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

**==========================================================================**

**C=DECK BOSDEC**

**C**

**C PURPOSE IS TO SET UP BIGBOSOR4 INPUT FILE FOR "balloon"**

**C**

**SUBROUTINE BOSDEC(INDX,IFIL14,ILOADX,INDIC)**

**C**

**C Insert labelled common blocks: balloon.COM**

**COMMON/FV03/EMOD1(10),IEMOD1**

**REAL EMOD1**

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

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

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

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

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

**REAL ALPHA2,TEMPER,DENSTY**

**COMMON/FV21/PINNER(20)**

**REAL PINNER**

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

**REAL GENBUK,GENBUKA,GENBUKF**

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

**REAL STRM1,STRM1A,STRM1F**

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

**REAL STRM2,STRM2A,STRM2F**

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

**REAL STRM3,STRM3A,STRM3F**

**COMMON/IV01/NMODUL**

**INTEGER NMODUL**

**COMMON/FV18/TFINNR,TFOUTR,TFWEBS,WEIGHT**

**REAL TFINNR,TFOUTR,TFWEBS,WEIGHT**

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

**REAL PMIDDL,POUTER**

**C end of trusscomp.COM**

**COMMON/ITRYX/ITRY**

**COMMON/FLNFLO/FLINNR,FLOUTR**

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

**REAL N2DIFF**

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

**COMMON/MODULX/MODULL,MODULG**

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

**COMMON/NUMSEG/NSEGS**

**common/caseblock/CASE**

**CHARACTER\*28 CASE**

**CHARACTER\*35 CASA2,CASA3**

**CHARACTER\*10 CN**

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

**DIMENSION RA(40),RB(40),ZA(40),ZB(40)**

**DIMENSION RCL(40),RDL(40),ZCL(40),ZDL(40)**

**DIMENSION R1(8,40),R2(8,40),Z1(8,40),Z2(8,40)**

**DIMENSION RC(8,40),ZC(8,40),SROT(8,40),PFIXED(8,40),PEIGEN(8,40)**

**DIMENSION MATTYP(8,40),THICK(8,40),MATLJ(295),THICKJ(295)**

**DIMENSION LAYTYP(8,40),LTYPEJ(295)**

**DIMENSION R1J(295),Z1J(295),R2J(295),Z2J(295),RCJ(295),ZCJ(295)**

**DIMENSION NODJ(295),NSHPJ(295),SROTJ(295),PFIXJ(295),PEIGJ(295)**

**DIMENSION NSHAPE(8,40),NGRND(8,40),NODGRD(8,40),NPREV(8,40)**

**DIMENSION JPREV1(8,40),NDCUR1(8,40),NDPRV1(8,40)**

**DIMENSION JPREV2(8,40),NDCUR2(8,40),NDPRV2(8,40)**

**DIMENSION NGRNDJ(295),NODGRJ(295),NPREVJ(295)**

**DIMENSION JPREVJ(295,2),NDCURJ(295,2),NDPRVJ(295,2)**

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

**REWIND IFIL14**

**C**

**WRITE(IFILE4,3)**

**3 FORMAT(//' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* BOSDEC \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'/**

**1' The purpose of BOSDEC is to set up an input file, NAME.ALL,'/**

**1' for a cylindrical shell. NAME is your name for'/**

**1' the case. The file NAME.ALL is a BOSOR4 input "deck" used'/**

**1' by SUBROUTINE B4READ.'/**

**1' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'/)**

**C**

**NSEGS = 6**

**PI = 3.1415927**

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

**NMODUL = NMODUL**

**RBEG = RBIGG**

**ENDIF**

**IF (INDX.EQ.1) THEN**

**NMODUL = NMODUL**

**RBEG = RBIGG**

**ENDIF**

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

**NMODUL = NMODUL**

**RBEG = RBIGG**

**ENDIF**

**RSTART = RBEG**

**C**

**CALL MOVER(0.,0,R1,1,320)**

**CALL MOVER(0.,0,Z1,1,320)**

**CALL MOVER(0.,0,R2,1,320)**

**CALL MOVER(0.,0,Z2,1,320)**

**CALL MOVER(0.,0,RC,1,320)**

**CALL MOVER(0.,0,ZC,1,320)**

**CALL MOVER(0.,0,PFIXED,1,320)**

**CALL MOVER(0.,0,PEIGEN,1,320)**

**CALL MOVER(0,0,NGRND,1,320)**

**CALL MOVER(0,0,NODGRD,1,320)**

**CALL MOVER(0,0,NPREV,1,320)**

**CALL MOVER(0,0,JPREV1,1,320)**

**CALL MOVER(0,0,NDCUR1,1,320)**

**CALL MOVER(0,0,NDPRV1,1,320)**

**CALL MOVER(0,0,JPREV2,1,320)**

**CALL MOVER(0,0,NDCUR2,1,320)**

**CALL MOVER(0,0,NDPRV2,1,320)**

**C**

**C Single truss-core sandwich 6-segment module...**

**C**

**IF (IMODX.EQ.0) MODULL = NMODUL**

**MODULS = MODULL**

**C**

**C The number of modules, "MODULG" should be such that the**

**C model for general buckling subtends close to 90 degrees**

**C of circumference...**

**C IF (IMODX.EQ.0) MODULG = (PI/1.89)\*RADIUS/PITCH**

**C MODULS = MODULG**

**C IF (IMODX.EQ.0.AND.NSEGSM\*MODULS.GT.288) THEN**

**C WRITE(IFILE4,'(/,A,/,A,2I5,/,A,/,A,/,A,)')**

**C 1 ' \*\*\*\*\*\*\*\*\*\*\*\*\*\* TOO MANY MODULES \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*',**

**C 1 ' MODULS, MSEGS = NSEGSM\*MODULS = ',MODULS, NSEGSM\*MODULS,**

**C 1 ' MSEGS = Too many segments. Maximum number is 288.',**

**C 1 ' Now reducing the number of modules in the model.',**

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

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**C RATIOM = FLOAT(280)/FLOAT(NSEGSM\*MODULS)**

**C MODULG = RATIOM\*MODULG**

**C MODULS = MODULG**

**C ENDIF**

**C**

**C**

**C INDX = 1 means SUBROUTINE BOSDEC generates GENERAL buckling model**

**C INDX = 2 means SUBROUTINE BOSDEC computes maximum stresses.**

**C**

**C IMODUL = the module number in a multi-module model**

**C MODULS = number of modules in the model**

**C RBEG = horizontal radius to the beginning of the first module**

**C in the multi-module model, where**

**C 1. for the LOCAL buckling model:**

**C RBEG = RBIGL = RAVE -FLOAT(NMODULL)\*PITCH/2.**

**C in which RAVE = 100.\*LENGTH\*FACLEN/PI**

**C 2. for the GENERAL buckling model:**

**C RBEG = RBIGG = RAVE -0.707\*RADIUS**

**C in which RAVE = 100.\*LENGTH/PI**

**C where**

**C NMODUL = number of modules in the buckling model**

**C LENGTH = the total length of the cylindrical shell**

**C FACLEN = fraction of LENGTH used for local buckling model**

**C RADIUS = radius of the cylindrical shell measured to**

**C the innermost face sheet.**

**C**

**C NSHAPE = BIGBOSOR4 index for shape of shell segment:**

**C NSHAPE = 1 means cone, cylinder, flat plate**

**C NSHAPE = 2 means toroidal, spherical**

**C SROT = BIGBOSOR4 index for direction of travel along**

**C a spherical or toroidal shell segment:**

**C SROT = 1 means clockwise travel**

**C SROT =-1 means anticlockwise travel**

**C**

**C R1(i),R2(i),Z1(i),Z2(i),RC(i),ZC(i) =**

**C (r,z) end points and center of curvature (rc,zc)**

**C for the ith shell segment**

**C**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**C**

**DANGLE = 0.5\*PI/FLOAT(2\*MODULS)**

**FLOUTR = 2.\*(RADIUS + HEIGHT)\*SIN(DANGLE)**

**FLINNR = 2.\*RADIUS\*SIN(DANGLE)**

**PHIOUT = ASIN(0.5\*FLOUTR/ROUTER)**

**PHIINR = ASIN(0.5\*FLINNR/RINNER)**

**DSMALO = (RADIUS+HEIGHT)\*(1.-COS(DANGLE))**

**DSMALI = RADIUS\*(1.-COS(DANGLE))**

**RCOUTR = RADIUS+HEIGHT -DSMALO - ROUTER\*COS(PHIOUT)**

**RCINNR = RADIUS -DSMALI + RINNER\*COS(PHIINR)**

**C**

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

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' DANGLE=',DANGLE**

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' FLOUTR=',FLOUTR**

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' FLINNR=',FLINNR**

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' PHIOUT=',PHIOUT**

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' PHIINR=',PHIINR**

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' DSMALO=',DSMALO**

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' DSMALI=',DSMALI**

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' RCOUTR=',RCOUTR**

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' RCINNR=',RCINNR**

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' RSTART=',RSTART**

**ENDIF**

**C CALL EXIT**

**C**

**NODSEG = 31**

**C**

**DO 100 IMODUL = 1,MODULS**

**C**

**FMODUL = IMODUL**

**ANGLE = DANGLE\*(2.\*FMODUL -1.0)**

**ANGLEM = ANGLE - DANGLE**

**ANGLEP = ANGLE + DANGLE**

**RA(IMODUL) = RSTART + (RADIUS+HEIGHT)\*SIN(ANGLE)**

**ZA(IMODUL) = (RADIUS + HEIGHT)\*COS(ANGLE)**

**RB(IMODUL) = RSTART + RADIUS\*SIN(ANGLEP)**

**ZB(IMODUL) = RADIUS\*COS(ANGLEP)**

**RCL(IMODUL)= RSTART + RCOUTR\*SIN(ANGLEM)**

**ZCL(IMODUL)= RCOUTR\*COS(ANGLEM)**

**RDL(IMODUL)= RSTART + RCINNR\*SIN(ANGLE)**

**ZDL(IMODUL)= RCINNR\*COS(ANGLE)**

**C**

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

**WRITE(IFILE4,'(A,I5)')**

**1 ' IMODUL=',IMODUL**

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' ANGLE =',ANGLE**

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' ANGLEM=',ANGLEM**

**WRITE(IFILE4,'(A,1P,E12.4)')**

**1 ' ANGLEP=',ANGLEP**

**WRITE(IFILE4,'(A,I2,A,1P,E12.4)')**

**1 ' RA(',IMODUL,') =',RA(IMODUL)**

**WRITE(IFILE4,'(A,I2,A,1P,E12.4)')**

**1 ' ZA(',IMODUL,') =',ZA(IMODUL)**

**WRITE(IFILE4,'(A,I2,A,1P,E12.4)')**

**1 ' RB(',IMODUL,') =',RB(IMODUL)**

**WRITE(IFILE4,'(A,I2,A,1P,E12.4)')**

**1 ' ZB(',IMODUL,') =',ZB(IMODUL)**

**WRITE(IFILE4,'(A,I2,A,1P,E12.4)')**

**1 ' RCL(',IMODUL,')=',RCL(IMODUL)**

**WRITE(IFILE4,'(A,I2,A,1P,E12.4)')**

**1 ' ZCL(',IMODUL,')=',ZCL(IMODUL)**

**WRITE(IFILE4,'(A,I2,A,1P,E12.4)')**

**1 ' RDL(',IMODUL,')=',RDL(IMODUL)**

**WRITE(IFILE4,'(A,I2,A,1P,E12.4)')**

**1 ' ZDL(',IMODUL,')=',ZDL(IMODUL)**

**ENDIF**

**100 CONTINUE**

**C**

**C CALL EXIT**

**C**

**DO 110 IMODUL = 1,MODULS**

**C**

**C Segment 1 in module, IMODUL**

**IMODUL1 = IMODUL - 1**

**NSHAPE(1,IMODUL) = 1**

**IF (IMODUL.GT.1) THEN**

**NGRND(1,IMODUL) = 0**

**NPREV(1,IMODUL) = 1**

**JPREV1(1,IMODUL) = -6**

**NDCUR1(1,IMODUL) = 1**

**NDPRV1(1,IMODUL) = NODSEG**

**R1(1,IMODUL) = RA(IMODUL1)**

**Z1(1,IMODUL) = ZA(IMODUL1)**

**ELSE**

**NGRND(1,IMODUL) = 1**

**NODGRD(1,IMODUL) = 1**

**NPREV(1,IMODUL) = 0**

**R1(1,IMODUL) = RSTART**

**Z1(1,IMODUL) = RADIUS + HEIGHT - DSMALO**

**ENDIF**

**R2(1,IMODUL) = RA(IMODUL)**

**Z2(1,IMODUL) = ZA(IMODUL)**

**MATTYP(1,IMODUL) = 2**

**THICK(1,IMODUL) = TFOUTR**

**LAYTYP(1,IMODUL) = 1**

**C**

**C Segment 2 in module, IMODUL**

**NSHAPE(2,IMODUL) = 2**

**SROT(2,IMODUL) = 1.**

**IF (IMODUL.GT.1) THEN**

**NGRND(2,IMODUL) = 0**

**NPREV(2,IMODUL) = 2**

**JPREV1(2,IMODUL) = -7**

**NDCUR1(2,IMODUL) = 1**

**NDPRV1(2,IMODUL) = NODSEG**

**JPREV2(2,IMODUL) = -1**

**NDCUR2(2,IMODUL) = NODSEG**

**NDPRV2(2,IMODUL) = NODSEG**

**R1(2,IMODUL) = RA(IMODUL1)**

**Z1(2,IMODUL) = ZA(IMODUL1)**

**ELSE**

**NGRND(2,IMODUL) = 1**

**NODGRD(2,IMODUL) = 1**

**NPREV(2,IMODUL) = 1**

**JPREV1(2,IMODUL) = -1**

**NDCUR1(2,IMODUL) = NODSEG**

**NDPRV1(2,IMODUL) = NODSEG**

**R1(2,IMODUL) = RSTART**

**Z1(2,IMODUL) = RCOUTR + ROUTER**

**ENDIF**

**R2(2,IMODUL) = RA(IMODUL)**

**Z2(2,IMODUL) = ZA(IMODUL)**

**RC(2,IMODUL) = RCL(IMODUL)**

**ZC(2,IMODUL) = ZCL(IMODUL)**

**MATTYP(2,IMODUL) = 1**

**THICK(2,IMODUL) = TOUTER**

**LAYTYP(2,IMODUL) = 2**

**PFIXED(2,IMODUL) = -PMIDDL(ILOADX)**

**PEIGEN(2,IMODUL) = POUTER(ILOADX)**

**C**

**C Segment 3 in module, IMODUL**

**NSHAPE(3,IMODUL) = 1**

**IF (IMODUL.GT.1) THEN**

**NGRND(3,IMODUL) = 0**

**IF (IMODUL.EQ.MODULS) THEN**

**NGRND(3,IMODUL) = 1**

**NODGRD(3,IMODUL) = NODSEG**

**ENDIF**

**NPREV(3,IMODUL) = 1**

**JPREV1(3,IMODUL) = -6**

**NDCUR1(3,IMODUL) = 1**

**NDPRV1(3,IMODUL) = NODSEG**

**R1(3,IMODUL) = RB(IMODUL1)**

**Z1(3,IMODUL) = ZB(IMODUL1)**

**ELSE**

**NGRND(3,IMODUL) = 1**

**NODGRD(3,IMODUL) = 1**

**NPREV(3,IMODUL) = 0**

**R1(3,IMODUL) = RSTART**

**Z1(3,IMODUL) = RADIUS**

**ENDIF**

**R2(3,IMODUL) = RB(IMODUL)**

**Z2(3,IMODUL) = ZB(IMODUL)**

**MATTYP(3,IMODUL) = 2**

**THICK(3,IMODUL) = TFINNR**

**LAYTYP(3,IMODUL) = 3**

**C**

**C Segment 4 in module, IMODUL**

**NGRND(4,IMODUL) = 0**

**NPREV(4,IMODUL) = 2**

**IF (IMODUL.EQ.1) JPREV1(4,IMODUL) = -1**

**IF (IMODUL.GT.1) JPREV1(4,IMODUL) = -7**

**NDCUR1(4,IMODUL) = 1**

**IF (IMODUL.EQ.1) NDPRV1(4,IMODUL) = 1**

**IF (IMODUL.GT.1) NDPRV1(4,IMODUL) = NODSEG**

**JPREV2(4,IMODUL) = -1**

**NDCUR2(4,IMODUL) = NODSEG**

**NDPRV2(4,IMODUL) = NODSEG**

**NSHAPE(4,IMODUL) = 2**

**SROT(4,IMODUL) = -1.**

**IF (IMODUL.GT.1) THEN**

**R1(4,IMODUL) = RB(IMODUL1)**

**Z1(4,IMODUL) = ZB(IMODUL1)**

**ELSE**

**R1(4,IMODUL) = RSTART**

**Z1(4,IMODUL) = RADIUS**

**ENDIF**

**R2(4,IMODUL) = RB(IMODUL)**

**Z2(4,IMODUL) = ZB(IMODUL)**

**MATTYP(4,IMODUL) = 1**

**THICK(4,IMODUL) = TINNER**

**LAYTYP(4,IMODUL) = 4**

**RC(4,IMODUL) = RDL(IMODUL)**

**ZC(4,IMODUL) = ZDL(IMODUL)**

**PFIXED(4,IMODUL) = PMIDDL(ILOADX) - PINNER(ILOADX)**

**PEIGEN(4,IMODUL) = 0.**

**C**

**C Segment 5 in module, IMODUL**

**NGRND(5,IMODUL) = 0**

**NPREV(5,IMODUL) = 2**

**IF (IMODUL.EQ.1) JPREV1(5,IMODUL) = -2**

**IF (IMODUL.GT.1) JPREV1(5,IMODUL) = -8**

**NDCUR1(5,IMODUL) = 1**

**IF (IMODUL.EQ.1) NDPRV1(5,IMODUL) = 1**

**IF (IMODUL.GT.1) NDPRV1(5,IMODUL) = NODSEG**

**JPREV2(5,IMODUL) = -4**

**NDCUR2(5,IMODUL) = NODSEG**

**NDPRV2(5,IMODUL) = NODSEG**

**NSHAPE(5,IMODUL) = 1**

**IF (IMODUL.GT.1) THEN**

**R1(5,IMODUL) = RB(IMODUL1)**

**Z1(5,IMODUL) = ZB(IMODUL1)**

**ELSE**

**R1(5,IMODUL) = RSTART**

**Z1(5,IMODUL) = RADIUS**

**ENDIF**

**R2(5,IMODUL) = RA(IMODUL)**

**Z2(5,IMODUL) = ZA(IMODUL)**

**MATTYP(5,IMODUL) = 3**

**THICK(5,IMODUL) = TFWEBS**

**LAYTYP(5,IMODUL) = 5**

**C**

**C Segment 6 in module, IMODUL**

**NGRND(6,IMODUL) = 0**

**NPREV(6,IMODUL) = 2**

**JPREV1(6,IMODUL) = -3**

**NDCUR1(6,IMODUL) = 1**

**NDPRV1(6,IMODUL) = NODSEG**

**JPREV2(6,IMODUL) = -5**

**NDCUR2(6,IMODUL) = NODSEG**

**NDPRV2(6,IMODUL) = NODSEG**

**NSHAPE(6,IMODUL) = 1**

**R1(6,IMODUL) = RB(IMODUL)**

**Z1(6,IMODUL) = ZB(IMODUL)**

**R2(6,IMODUL) = RA(IMODUL)**

**Z2(6,IMODUL) = ZA(IMODUL)**

**MATTYP(6,IMODUL) = 3**

**THICK(6,IMODUL) = TFWEBS**

**LAYTYP(6,IMODUL) = 5**

**C**

**C Additional two shell segments at the end of the model...**

**IF (IMODUL.EQ.MODULS) THEN**

**NGRND(7,IMODUL) = 1**

**NODGRD(7,IMODUL) = NODSEG**

**NPREV(7,IMODUL) = 1**

**JPREV1(7,IMODUL) = -6**

**NDCUR1(7,IMODUL) = 1**

**NDPRV1(7,IMODUL) = NODSEG**

**NSHAPE(7,IMODUL) = 1**

**R1(7,IMODUL) = RA(IMODUL)**

**Z1(7,IMODUL) = ZA(IMODUL)**

**R2(7,IMODUL) = RSTART + RADIUS + HEIGHT -DSMALO**

**Z2(7,IMODUL) = 0.**

**MATTYP(7,IMODUL) = 2**

**THICK(7,IMODUL) = TFOUTR**

**LAYTYP(7,IMODUL) = 1**

**C**

**NGRND(8,IMODUL) = 1**

**NODGRD(8,IMODUL) = NODSEG**

**NPREV(8,IMODUL) = 1**

**JPREV1(8,IMODUL) = -7**

**NDCUR1(8,IMODUL) = 1**

**NDPRV1(8,IMODUL) = NODSEG**

**NSHAPE(8,IMODUL) = 2**

**SROT(8,IMODUL) = 1.**

**R1(8,IMODUL) = RA(IMODUL)**

**Z1(8,IMODUL) = ZA(IMODUL)**

**R2(8,IMODUL) = RSTART + RCOUTR + ROUTER**

**Z2(8,IMODUL) = 0.**

**RC(8,IMODUL) = RSTART + RCOUTR**

**ZC(8,IMODUL) = 0.**

**MATTYP(8,IMODUL) = 1**

**THICK(8,IMODUL) = TOUTER**

**LAYTYP(8,IMODUL) = 2**

**PFIXED(8,IMODUL) = -PMIDDL(ILOADX)**

**PEIGEN(8,IMODUL) = POUTER(ILOADX)**

**ENDIF**

**110 CONTINUE**

**C**

**DO 115 J = 1,MODULS**

**DO 114 I = 1,8**

**WRITE(IFILE4,'(A,2I3,1P6E12.4)')**

**1 ' J,I,R1(I,J),Z1(I,J),R2(I,J),Z2(I,J),RC(I,J),ZC(I,J)=',**

**1 J,I,R1(I,J),Z1(I,J),R2(I,J),Z2(I,J),RC(I,J),ZC(I,J)**

**114 CONTINUE**

**115 CONTINUE**

**C CALL EXIT**

**C**

**ISEGT = 0**

**DO 150 IMODUL = 1,MODULS**

**DO 120 ISEG = 1,6**

**ISEGT = ISEGT + 1**

**NGRNDJ(ISEGT) = NGRND(ISEG,IMODUL)**

**NODGRJ(ISEGT) = NODGRD(ISEG,IMODUL)**

**NPREVJ(ISEGT) = NPREV(ISEG,IMODUL)**

**JPREVJ(ISEGT,1) = JPREV1(ISEG,IMODUL)**

**NDCURJ(ISEGT,1) = NDCUR1(ISEG,IMODUL)**

**NDPRVJ(ISEGT,1) = NDPRV1(ISEG,IMODUL)**

**JPREVJ(ISEGT,2) = JPREV2(ISEG,IMODUL)**

**NDCURJ(ISEGT,2) = NDCUR2(ISEG,IMODUL)**

**NDPRVJ(ISEGT,2) = NDPRV2(ISEG,IMODUL)**

**NSHPJ(ISEGT) = NSHAPE(ISEG,IMODUL)**

**NODJ(ISEGT) = NODSEG**

**R1J(ISEGT) = R1(ISEG,IMODUL)**

**Z1J(ISEGT) = Z1(ISEG,IMODUL)**

**R2J(ISEGT) = R2(ISEG,IMODUL)**

**Z2J(ISEGT) = Z2(ISEG,IMODUL)**

**MATLJ(ISEGT) = MATTYP(ISEG,IMODUL)**

**THICKJ(ISEGT) = THICK(ISEG,IMODUL)**

**LTYPEJ(ISEGT) = LAYTYP(ISEG,IMODUL)**

**IF (NSHAPE(ISEG,IMODUL).EQ.2) THEN**

**RCJ(ISEGT) = RC(ISEG,IMODUL)**

**ZCJ(ISEGT) = ZC(ISEG,IMODUL)**

**SROTJ(ISEGT) = SROT(ISEG,IMODUL)**

**PFIXJ(ISEGT) = PFIXED(ISEG,IMODUL)**

**PEIGJ(ISEGT) = PEIGEN(ISEG,IMODUL)**

**ENDIF**

**120 CONTINUE**

**IF (IMODUL.EQ.MODULS) THEN**

**DO 130 ISEG = 7,8**

**ISEGT = ISEGT + 1**

**NGRNDJ(ISEGT) = NGRND(ISEG,IMODUL)**

**NODGRJ(ISEGT) = NODGRD(ISEG,IMODUL)**

**NPREVJ(ISEGT) = NPREV(ISEG,IMODUL)**

**JPREVJ(ISEGT,1) = JPREV1(ISEG,IMODUL)**

**NDCURJ(ISEGT,1) = NDCUR1(ISEG,IMODUL)**

**NDPRVJ(ISEGT,1) = NDPRV1(ISEG,IMODUL)**

**JPREVJ(ISEGT,2) = JPREV2(ISEG,IMODUL)**

**NDCURJ(ISEGT,2) = NDCUR2(ISEG,IMODUL)**

**NDPRVJ(ISEGT,2) = NDPRV2(ISEG,IMODUL)**

**NSHPJ(ISEGT) = NSHAPE(ISEG,IMODUL)**

**NODJ(ISEGT) = NODSEG**

**R1J(ISEGT) = R1(ISEG,IMODUL)**

**Z1J(ISEGT) = Z1(ISEG,IMODUL)**

**R2J(ISEGT) = R2(ISEG,IMODUL)**

**Z2J(ISEGT) = Z2(ISEG,IMODUL)**

**MATLJ(ISEGT) = MATTYP(ISEG,IMODUL)**

**THICKJ(ISEGT) = THICK(ISEG,IMODUL)**

**LTYPEJ(ISEGT) = LAYTYP(ISEG,IMODUL)**

**IF (NSHAPE(ISEG,IMODUL).EQ.2) THEN**

**RCJ(ISEGT) = RC(ISEG,IMODUL)**

**ZCJ(ISEGT) = ZC(ISEG,IMODUL)**

**SROTJ(ISEGT) = SROT(ISEG,IMODUL)**

**PFIXJ(ISEGT) = PFIXED(ISEG,IMODUL)**

**PEIGJ(ISEGT) = PEIGEN(ISEG,IMODUL)**

**ENDIF**

**130 CONTINUE**

**ENDIF**

**150 CONTINUE**

**C**

**C Get correct temperature to generate membrane tension in the**

**C axial direction;**

**C**

**C**

**FMODS = MODULS**

**ARCOUT = 2.\*PHIOUT\*ROUTER\*FMODS**

**ARCINR = 2.\*PHIINR\*RINNER\*FMODS**

**SLANT = SQRT((R2J(5)-R1J(5))\*\*2 +(Z2J(5)-Z1J(5))\*\*2)**

**ARCFOT = FLOUTR\*FMODS**

**ARCFIN = FLINNR\*FMODS**

**ARCWEB = 2.\*SLANT\*FMODS**

**ENDFCE = PMIDDL(ILOADX)\*PI\*((RADIUS+HEIGHT)\*\*2 -RADIUS\*\*2)/4.**

**FNU21O = EMOD1(1)\*NU(1)/EMOD2(1)**

**FNU21I = EMOD1(2)\*NU(2)/EMOD2(2)**

**FNU21M = EMOD1(3)\*NU(3)/EMOD2(3)**

**C22OUT = EMOD2(1)\*TOUTER/(1.-NU(1)\*FNU21O)**

**C22INR = EMOD2(1)\*TINNER/(1.-NU(1)\*FNU21O)**

**C22FOT = EMOD2(2)\*TFOUTR/(1.-NU(2)\*FNU21I)**

**C22FIN = EMOD2(2)\*TFINNR/(1.-NU(2)\*FNU21I)**

**C22WEB = EMOD2(3)\*TFWEBS/(1.-NU(3)\*FNU21M)**

**C44FIN = (EMOD1(2)\*TFINNR\*\*3)/(12.\*(1.-NU(2)\*FNU21I))**

**C**

**ARCTOT = ARCOUT\*C22OUT\*ALPHA2(1)**

**1 +ARCINR\*C22INR\*ALPHA2(1)**

**1 +ARCFOT\*C22FOT\*ALPHA2(2)**

**1 +ARCFIN\*C22FIN\*ALPHA2(2)**

**1 +ARCWEB\*C22WEB\*ALPHA2(3)**

**C**

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

**C**

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

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

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

**C**

**C Load set A delta temperature = DELT:**

**ENDF12 = 0.**

**DELT = 0.**

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

**WRITE(IFILE4,'(/,A,1P,6E12.4)')**

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

**ENDF12 = N2DIFF(1)\*ARCFIN +N2DIFF(2)\*ARCFOT +N2DIFF(3)\*ARCFIN**

**1 +N2DIFF(4)\*ARCINR +N2DIFF(5)\*ARCWEB**

**DELT = -ENDF12/ARCTOT**

**ENDIF**

**C**

**C Load set B delta temperature = DELTAT:**

**DELTAT = -ENDFCE/ARCTOT**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**WRITE(IFILE4,'(/,A,1P,2E12.4)')**

**1' delta temps for generating axial resultants, DELTAT,DELT=',**

**1 DELTAT,DELT**

**WRITE(IFILE4,'(A,1P,E12.4)') ' PMIDDL(ILOADX)=',PMIDDL(ILOADX)**

**WRITE(IFILE4,'(A,1P,E12.4)') ' FMODS =',FMODS**

**WRITE(IFILE4,'(A,1P,E12.4)') ' ENDFCE=',ENDFCE**

**WRITE(IFILE4,'(A,1P,E12.4)') ' ARCTOT=',ARCTOT**

**WRITE(IFILE4,'(A,1P,E12.4)') ' ARCOUT=',ARCOUT**

**WRITE(IFILE4,'(A,1P,E12.4)') ' ARCINR=',ARCINR**

**WRITE(IFILE4,'(A,1P,E12.4)') ' ARCFOT=',ARCFOT**

**WRITE(IFILE4,'(A,1P,E12.4)') ' ARCFIN=',ARCFIN**

**WRITE(IFILE4,'(A,1P,E12.4)') ' ARCWEB=',ARCWEB**

**WRITE(IFILE4,'(A,1P,E12.4)') ' SLANT =',SLANT**

**WRITE(IFILE4,'(A,1P,E12.4)') ' C22OUT=',C22OUT**

**WRITE(IFILE4,'(A,1P,E12.4)') ' C22INR=',C22INR**

**WRITE(IFILE4,'(A,1P,E12.4)') ' C22FOT=',C22FOT**

**WRITE(IFILE4,'(A,1P,E12.4)') ' C22FIN=',C22FIN**

**WRITE(IFILE4,'(A,1P,E12.4)') ' C22WEB=',C22WEB**

**WRITE(IFILE4,'(A,1P,E12.4)') ' FNU21O=',FNU21O**

**WRITE(IFILE4,'(A,1P,E12.4)') ' FNU21I=',FNU21I**

**WRITE(IFILE4,'(A,1P,E12.4)') ' FNU21M=',FNU21M**

**C CALL EXIT**

**C**

**MSEGS = ISEGT**

**C**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**C**

**C Next, we generate a valid input data file, \*.ALL, for BIGBOSOR4**

**C**

**C Global input before segment data...**

**C**

**C IF (INDX.EQ.0.OR.INDX.EQ.1) WRITE(IFIL14,'(A,I3,A)')**

**C 1' local buckling, ',MODULS,'-module model (INDIC=0) ixprism'**

**IF (INDX.EQ.0) WRITE(IFIL14,'(A,I3,A)')**

**1' Load B equilib, ',MODULS,'-module model (INDIC=0) ixprism'**

**IF (INDX.EQ.1) WRITE(IFIL14,'(A,I3,A)')**

**1' general buckling, ',MODULS,'-module model (INDIC=1) ixprism'**

**IF (INDX.EQ.2) WRITE(IFIL14,'(A,I3,A)')**

**1' stress components, ',MODULS,'-module model (INDIC=0) ixprism'**

**C IF (INDX.EQ.0.OR.INDX.EQ.1)**

**C 1 WRITE(IFIL14,'(1P,E14.6,A)') LENGTH\*FACLEN,**

**C 1' $ AXIALL = reduced axial length, LENGTH x FACLEN, local buck'**

**IF (INDX.EQ.0.OR.INDX.EQ.1.OR.INDX.EQ.2)**

**1 WRITE(IFIL14,'(1P,E14.6,A)') LENGTH,**

**1' $ AXIALL = axial length of cyl.'**

**IF (INDX.EQ.0.OR.INDX.EQ.1) WRITE(IFIL14,'(I5,A,I3,A)')**

**1 INDIC,' 1, 0, ',MSEGS,' $ INDIC,NPRT,ISTRESS,NSEG'**

**IF (INDX.EQ.2) WRITE(IFIL14,'(I5,A,I3,A)')**

**1 INDIC,' 1, 1, ',MSEGS,' $ INDIC,NPRT,ISTRESS,NSEG'**

**C**

**C Segment data...**

**C**

**C First, provide the input for each of the MSEGS shell segments**

**C**

**DO 200 ISEG = 1,MSEGS**

**I = ISEG**

**WRITE(IFIL14,'(A,4I6)')' H $ Segment number ',I,I,I,I**

**WRITE(IFIL14,'(I4,A,I3,A)') NODJ(ISEG),', 3, ',NSHPJ(ISEG),**

**1 ' $ NMESH,NTYPEH,NSHAPE'**

**WRITE(IFIL14,'(1P,4E14.6,A)')**

**1 R1J(ISEG),Z1J(ISEG),R2J(ISEG),Z2J(iSEG),' $ R1,Z1,R2,Z2'**

**C END JUL 2010**

**IF (NSHPJ(ISEG).EQ.2) THEN**

**WRITE(IFIL14,'(1P,3E14.6,A)') RCJ(ISEG),ZCJ(ISEG),SROTJ(ISEG),**

**1 ' $ RC,ZC,SROT'**

**ENDIF**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**WRITE(IFIL14,'(A,1P,E14.6,A)')' 0, 3, ',0.5\*THICKJ(ISEG),**

**1 ' $ IMP,NTYPEZ,ZVAL'**

**WRITE(IFIL14,'(A)')' N $ do not print r(s), etc.'**

**NRINGS = 0**

**FOUND = 0.**

**LINTYP = 0**

**C**

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

**C**

**IDISAB = 1**

**WRITE(IFIL14,'(I5,1PE14.6,2I4,A)')**

**1 NRINGS,FOUND,LINTYP,IDISAB, ' $ NRINGS,K,LINTYP,IDISAB'**

**C Load Set A...**

**NLTYPE = 3**

**C normal pressure...**

**NPSTAT = 2**

**NLOAD1 = 0**

**NLOAD2 = 0**

**NLOAD3 = 1**

**WRITE(IFIL14,'(5I5,A)')**

**1 NLTYPE,NPSTAT,NLOAD1,NLOAD2,NLOAD3,**

**1 ' $ NLTYPE,NPSTAT,NLOAD(1),NLOAD(2),NLOAD(3)'**

**WRITE(IFIL14,'(1P,2E14.6,A)')**

**1 PFIXJ(ISEG),PFIXJ(ISEG),' $ PN(1),PN(2)'**

**NTYPE = 1**

**ISTA1 = 1**

**ISTA2 = NODJ(ISEG)**

**WRITE(IFIL14,'(3I5,A)')**

**1 NTYPE,ISTA1,ISTA2,' $ NTYPE,IPOINT(1),IPOINT(2)'**

**C**

**C temperature...**

**NTSTAT = 2**

**NTGRAD = 1**

**NLOAD1 = 1**

**NLOAD2 = 0**

**NLOAD3 = 0**

**WRITE(IFIL14,'(5I5,A)')**

**1 NTSTAT,NTGRAD,NLOAD1,NLOAD2,NLOAD3,**

**1 ' $ NTSTAT,NTGRAD,NLOAD(1),NLOAD(2),NLOAD(3)'**

**MATJ = MATLJ(ISEG)**

**WRITE(IFIL14,'(1P,2E14.6,A)')**

**1 DELTAT,DELTAT,' $ T1(1),T1(2)'**

**NTYPE = 1**

**ISTA1 = 1**

**ISTA2 = NODJ(ISEG)**

**WRITE(IFIL14,'(3I5,A)')**

**1 NTYPE,ISTA1,ISTA2,' $ NTYPE,IPOINT(1),IPOINT(2)'**

**C**

**C End of INDX.EQ.0 condition**

**ELSE**

**C Begin INDX.NE.0 condition**

**C**

**IDISAB = 3**

**WRITE(IFIL14,'(I5,1PE14.6,2I4,A)')**

**1 NRINGS,FOUND,LINTYP,IDISAB, ' $ NRINGS,K,LINTYP,IDISAB'**

**C Load set A...**

**NLTYPE = 3**

**C normal pressure...**

**NPSTAT = 2**

**NLOAD1 = 0**

**NLOAD2 = 0**

**NLOAD3 = 1**

**WRITE(IFIL14,'(5I5,A)')**

**1 NLTYPE,NPSTAT,NLOAD1,NLOAD2,NLOAD3,**

**1 ' $ NLTYPE,NPSTAT,NLOAD(1),NLOAD(2),NLOAD(3)'**

**WRITE(IFIL14,'(1P,2E14.6,A)')**

**1 PEIGJ(ISEG),PEIGJ(ISEG),' $ PN(1),PN(2)'**

**NTYPE = 1**

**ISTA1 = 1**

**ISTA2 = NODJ(ISEG)**

**WRITE(IFIL14,'(3I5,A)')**

**1 NTYPE,ISTA1,ISTA2,' $ NTYPE,IPOINT(1),IPOINT(2)'**

**C**

**C temperature...**

**NTSTAT = 2**

**NTGRAD = 1**

**NLOAD1 = 1**

**NLOAD2 = 0**

**NLOAD3 = 0**

**WRITE(IFIL14,'(5I5,A)')**

**1 NTSTAT,NTGRAD,NLOAD1,NLOAD2,NLOAD3,**

**1 ' $ NTSTAT,NTGRAD,NLOAD(1),NLOAD(2),NLOAD(3)'**

**WRITE(IFIL14,'(1P,2E14.6,A)')**

**1 DELT,DELT,' $ T1(1),T1(2)'**

**NTYPE = 1**

**ISTA1 = 1**

**ISTA2 = NODJ(ISEG)**

**WRITE(IFIL14,'(3I5,A)')**

**1 NTYPE,ISTA1,ISTA2,' $ NTYPE,IPOINT(1),IPOINT(2)'**

**C**

**C Load Set B...**

**NLTYPE = 3**

**C normal pressure...**

**NPSTAT = 2**

**NLOAD1 = 0**

**NLOAD2 = 0**

**NLOAD3 = 1**

**WRITE(IFIL14,'(5I5,A)')**

**1 NLTYPE,NPSTAT,NLOAD1,NLOAD2,NLOAD3,**

**1 ' $ NLTYPE,NPSTAT,NLOAD(1),NLOAD(2),NLOAD(3)'**

**WRITE(IFIL14,'(1P,2E14.6,A)')**

**1 PFIXJ(ISEG),PFIXJ(ISEG),' $ PN(1),PN(2)'**

**NTYPE = 1**

**ISTA1 = 1**

**ISTA2 = NODJ(ISEG)**

**WRITE(IFIL14,'(3I5,A)')**

**1 NTYPE,ISTA1,ISTA2,' $ NTYPE,IPOINT(1),IPOINT(2)'**

**C**

**C temperature...**

**NTSTAT = 2**

**NTGRAD = 1**

**NLOAD1 = 1**

**NLOAD2 = 0**

**NLOAD3 = 0**

**WRITE(IFIL14,'(5I5,A)')**

**1 NTSTAT,NTGRAD,NLOAD1,NLOAD2,NLOAD3,**

**1 ' $ NTSTAT,NTGRAD,NLOAD(1),NLOAD(2),NLOAD(3)'**

**MATJ = MATLJ(ISEG)**

**WRITE(IFIL14,'(1P,2E14.6,A)')**

**1 DELTAT,DELTAT,' $ T1(1),T1(2)'**

**NTYPE = 1**

**ISTA1 = 1**

**ISTA2 = NODJ(ISEG)**

**WRITE(IFIL14,'(3I5,A)')**

**1 NTYPE,ISTA1,ISTA2,' $ NTYPE,IPOINT(1),IPOINT(2)'**

**C**

**C End of INDX.NE.0 condition**

**ENDIF**

**C**

**C shell wall construction...**

**NWALL = 4**

**LAYERS = 1**

**WRITE(IFIL14,'(2I3,A)') NWALL,LAYERS,' $ NWALL,NLAYER'**

**WRITE(IFIL14,'(I3,A)') LTYPEJ(ISEG),' $ layer index'**

**IF (ISEG.LE.5) THEN**

**ANGLE2 = 0.**

**WRITE(IFIL14,'(A)')' Y $ is this a new layer type?'**

**WRITE(IFIL14,'(1P2E14.6,I3,A)') THICKJ(ISEG),**

**1 ANGLE2,MATLJ(ISEG),' $ thickness,angle,material'**

**ELSE**

**WRITE(IFIL14,'(A)')' N $ is this a new layer type?'**

**ENDIF**

**NEWMAT = 0**

**IF (ISEG.LE.5) THEN**

**IF (ISEG.LE.2.OR.ISEG.EQ.5) NEWMAT = 1**

**ENDIF**

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

**IF (ISEG.LE.5)**

**1 WRITE(IFIL14,'(A)')' N $ Is this material new?'**

**ELSE**

**IMATL = MATLJ(ISEG)**

**WRITE(IFIL14,'(A)')' Y $ Is this material new?'**

**WRITE(IFIL14,'(1P4E14.6,A)') EMOD1(IMATL),EMOD2(IMATL),**

**1 G12(IMATL),NU(IMATL),' $ E1,E2,G12,NU'**

**WRITE(IFIL14,'(1P4E14.6,A)') ALPHA1(IMATL),ALPHA2(IMATL),**

**1 TEMPER(IMATL),DENSTY(IMATL),' $ A1,A2,TEMPTUR,DENS'**

**IF (IMATL.EQ.1) THEN**

**WRITE(IFIL14,'(1P3E14.6,A)') STRM1A(ILOADX,1),**

**1 STRM1A(ILOADX,2),STRM1A(ILOADX,3),' $ S(1),S(2),S(3)'**

**WRITE(IFIL14,'(1P2E14.6,A)') STRM1A(ILOADX,4),**

**1 STRM1A(ILOADX,5),' $ S(4),S(5)'**

**ENDIF**

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

**WRITE(IFIL14,'(1P3E14.6,A)') STRM2A(ILOADX,1),**

**1 STRM2A(ILOADX,2),STRM2A(ILOADX,3),' $ S(1),S(2),S(3)'**

**WRITE(IFIL14,'(1P2E14.6,A)') STRM2A(ILOADX,4),**

**1 STRM2A(ILOADX,5),' $ S(4),S(5)'**

**ENDIF**

**IF (IMATL.EQ.3) THEN**

**WRITE(IFIL14,'(1P3E14.6,A)') STRM3A(ILOADX,1),**

**1 STRM3A(ILOADX,2),STRM3A(ILOADX,3),' $ S(1),S(2),S(3)'**

**WRITE(IFIL14,'(1P2E14.6,A)') STRM3A(ILOADX,4),**

**1 STRM3A(ILOADX,5),' $ S(4),S(5)'**

**ENDIF**

**ENDIF**

**WRITE(IFIL14,'(A)')' 0 $ no additional smeared stiffeners'**

**WRITE(IFIL14,'(A)')' Y $ do you want output for all nodes?'**

**WRITE(IFIL14,'(A)')' N $ do you want to print out Cij?'**

**WRITE(IFIL14,'(A)')' N $ do you want to print out loads?'**

**C**

**C The old input for NWALL = 5 follows. (It is commented out).**

**C WRITE(IFIL14,'(4I5,A)')**

**C 1 NWALL,LAYERS,NRS,NTYPET,' $ NWALL,LAYERS,NRS,NTYPET'**

**C**

**C MATJ = MATLJ(ISEG)**

**C WRITE(IFIL14,'(1P,3E14.6,A)')**

**C 1 THICKJ(ISEG),G12(MATJ),EMOD1(MATJ),' $ T,G,EX'**

**C WRITE(IFIL14,'(1P,3E14.6,A)')**

**C 1 EMOD2(MATJ),NU(MATJ),DENSTY(MATJ),' $ EY,UXY,SM'**

**C WRITE(IFIL14,'(1P,2E14.6,A)')**

**C 1 ALPHA1(MATJ),ALPHA2(MATJ),' $ ALPHA1,ALPHA2'**

**C**

**C WRITE(IFIL14,'(A)')' N $ do you want to print C(i,j)?'**

**C WRITE(IFIL14,'(A)')' N $ do you want to print distrib.loads?'**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**C**

**200 CONTINUE**

**C End of loop over the number of shell segments in the model.**

**C**

**C Start the global data...**

**C**

**NLAST = 0**

**N0B = 1**

**NMINB = 1**

**NMAXB = 1**

**INCRB = 1**

**NVEC = 1**

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

**IF (ITRY.EQ.1) THEN**

**PMULT = 0.1**

**DPMULT= 0.1**

**TEMP = 0.1**

**DTEMP = 0.1**

**NUMSTP = 10**

**ELSE**

**IF (ITRY.EQ.3) THEN**

**PMULT = 0.02**

**DPMULT= 0.02**

**TEMP = 0.02**

**DTEMP = 0.02**

**NUMSTP = 50**

**ELSE**

**PMULT = 1.0**

**DPMULT= 1.0**

**TEMP = 1.0**

**DTEMP = 1.0**

**NUMSTP = 1**

**ENDIF**

**ENDIF**

**ELSE**

**PMULT = 0.**

**DPMULT = 1.0**

**TEMP = 1.**

**DTEMP = 0.**

**NUMSTP = 2**

**ENDIF**

**OMEGA = 0.**

**DOMEGA = 0.**

**OMEGA = 0.**

**DOMEGA = 0.**

**WRITE(IFIL14,'(I5,A)')**

**1 NLAST,' $ NLAST'**

**WRITE(IFIL14,'(A)')' N $ are there any expanded plots?'**

**IF (INDIC.EQ.1) WRITE(IFIL14,'(5I5,A)')**

**1 N0B,NMINB,NMAXB,INCRB,NVEC,' $ N0B,NMINB,NMAXB,INCRB,NVEC'**

**WRITE(IFIL14,'(1P,4E14.6,A)')**

**1 PMULT,DPMULT,TEMP,DTEMP,' $ P,DP,TEMP,DTEMP'**

**IF (INDIC.EQ.0) WRITE(IFIL14,'(I5,A)')**

**1 NUMSTP,' $ number of load steps'**

**WRITE(IFIL14,'(1P,2E14.6,A)')**

**1 OMEGA,DOMEGA,' $ OMEGA,DOMEGA'**

**C**

**C Constraint conditions follow...**

**C**

**WRITE(IFIL14,'(I5,A)')**

**1 MSEGS,' $ How many segments in the structure?'**

**C**

**DO 300 I = 1,MSEGS**

**C**

**WRITE(IFIL14,'(A,4I6)')**

**1 ' H $ CONSTRAINT CONDITIONS FOR SEGMENT ',I,I,I,I**

**WRITE(IFIL14,'(A)')' 0 $ number of poles'**

**WRITE(IFIL14,'(I3,A)') NGRNDJ(I),' $ connect to ground'**

**IF (NGRNDJ(I).GT.0) THEN**

**NGRNDI = NGRNDJ(I)**

**DO 250 J = 1,NGRNDI**

**WRITE(IFIL14,'(I3,A)') NODGRJ(I),' $ node to ground'**

**IF (I.EQ.1.OR.I.EQ.2.OR.I.EQ.3) THEN**

**IUSTAR = 0**

**IVSTAR = 0**

**IWSTAR = 1**

**ICHI = 1**

**ELSE**

**IUSTAR = 1**

**IVSTAR = 0**

**IWSTAR = 0**

**ICHI = 1**

**ENDIF**

**WRITE(IFIL14,'(4I5,A)')**

**1 IUSTAR,IVSTAR,IWSTAR,ICHI,**

**1 ' $ IUSTAR,IVSTAR,IWSTAR,ICHI'**

**D1 = 0.**

**D2 = 0.**

**WRITE(IFIL14,'(1P,2E14.6,A)') D1,D2,' $ D1,D2'**

**WRITE(IFIL14,'(A)')' Y $ is constraint same for buck.?'**

**250 CONTINUE**

**ENDIF**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**C**

**IF (I.EQ.1.OR.I.EQ.3) THEN**

**WRITE(IFIL14,'(A)')' N $ joined to previous segments?'**

**ELSE**

**WRITE(IFIL14,'(A)')' Y $ joined to previous segments?'**

**WRITE(IFIL14,'(I3,A)') NPREVJ(I),' $ connects to prev.segs'**

**NPREVI = NPREVJ(I)**

**DO 220 J = 1,NPREVI**

**WRITE(IFIL14,'(I3,A)') NDCURJ(I,J),' $ node current seg'**

**WRITE(IFIL14,'(I3,A)')**

**1 I + JPREVJ(I,J),' $ prev.segment no.'**

**WRITE(IFIL14,'(I3,A)') NDPRVJ(I,J),' $ node in prev.seg.'**

**WRITE(IFIL14,'(A)')' 1, 1, 1, 0 $ IU,IV,IW,ICHI'**

**WRITE(IFIL14,'(A)')' 0., 0. $ D1,D2'**

**WRITE(IFIL14,'(A)')' Y $ is constraint same for buck.?'**

**C23456789012345678901234567890123456789012345678901234567890123456789012**

**220 CONTINUE**

**ENDIF**

**C**

**300 CONTINUE**

**C**

**WRITE(IFIL14,'(A)')' N $ are rigid body motions possible?'**

**C**

**DO 410 ISEG = 1,MSEGS**

**WRITE(IFIL14,'(A)')' Y $ do you want to list seg. output?'**

**410 CONTINUE**

**WRITE(IFIL14,'(A)')' Y $ do you want to list ring forces?'**

**C**

**RETURN**

**END**

**=========================================================================**