER

C

C

C INSERT SUBROUTINE STATEMENTS HERE.

C

**C Get the weight per axial length of the balloon.**

**C The quantity, WEIGHT, is computed as follows in**

**C SUBROUTINE BOSDEC:**

**C**

**C WEIGHT = 4.\*(ARCOUT\*TOUTER\*DENSTY(1) +ARCINR\*TINNER\*DENSTY(1)**

**C 1 +ARCFOT\*TFOUTR\*DENSTY(2) +ARCFIN\*TFINNR\*DENSTY(2)**

**C 1 +ARCWEB\*TFWEBS\*DENSTY(3))**

**C**

**C in which**

**C ARCOUT = total arc length of the outer curved membranes**

**C ARCINR = total arc length of the inner curved membranes**

**C ARCFOT = total arc length of the outer flat membranes**

**C ARCFIN = total arc length of the inner flat membranes**

**C ARCWEB = total length of the slanted webs.**

**C**

**OBJGEN =WEIGHT**

**C**

C

RETURN

END

C

C

C