Appendix A

Summary of Grisham's equations

A summary of the equations in Grisham's paper is given below. Note that for all equations, the buckling stress is positive.

First, the modified shear buckling allowable stress is evaluated:

$$\tau_{xy_{cr}} = \tau_{xy_{cr_0}} \sqrt{1 - \frac{\sigma_x}{\sigma_{x_{cr_0}}} - \frac{\sigma_y}{\sigma_{y_{cr_0}}}} \tag{A.1}$$

If the linear finite element solution indicates a tension-tension stress state, else for tension-compression and compression-compression the following equation is used:

$$\tau_{xy_{cr}}^2 + \frac{\tau_{xy_{cr_0}}^2(\sigma_x\sigma_{y_{cr_0}} + \sigma_y\sigma_{x_{cr_0}})}{|\tau_{xy}|\sigma_{x_{cr_0}}\sigma_{y_{cr_0}}}\tau_{xy_{cr}} - \tau_{xy_{cr_0}}^2 = 0 \tag{A.2}$$

Once this is done, the modified membrane buckling stresses are determined:

$$\frac{|\tau_{xy}|}{|\tau_{xy_{cr}}|} = \frac{\sigma_x}{\sigma_{x_{cr}}} = \frac{\sigma_y}{\sigma_{y_{cr}}} \tag{A.3}$$

Once these have been calculated an interaction equation is used to determine if the web buckles:

$$\frac{\sigma_{x_{cr}}}{\sigma_{x_{cr_0}}} + \frac{\sigma_{y_{cr}}}{\sigma_{y_{cr_0}}} + (\frac{\tau_{xy_{cr}}}{\tau_{xy_{cr_0}}})^2 = 1 \tag{A.4}$$

If the web buckles, the diagonal tension factor, angle and stresses in the x- and y-directions can be calculated:

$$k = \tanh(0.5 \log \frac{\tau_{xy}}{\tau_{xy_{cr}}}) \tag{A.5}$$

The diagonal tension angle is calculated using the following equations and a series of successive approximations:

$$\tan^2 \alpha = \frac{\epsilon - \epsilon_f}{\epsilon - \epsilon_u} \tag{A.6}$$

where:

$$\epsilon_f = \frac{H_1}{\tan \alpha} \tag{A.7}$$

$$\epsilon_u = H_2 \tan \alpha \tag{A.8}$$

$$\epsilon = \frac{H_3}{\sin 2\alpha} + H_4 \sin 2\alpha \tag{A.9}$$

$$H_1 = \frac{-k|\tau_{xy}|}{E_f(\frac{A_f}{L_y t} + 0.5(1-k))} \tag{A.10}$$

$$H_2 = \frac{-k|\tau_{xy}|}{E_u(\frac{A_u}{L_x t} + 0.5(1-k))} \tag{A.11}$$

$$H_3 = \frac{2k|\tau_{xy}|}{E_w} \tag{A.12}$$

$$H_4 = (1 - k + \mu - \mu k)(\frac{|\tau_{xy}|}{E_w}) \tag{A.13}$$

A_f and A_u are half of the summed stiffener areas in the x- and y-directions respectively.

The diagonal tension stress to be developed in the plate following shear buckling is:

$$\sigma_{x_{DT}} = k|\tau_{xy}| \cot \alpha \tag{A.14}$$

$$\sigma_{y_{DT}} = k|\tau_{xy}| \tan \alpha \tag{A.15}$$

The shear strain of the plate in its post-buckled state is:

$$\gamma_{xy} = [\frac{1-k}{G_w} + \frac{k}{G_{pdt}}]\tau_{xy} \tag{A.16}$$

where:

$$\frac{1}{G_{pdt}} = \frac{1}{2G_w(1+\mu)}\{\frac{4}{\sin^2 2\alpha} + \frac{E_w L_x t \tan^2\alpha}{E_u(A_u + 0.5L_x t(1-k))} + \frac{E_w L_y t \cot^2\alpha}{E_f(2A_f + 0.5L_y t(1-k))}\} \tag{A.17}$$

Defining:

$$\frac{1}{G_{pdt}} = \frac{1}{G_w}\theta \tag{A.18}$$

then:

$$\gamma_{xy} = (\frac{1}{G_w} + \frac{k(\theta-1)}{G_w})\tau_{xy} \tag{A.19}$$

where the second term is the shear deformation of the plate in its post-buckled state.

Therefore, the set of pre-strains required to induce the diagonal tension stresses in the plate is:

$$\epsilon_{x_{DT}} = -\frac{1}{E_w}(\frac{E_w L_y t}{A_f E_f} + 1)(\sigma_{x_{DT}} - \mu\sigma_{y_{DT}}) \tag{A.20}$$

$$\epsilon_{y_{DT}} = -\frac{1}{E_w}(\frac{E_w L_x t}{A_u E_u} + 1)(\sigma_{y_{DT}} - \mu\sigma_{x_{DT}}) \tag{A.21}$$

$$\gamma_{xy_{DT}} = \frac{k(\theta-1)}{G_w})\tau_{xy} \tag{A.22}$$

The compressive stresses in the buckled plate are:

$$\sigma_{x_c} = \frac{C_2C_3[\sigma_x - \sigma_{x_{cr}}]L + \beta_x\mu C_4[\sigma_y - \sigma_{y_{cr}}]L}{C_1C_2} \tag{A.23}$$

$$\sigma_{y_c} = \frac{C_1C_4[\sigma_y - \sigma_{y_{cr}}]L + \beta_y\mu C_3[\sigma_x - \sigma_{x_{cr}}]L}{C_1C_2} \tag{A.24}$$

where:

L is a factor to control the rate at which compressive buckling is incorporated and has a value between '0' and '1'.

$$C_1 = 1 + \frac{L_y t E_w \beta_x}{A_f E_f} \tag{A.25}$$

$$C_2 = 1 + \frac{L_x t E_w \beta_y}{A_u E_u} \tag{A.26}$$

$$C_3 = 1 - \beta_x \tag{A.27}$$

$$C_4 = 1 - \beta_y \tag{A.28}$$

and

$$\beta_x = \frac{E_{w_x}}{E_w} \tag{A.29}$$

$$\beta_y = \frac{E_{w_y}}{E_w} \tag{A.30}$$

The pre-strains corresponding to these changes in stress due to compression buckling are:

$$\epsilon_{x_c} = -\frac{1}{E_w}(1 + \frac{L_y t E_w}{A_f E_f})(\sigma_x - \mu\sigma_y)|\beta_x + \beta_x\beta_y - 1| \tag{A.31}$$

$$\epsilon_{y_c} = -\frac{1}{E_w}(1 + \frac{L_x t E_w}{A_u E_u})(\sigma_y - \mu\sigma_x)|\beta_y + \beta_x\beta_y - 1| \tag{A.32}$$

$$\gamma_{xy_c} = 0.0 \tag{A.33}$$

The total change in strain for the first iteration is the sum of the diagonal tension strain and the compressive strain:

$$\triangle\epsilon_x = \epsilon_{x_{DT}} + \epsilon_{x_c} \tag{A.34}$$

$$\triangle\epsilon_y = \epsilon_{y_{DT}} + \epsilon_{y_c} \tag{A.35}$$

$$\triangle\gamma_{xy} = \frac{k(\theta - 1)}{G_w})\tau_{xy} \tag{A.36}$$

These strains then become the pre-strains in the finite element model, for the next iteration. For the second and succeeding iterations, all the above equations are used. The diagonal tension stress must first be removed from the finite element stress though. The total strain for the n-th iteration is:

$$\epsilon^n_{x_{total}} = \epsilon^{n-1}_x + \triangle\epsilon^n_x \tag{A.37}$$

$$\epsilon^n_{y_{total}} = \epsilon^{n-1}_y + \triangle\epsilon^n_y \tag{A.38}$$

$$\gamma_{xy_{total}} = \frac{k^n(\theta^n - 1)}{G_w}\tau^n_{xy} \tag{A.39}$$

Appendix B

Source code of the software developed

The Grisham Algorithm was implemented by coding the procedure in FORTRAN 77. The program consists of a main routine and numerous subprograms, each fulfilling a different task. An additional postprocessing program was also coded to extract data from the linear finite element analysis results in the ABAQUS® environment which could then be read by the main routine.

The code generates a flat structure of np_x by np_y panels in the x- and y- directions respectively as specified by the user. Each panel has stiffener members around its perimeter which may have totally different cross-sectional areas for each member. Once the number of elements per flange and the elements per upright are chosen, they remain the same for all flanges and all uprights.

The program also makes provision for all possible configurations of boundary conditions for the web buckling critical values (all sides simply supported; all sides fixed; 2 horizontal sides fixed and 2 vertical sides simply supported; 2 horizontal sides simply supported and 2 vertical sides fixed).

The web can be modelled using either shell or membrane elements. The flanges and uprights can be modelled using beam or truss elements. Elements can be first or second order. Only buckling of the web is considered, buckling of the uprights is not taken into account.

```
C
C     VARIABLE DEFINITIONS: (DOUBLE PRECISION)
C
C     ALFA      Diagonal tension angle;  positive when measured
C               from the local x- axis using the right hand rule
C     ALFA11    Expansion coefficient in the x- direction
C     ALFA22    Expansion coefficient in the y- direction
C     ALFA12    Expansion coefficient - shear
C     AX        Half the summed stiffener areas in the x- direction
C               of each panel
C     AY        Half the summed stiffener areas in the y- direction
C               of each panel
C----------------------------------------------------------------------
C     B         Shortest side of plate (minimum of LX or LY)
C     BETAX/Y   Parameter used to indicate whether pre- or post-buckling
C               modulus of elasticity is applicable in the x- or y-
C               direction of the plate/sheet
C     BBETAX/Y Coefficients used in incremental change in strain equations
C               (second and succeeding iterations) to assist convergence of
C               previously compressively buckled plates that have become
C               unbuckled
C     BNF1      Beam nodal force - component 1
C     BNF2      Beam nodal force - component 2
C     BNF3      Beam nodal force - component 3
C     BUCK      The interaction equation used for bi-axial compression
C               and shear buckling
C----------------------------------------------------------------------
C     C1-C4     Variables used in calculating the compressive stresses in
C               a plate
C     CONV1X    Convergence requirement 1
C     CONV1Y    Convergence requirement 1
C     CONV2X    Convergence requirement 2
C     CONV2Y    Convergence requirement 2
C----------------------------------------------------------------------
C     DFX12     Difference in total nodal force between points 1 and 2
C               of the panel in the x-direction (normal to side)
C     DFX23     Difference in total nodal force between points 2 and 3
C               of the panel in the x-direction (normal to side)
C     DFX34     Difference in total nodal force between points 3 and 4
C               of the panel in the x-direction (normal to side)
C     DFX41     Difference in total nodal force between points 4 and 1
C               of the panel in the x-direction (normal to side)
C     DFY12     Difference in total nodal force between points 1 and 2
C               of the panel, in the y-direction (// to side)
C     DFY23     Difference in total nodal force between points 2 and 3
```

```
C               of the panel, in the y-direction (// to side)
C      DFY34    Difference in total nodal force between points 3 and 4
C               of the panel, in the y-direction (// to side)
C      DFY41    Difference in total nodal force between points 4 and 1
C               of the panel, in the y-direction (// to side)
C----------------------------------------------------------------------
C      ENODE    Eccentricity of uprights !! (not flanges) - node to node
C      ECC      Eccentricity of uprights - centroid to centroid
C      EP       Modulus of Elasticity of plate
C      EQN1X    Equation used in first test for BETAs
C      EQN1Y    Equation used in first test for BETAs
C      EQN2X    Equation used in second test for BETAs
C      EQN2Y    Equation used in second test for BETAs
C      EQN3X    Equation used in third test for BETAs
C      EQN3Y    Equation used in third test for BETAs
C      ERR      % error made while calculating successive approximations of the
C               diagonal tension angle (ALFA)
C      ES       Minimum % error allowed in calculating root of
C               equation (subroutine)
C      EXB      Modulus of Elasticity of beam in x- direction
C      EXC      Strain in the x-direction due to compressive loading
C               only
C      EXDT     Strain in the x- direction due to diagonal tension
C      EXP      Effective Modulus of Elasticity of buckled plate in x-
C               direction
C      EXT      Total strain in the x- direction [compressive loading +
C               diagonal tension]
C      EXYDT    Shear strain due to diagonal tension
C      EXYT     Total shear strain
C      EYB      Modulus of Elasticity of beam in y-direction
C      EYC      Strain in the y- direction due to compressive loading
C               only
C      EYDT     Strain in the y- direction due to diagonal tension
C      EYP      Effective Modulus of Elasticity of buckled plate in y-
C               direction
C      EYT      Total strain in the y- direction[compressive loading +
C               diagonal tension]
C---------------------------------------------------------------------
C      FLA      Flange T-section dimension
C      FLB      Flange T-section dimension
C      FLT1     Flange T-section dimension
C      FLT2     Flange T-section dimension
C----------------------------------------------------------------------
C      GP       Shear modulus of plate
C----------------------------------------------------------------------
```

```
C     H1-H4     Variables used to calculate the diagonal tension angle in
C               a plate
C-----------------------------------------------------------------------
C     IXX       Moment of Inertia of upright about an axis through its own
C               centroid and parallel to the web plane
C-----------------------------------------------------------------------
C     K         Diagonal tension factor
C     KS        Shear buckling coefficient for a flat plate
C     KX        Buckling coefficient in x-direction for a flat plate
C     KY        Buckling coefficient in y-direction for a flat plate
C-----------------------------------------------------------------------
C     L         Factor between 0.0 and 1.0 to control the rate at which
C               buckling is incorporated in the solution
C     L1        if LX >= LY then L1 = LY and L2 = LX
C     L2        if LY > LX then L1 = LX and L2 = Ly
C     L12       Plate side length; between corner nodes 1 and 2
C     L23       Plate side length; between corner nodes 2 and 3
C     L34       Plate side length; between corner nodes 3 and 4
C     L41       Plate side length; between corner nodes 4 and 1
C     LPNX      Length between nodes in the x-direction
C     LPNY      Length between nodes in the y-direction
C     LTOTX     Total length of model in x-direction
C     LTOTY     Total length of model in y-direction
C     LX        Length of plate in x-direction
C     LY        Length of plate in y-direction
C------------------------------------------------------------------------
C     NCX       Nodal x-coordinate generated within program
C     NCY       Nodal y-coordinate generated within program
C     NCZ       Nodal z-coordinate generated within program
C     NF1T      Total force at node - component 1 read from ABAQUS output
C     NF2T      Total force at node - component 2 read from ABAQUS output
C     NF3T      Total force at node - component 3 read from ABAQUS output
C------------------------------------------------------------------------
C     POISS     Poisson's ratio
C------------------------------------------------------------------------
C     RHO       Radius of gyration of the upright area
C------------------------------------------------------------------------
C     SNF1      Shell nodal force - component 1
C     SNF2      Shell nodal force - component 2
C     SNF3      Shell nodal force - component 3
C     SX        Average panel normal stress in the x-direction
C     SX12      Normal stress to plate edge between nodes 1 and 2
C     SX23      Normal stress to plate edge between nodes 2 and 3
C     SX34      Normal stress to plate edge between nodes 3 and 4
C     SX41      Normal stress to plate edge between nodes 4 and 1
```

```
C     SXC      Total compressive stress acting on plate in x-direction
C     SXCR     Modified buckling allowable stress in x- direction
C     SXCR0    Critical buckling stress in x-direction (geometry only)
C     SXDT     Diagonal tension stress in plate following buckling
C     SXEFF    Effective (resultant) tensile/compressive stress applied to
C              panel/sheet calculated from panel/sheet nodal
C              forces [x-direction]
C     SXY      Average panel shear stress
C     SXY12    Shear stress alongside plate edge between nodes 1 and 2
C     SXY23    Shear stress alongside plate edge between nodes 2 and 3
C     SXY34    Shear stress alongside plate edge between nodes 3 and 4
C     SXY41    Shear stress alongside plate edge between nodes 4 and 1
C     SXYEFF   Effective (resultant) shear stress applied to panel/sheet
C              calculated form panel/sheet nodal forces [same value on each
C              of the 4 sides]
C     SXYCR    Modified buckling shear allowable stress
C     SXYCR0   Critical shear buckling stress (geometry only)
C     SY       Average panel normal stress in the y-direction
C     SYC      Total compressive stress acting on plate in y-direction
C     SYCR     Modified buckling allowable stress in y- direction
C     SYCR0    Critical buckling stress in y-direction (geometry only)
C     SYDT     Diagonal tension stress in plate following buckling
C     SYEFF    Effective (resultant) tensile/compressive stress applied to
C              panel/sheet calculated from panel/sheet nodal
C              forces [x-direction]
C-----------------------------------------------------------------------
C     T        Plate/sheet thickness
C     TDB      Buckling equation rounded to four decimal places value
C     THETA    Diagonal tension angle
C-----------------------------------------------------------------------
C     UPRA     Angle cross-section dimension
C     UPRT     Angle cross-section thickness dimension
C     UPRY     Angle cross-section dimension
C-----------------------------------------------------------------------
C     XALFA    Guestimate of diagonal tension angle in method of successive
C              approximations to determine ALFA
C     XN       The root of the equation (subroutine)
C     XR1      Initial guess of root of equation (subroutine)
C     XR2      Initial guess of root of equation (subroutine)
C-----------------------------------------------------------------------
C     YCENT    Centroid position of angled upright
C-----------------------------------------------------------------------
C
C
C     VARIABLE DEFINITIONS: (INTEGERS)
```

```
C
C     BENF     Beam element node force array used with node forces
C     BNNNF    Beam node number for node force data
C     BTOT     Total number of panels that are not buckled at the end
C              of the analysis
C-----------------------------------------------------------------------
C     C2O      Constant that = 2 when second order elements are used
C-----------------------------------------------------------------------
C     ELBH     Horizontal beam element set
C     ELBHN    Horizontal beam element number
C     ELBV     Vertical beam element set
C     ELBVN    Vertical beam element number
C     ELS      Shell element set
C     ELSN     Shell element number
C-----------------------------------------------------------------------
C     FN       File number for each panel
C     FN       FIle number for each panel
C-----------------------------------------------------------------------
C     IC1,2..  Counters
C     ICB      Beam element counter
C     ICBH     Horizontal beam element counter
C     ICBV     Vertical beam element counter
C     ICS      Shell element counter
C     IM       Maximum number of iterations allowed to determine root of
C              equation (subroutine)
C     ITN      Iteration number
C-----------------------------------------------------------------------
C     K1,K2..  Constants
C-----------------------------------------------------------------------
C     NALL     All the nodes in the linear FEA model
C     NBEPP    Total number of beam elements associated with each panel
C     NEX      Number of elements per panel/sheet along the x-axis
C     NEY      Number of elements per panel/sheet along the y-axis
C     NFLS     Number of flange element sets
C     NI       Same as ITER
C     NINCY    Node number increment per row along the y-axis
C     NIPPX    Node increment per panel/sheet in the x-direction
C     NIPPY    Node increment per panel/sheet in the y-direction
C     NNXMAX   Maximum node number value in the x-direction
C     NNYMAX   Maximum node number value in the y-direction
C     NONX     Total number of nodes along the x-axis
C     NONY     Total number of nodes along the y-axis
C     NP       Number of panels in structure
C     NPX      Number of panels/sheets along the x-axis
C     NPY      Number of panels/sheets along the y-axis\
```

```
C     NRBNF     Number of records in the beam element node forces file
C     NSEPP     Total number of shell elements associated with each panel
C     NTOTAL    Total number of nodes generated
C-----------------------------------------------------------------------
C     PCNN      Panel corner node number
C     PNLBE     Panel beam elements (panel number,panel beam element
C               number - random)
C     PNLNN     Panel corner node numbers [4] (panel number,panel corner node
C               number - clockwise; must be clockwise and start at 0,0,0
C               so that stresses in each direction can be calculated
C               correctly)
C     PNLSE     Panel shell elements (panel number,panel shell element
C               number - random)
C-----------------------------------------------------------------------
C     SHENF     Shell element node force array used with node forces
C     SHNNNF    Shell node number for node force data
C-----------------------------------------------------------------------
C     YINC      Node number increment per row along the y-axis;
C               = relevant NINCY value
C-----------------------------------------------------------------------
C
C***********************************************************************
C               VARIABLE DEFINITIONS:  (CHARACTERS)
C
C     COMMA     A comma
C     UPRE      Upright eccentricity presence (=NOE or ECC)
C
C***********************************************************************
C***********************************************************************
C
C                MAIN PROGRAM
C
C***********************************************************************
C***********************************************************************
C
C
      DOUBLE PRECISION ALFA,
     +AX,AY,
     +B,BETAX(301),BETAY(301),BBETAX,BBETAY,
     +EP,ERR,EXB,EXP,EYB,EYP,ECC(301),ENODE(301),
     +EQN1X,EQN1Y,EQN2X,EQN2Y,EQN3X,EQN3Y,
     +FLA(301),FLB(301),FLT1(301),FLT2(301),
     +GP,
     +H1,H2,H3,H4,
     +IXX(301),
```

```
     +K,KS,KX,KY,
     +L,L1,L2,LPNX,LPNY,LTOTX,LTOTY,LX,LY,
     +POISS,PI,
     +RHO(301),
     +SXCR0,SYCR0,SXYCR0,
     +T(301),THETA,TDB,
     +UPRA(301),UPRT(301),UPRY(301),
     +XALFA,
     +YCENT(301),
     +SXAV,SYAV,SXYAV,SX,SY,SXY,BB,SXYCR1,SXYCR2,SXYCR,
     +SXCR,SYCR,SXCRI(301),SYCRI(301),SXMOD,SYMOD,
     +BUCK,EPSX,EPSY,EPS,
     +SXDT(301),SYDT(301),EXDT,EYDT,EXYDT,C1,C2,C3,C4,
     +SXC,SYC,EXC,EYC,EXT(301),EYT(301),EXYT,
     +SXPS(301),SYPS(301),SXYPS(301),
     +SXDTI(301),SYDTI(301),SXDTII(301),SYDTII(301),
     +NCX(1000),NCY(1000),NCZ(1000),
     +S11(1000),S22(1000),S12(1000),
     +SXAVE(301),SYAVE(301),SXYAVE(301),
     +SXAVP(301),SYAVP(301),SXYAVP(301),
     +SXTE(1000),SYTE(1000),SXYTE(1000),
     +SXTP(1000),SYTP(1000),SXYTP(1000),
     +K1X,K1Y,K1XP,K1YP,K2X,K2Y,K2XP,K2YP,K1XN(301),K1YN(301),
     +K1XNN(301),K1YNN(301),K2XN(301),K2YN(301),K1XPP,K1YPP,
     +K2XPP,K2YPP,K2XNN(301),K2YNN(301),K2XPPP,K2YPPP,
     +K1XVAL(301),K1YVAL(301),
     +K1XNC(301),K1YNC(301),PVAL,
     +UPRRAD(301),FLRAD(301),AFL(301),AUPR(301),AUPRE(301)
C
      INTEGER BHTOT,BVTOT,C20,
     +CONA(301),CONB(301),CONC(301),COND(301),
     +BCONA(301),BCONB(301),
     +CTOT,BCTOT,
     +FN,FN2,
     +ITN,
     +NBEPP,NP,NFLS,NRBNF,NINCY,
     +NSEPP,NUPRS,
     +SEC,
     +TSTART,TSTOP,
     +YINC,
     +ELE(1000),
     +ELBH(1000,1000),ELBHN(1000),ELBV(1000,1000),ELBVN(1000),
     +ELSN(1000),
     +IPT(1000),SPT(1000),
     +NALL(1000),
```

```
     +PCNN(1000),ELS(1000,1000),
     +PNLNN(1000,6),PNLSE(1000,20),PNLBE(1000,20),
     +K1XID(301),K1YID(301),
     +BPL(301),BTOT,BPID(301),
     +CONAT,CONBT,CONCT,CONDT,
     +IBCMAX,IFAC
C
      CHARACTER*(1)COMMA
      CHARACTER*(3)UPRE
C
      OPEN(10,FILE='bukl.dat')
      OPEN(21,FILE='conv1.dat')
      OPEN(12,FILE='inputdata.dat')
      OPEN(13,FILE='conv2.dat')
      OPEN(31,FILE='pnl1-x.dat')
      OPEN(32,FILE='pnl2-x.dat')
      OPEN(33,FILE='pnl3-x.dat')
      OPEN(34,FILE='pnl4-x.dat')
      OPEN(35,FILE='pnl5-x.dat')
      OPEN(36,FILE='pnl6-x.dat')
      OPEN(41,FILE='pnl1-y.dat')
      OPEN(42,FILE='pnl2-y.dat')
      OPEN(43,FILE='pnl3-y.dat')
      OPEN(44,FILE='pnl4-y.dat')
      OPEN(45,FILE='pnl5-y.dat')
      OPEN(46,FILE='pnl6-y.dat')
C
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     INPUT DATA:
C
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C
      NPX=6
      NPY=1
      NEX=3
      NEY=3
      LX=0.254
      LY=0.7254
      NP=NPX*NPY
C
      DO 63 I=1,NP+1
       UPRA(I)=0.0254
       UPRT(I)=0.003175
```

```
63    CONTINUE
C
      UPRE='ECC'
C
      DO 64 J=1,NP
       T(J)=0.000635
       FLRAD(J)=0.00881
       FLRAD(NP+J)=0.01177
64    CONTINUE
C
      EXB=71.0E9
      EYB=71.0E9
      EP=72.4E9
      GP=26.92E9
      POISS=0.3
      C20=2
C
      COMMA=','
      PVAL=0.0
C
      IBCMAX=60
      PI=3.14159265359
C
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     PROGRAM STARTS
C
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      CALL TIME(SEC)
        TSTART=SEC
        WRITE(10,10)TSTART
10      FORMAT('TSTART = ',I20)
C
C            geometric calculations for structure
C
      LTOTX=LX*NPX
      LTOTY=LY*NPY
      NNXMAX=C20*NPX*NEX+1
      NONX=NNXMAX
      NONY=C20*NPY*NEY+1
      NIPPX=C20*NEX
      NBEPP=2*NEX+2*NEY
      NSEPP=NEX*NEY
      LPNX=LTOTX/(NONX-1)
```

```
      LPNY=LTOTY/(NONY-1)
      NINCY=NNXMAX
      NNYMAX=C20*NPY*NEY+1
      NTOTAL=NNXMAX*NNYMAX
      NIPPY=C20*NEY*NINCY
C
      DO 67 I=1,NPX*NPY+NPX
       AFL(I)=PI*FLRAD(I)**2
67    CONTINUE
C
C             Geometric properties of uprights calculated
C
      DO 14 I=1,NPX+1
       AUPR(I)=(UPRA(I)-UPRT(I))*UPRT(I)+UPRA(I)*UPRT(I)
       YCENT(I)=((UPRA(I)-UPRT(I))*UPRT(I)*UPRT(I)/2.0+
     +          UPRA(I)*UPRT(I)*UPRA(I)/2.0)/((UPRA(I)-
     +          UPRT(I))*UPRT(I)+UPRA(I)*UPRT(I))
       UPRY(I)=UPRA(I)-YCENT(I)
       IXX(I)=(1.0/3.0)*(UPRT(I)*UPRY(I)**3+UPRA(I)*
     +        (UPRA(I)-UPRY(I))**3-(UPRA(I)-UPRT(I))*
     +        (UPRA(I)-UPRY(I)-UPRT(I))**3)
       RHO(I)=SQRT(IXX(I)/AUPR(I))
       IF(I.EQ.1)THEN
        ECC(I)=YCENT(I)+T(I)/2.0
        ENODE(I)=-(T(I)/2.0+UPRT(I)/2.0)
       ELSE IF(I.EQ.NPX+1)THEN
        ECC(I)=YCENT(I)+T(I-1)/2.0
        ENODE(I)=-(T(I-1)/2.0+UPRT(I)/2.0)
       ELSE
        ECC(I)=YCENT(I)+(T(I-1)+T(I))/4.0
        ENODE(I)=-((T(I-1)+T(I))/4.0+UPRT(I)/2.0)
       ENDIF
       AUPRE(I)=AUPR(I)/(1+(ECC(I)/RHO(I))**2)
      WRITE(10,15)I,AUPR(I),YCENT(I),IXX(I),RHO(I),
     +ECC(I),AUPRE(I),ENODE(I)
15    FORMAT(/,'    Upright No',I3,/,
     +'AUPR =  ',G15.5,/,
     +'YCENT = ',G15.5,'       IXX =     ',G20.5,/,
     +'RHO =   ',G15.5,'       ECC =     ',G15.5,/,
     +'AUPRE = ',G15.5,'       ENODE =   ',G15.5)
14    CONTINUE
C
      CALL FNODES(NPX,NPY,NEX,NEY,C20,LTOTX,LTOTY,UPRE,
     +ENODE,NALL,NCX,NCY,NCZ,NTOT,NTFLAT)
      PRINT *,' Finished subroutine fnodes !@!!!!!'
```

```
C
      CALL ELEMENTS(NPX,NPY,NEX,NEY,C20,UPRE,NALL,ELS,ELBH,
     +ELBV,ELBVN,ELBHN,ELSN,PNLSE,ICS,ICBH,ICBV,NTFLAT)
      PRINT *,' Finished subroutine ele !!!!'
C
C          Iterative loop to determine best value for BETAX and BETAY
C          for each panel to satisfy convergence !
C
      DO 650 I=1,NP
      BETAX(I)=0.85
      BETAY(I)=0.85
650   CONTINUE
C
      DO 3500 IBC=1,IBCMAX
C
      PRINT *,'IBC = ',IBC
      WRITE(10,651)IBC
      WRITE(13,651)IBC
651   FORMAT(/,' %%%%%%%%%%%% Loop',I3,' for the BETAs',
     +' convergence iterations ! %%%%%%%%%%%%',/)
C
      DO 655 I=1,NP
       BPL(I)=0
       BPID(I)=0
       SXCRI(I)=0.0
       SYCRI(I)=0.0
       SXDTI(I)=0.0
       SYDTI(I)=0.0
       SXDTII(I)=0.0
       SYDTII(I)=0.0
       K1XN(I)=0.0
       K1YN(I)=0.0
       K1XNN(I)=0.0
       K1YNN(I)=0.0
       K2XN(I)=0.0
       K2YN(I)=0.0
       K2XNN(I)=0.0
       K2YNN(I)=0.0
       K1XVAL(I)=0
       K1YVAL(I)=0
       CONA(I)=0
       CONB(I)=0
       CONC(I)=0
       COND(I)=0
655   CONTINUE
```

```
C
C *************** ITERATION LOOP STARTS HERE ***********************
C
      DO 658 FN=31,36,1
      WRITE(FN,657)
657   FORMAT(/,'IBC ITN K1X=SXC/SXCR',4X,'SXC',3X,'SXCR=SXYCR/SXY',
     +4X,'SXYCR',8X,'SXY',6X,'SX-SXCR',7X,'SY-SYCR',8X,'SX',8X,'SXAV',
     +6X,'SXDT(I)')
658   CONTINUE
      DO 661 FN2=41,46,1
      WRITE(FN2,662)
662   FORMAT(/,'IBC ITN K1Y=SYC/SYCR',4X,'SYC',3X,'SYCR=SXYCR/SXY',
     +4X,'SXYCR',8X,'SXY',6X,'SY-SYCR',7X,'SX-SXCR',8X,'SY',8X,'SYAV',
     +6X,'SYDT(I)')
661   CONTINUE
C
      DO 3000 ITN=1,20
C
      WRITE(10,692)ITN
      WRITE(21,692)ITN
692   FORMAT(/,'###################  Iteration ',I2,
     +'  #########################',/)
C
      DO 690 I=1,NP
       SXTP(I)=0.0
       SYTP(I)=0.0
       SXYTP(I)=0.0
       SXAVP(I)=0.0
       SYAVP(I)=0.0
       SXYAVP(I)=0.0
690   CONTINUE
C
      DO 691 J=1,ICS
       SXTE(J)=0.0
       SYTE(J)=0.0
       SXYTE(J)=0.0
       SXAVE(J)=0.0
       SYAVE(J)=0.0
       SXYAVE(J)=0.0
691   CONTINUE
C
      CALL FEMINP(NPX,NPY,NEX,NEY,C20,ICS,ICBH,ICBV,
     +NTFLAT,NTOT,PNLSE,ITN,UPRE,T,SXPS,SYPS,SXYPS,NALL,
     +NCX,NCY,NCZ,ELS,ELSN,ELBHN,ELBH,ELBVN,ELBV)
C
```

```
      CALL SYSTEM("abq58 job=femmodel interactive")
C
      CALL SYSTEM("strdata.x")
C
C           read in the stresses for the panel elements
C
      NRBNF=0
      WRITE(12,1082)
1082  FORMAT(/,2X'ELE',3X,'INTGR PT',9X,'SECT PT',11X,'S11',11X,'S22',
     +3X,'S12')
      OPEN(4,FILE='panelstress.txt')
      DO 1080 I=1,10000000
       READ(4,*,END=1081)ELE(I),IPT(I),SPT(I),S11(I),S22(I),S12(I)
       WRITE(12,*)ELE(I),IPT(I),SPT(I),S11(I),S22(I),S12(I)
1080  CONTINUE
1081  NRBNF=I-1
      PRINT *,'NRBNF =',NRBNF
      CLOSE(4)
C
      DO 1090 I=1,ICS
      ICES=0
       WRITE(10,1092)ELSN(I)
1092   FORMAT(/,'Element Number:',I4,/,'-------------------',
     + /,8X,'Integr pt',
     + 6X,'S11',17X,'S22',17X,'S12')
      DO 1095 J=1,NRBNF
        IF(ELSN(I).EQ.ELE(J))THEN
        SXTE(I)=SXTE(I)+S11(J)
        SYTE(I)=SYTE(I)+S22(J)
        SXYTE(I)=SXYTE(I)+S12(J)
        ICES=ICES+1
        WRITE(10,1096)IPT(J),S11(J),S22(J),S12(J)
1096      FORMAT(10X,I4,6X,F18.3,2X,F18.3,2X,F18.3)
        ENDIF
1095  CONTINUE
       WRITE(10,1099)ICES,SXTE(I),SYTE(I),SXYTE(I)
1099   FORMAT(/,'TOTAL:',4X,I4,10X,F18.3,2X,F18.3,2X,F18.3)
        SXAVE(I)=SXTE(I)/ICES
        SYAVE(I)=SYTE(I)/ICES
        SXYAVE(I)=SXYTE(I)/ICES
        WRITE(10,1098)SXAVE(I),SYAVE(I),SXYAVE(I)
1098    FORMAT(/,'AVERAGE:',12X,F18.3,2X,F18.3,2X,F18.3,/)
1090  CONTINUE
C
      DO 1120 I=1,NP
```

```
       WRITE(10,1122)I
1122   FORMAT(/,'Panel Number: ',I3,/,'-----------------',/,
     + 9X,'SXAVE',16X,'SYAVE',13X,
     +'SXYAVE')
      DO 1130 J=1,NSEPP
      DO 1131 K=1,ICS
       IF(PNLSE(I,J).EQ.ELSN(K))THEN
       SXTP(I)=SXTP(I)+SXAVE(K)
       SYTP(I)=SYTP(I)+SYAVE(K)
       SXYTP(I)=SXYTP(I)+SXYAVE(K)
       WRITE(10,1140)SXAVE(K),SYAVE(K),SXYAVE(K)
1140    FORMAT(20X,F18.3,2X,F18.3,2X,F18.3)
        ENDIF
1131  CONTINUE
1130  CONTINUE
       WRITE(10,1125)SXTP(I),SYTP(I),SXYTP(I)
1125   FORMAT(/,'TOTAL:',18X,F18.3,2X,F18.3,2X,F18.3)
        SXAVP(I)=SXTP(I)/NSEPP
        SYAVP(I)=SYTP(I)/NSEPP
        SXYAVP(I)=SXYTP(I)/NSEPP
        WRITE(10,1128)SXAVP(I),SYAVP(I),SXYAVP(I)
1128    FORMAT(/,'AVERAGE:',12X,F18.3,2X,F18.3,2X,F18.3,/)
1120  CONTINUE
C
C          calculate average panel stress
C
      NPC=0
      BTOT=0
      CTOT=0
      CONAT=0
      CONBT=0
      CONCT=0
      CONDT=0
      DO 1163 IJ=1,NP
       K1XNC(IJ)=0
       K1YNC(IJ)=0
       K1XID(IJ)=0
       K1YID(IJ)=0
1163  CONTINUE
      WRITE(13,1166)IBC,ITN
1166  FORMAT('IBC = ',I4,/,'ITN = ',I4)
C
C          Individual panel loop !
C
      XFAC1=0.0
```

```
      IFAC=0
      DO 3002 I=1,NP
      WRITE(10,1161)I,ITN
1161  FORMAT(/,1X,'******************** Analyzing Panel No',I3,
     +' / Iteration No',I3,' *************************',/)
C
      SXAV=SXAVP(I)
      SYAV=SYAVP(I)
      SXYAV=SXYAVP(I)
      WRITE(10,1162)SXAV,SYAV,SXYAV
1162  FORMAT('SXAV = ',F18.3,'   SYAV = ',F18.3,'   SXYAV = ',F18.3)
C
      WRITE(*,1031)AFL(I),AFL(I+NPX)
      WRITE(10,1031)AFL(I),AFL(I+NPX)
1031  FORMAT('AFL(I) = ',F15.10,'    AFL(I+NPX) = ',F15.10)
       AX=0.5*(AFL(I)+AFL(I+NPX))
       IF(UPRE.EQ.'ECC')THEN
        AY=AUPRE(I)
       ELSE
        AY=0.5*(AUPR(I+IFAC)+AUPR(I+1+IFAC))
       ENDIF
       IFAC=INT(I/NPX)
      WRITE(10,1021)AX,AY,IFAC
1021  FORMAT('AX = ',F15.10,'    AY = ',F15.10,/,'IFAC = ',I4)
C
C         Buckling coefficients
C
      KX=1.985*((LY**2)/(LX**2))+0.941*(LY/LX)+6.31
      KY=1.985*((LX**2)/(LY**2))+0.941*(LX/LY)+6.31
      IF(LX.GE.LY)THEN
         according to ILENGTH requirements !
       L1=LY
       L2=LX
       ELSE
       L1=LX
       L2=LY
      ENDIF
      KS=5.21*((L1**2)/(L2**2))+0.14*(L1/L2)+8.05
C
      WRITE(10,FMT=1020)KX,KY,KS,L1,L2
1020  FORMAT('KX = ',F15.9,/,'KY = ',F15.9,/,'KS = ',
     +F15.9,/,'L1 = ',F15.9,/,'L2 = ',F15.9)
      IF(LX.GE.LY)THEN
       B=LY
       ELSE
```

```
        B=LX
       ENDIF
       WRITE(10,FMT=1042)B
1042   FORMAT('B = ',F15.9)
C
       SXCR0=KX*EP*((T(I)/LY)**2)
       SYCR0=KY*EP*((T(I)/LX)**2)
       SXYCR0=KS*EP*((T(I)/B)**2)
       WRITE(10,FMT=1045)SXCR0,SYCR0,SXYCR0
1045   FORMAT('SXCR0 = ',F15.3,3X,'SYCR0 = ',F15.3,3X,'SXYCR0 = ',F15.3)
C
       L=1.0
       IF(ITN.EQ.1)THEN
C
       SX=-1.0*SXAV
       SY=-1.0*SYAV
       SXY=SXYAV
       WRITE(10,1200)SX,SY,SXY
1200   FORMAT('SX = ',F18.3,'  SY = ',F18.3,'  SXY = ',F18.3)
C
       IF(SX.LT.0.0.AND.SY.LT.0.0)THEN
       SXYCR=SXYCR0*((1-SX/SXCR0-SY/SYCR0)**0.5)
       ELSE
       BB=((SXYCR0**2)*(SX*SYCR0+SY*SXCR0))/(ABS(SXY)*SXCR0*SYCR0)
       SXYCR1=(-BB+(BB**2+4*SXYCR0**2)**0.5)/2.0
       SXYCR2=(-BB-(BB**2+4*SXYCR0**2)**0.5)/2.0
       WRITE(10,1210)BB,SXYCR1,SXYCR2
1210   FORMAT('BB = ',F18.3,/,'SXYCR1 = ',F18.3,/,
      +'SXYCR2 = ',F18.3)
       IF(ABS(SXYCR1).GE.ABS(SXYCR2))THEN
        SXYCR=SXYCR1
       ELSE
        SXYCR=SXYCR2
       ENDIF
       ENDIF
       SXCR=SX*(ABS(SXYCR)/ABS(SXY))
       SYCR=SY*(ABS(SXYCR)/ABS(SXY))
       WRITE(10,1220)SXYCR,SXCR,SYCR
1220   FORMAT('SXYCR= ',F18.3,'  SXCR= ',F18.3,'  SYCR= ',F18.3)
C
       BUCKT1=SXCR/SXCR0
       BUCKT2=SYCR/SYCR0
       BUCKT3=(SXYCR/SXYCR0)**2
        WRITE(10,1215)BUCKT1,BUCKT2,BUCKT3
1215    FORMAT('BUCKT1 = ',F20.10,/,'BUCKT2 = ',F20.10,/,
```

```
     + 'BUCKT3 = ',F20.10)
      BUCK=SXCR/SXCR0+SYCR/SYCR0+(SXYCR/SXYCR0)**2
      TDB=NINT(BUCK*10000.0)/10000.0
      WRITE(10,1222)BUCK,TDB
1222  FORMAT('Buckling Equation = ',F20.10,/,'TDB =',F20.4)
C
      IF(TDB.LT.1.0)THEN
        BPL(I)=BPL(I)+1
      WRITE(10,1224)I
1224  FORMAT('    Panel no',I2,' does not buckle !',/)
      GOTO 3002
      ENDIF
C
      K=TANH(0.5*LOG10(ABS(SXY)/ABS(SXYCR)))
C
      H1=(-K*ABS(SXY))/(EXB*(2.0*AX/LY/T(I)+0.5*(1-K)))
      H2=(-K*ABS(SXY))/(EYB*(AY/LX/T(I)+0.5*(1-K)))
      H3=ABS(SXY)*2.0*K/EP
      H4=(1-K+POISS-POISS*K)*(ABS(SXY)/EP)
      WRITE(10,1230)K,H1,H2,H3,H4
1230  FORMAT('K = ',F7.5,/,
     +'H1 = ',F15.11,'  H2 = ',F15.11,'  H3 = ',F15.11,
     +'  H4 = ',F15.11)
C
      XALFA=PI/4.0
1240  WRITE(10,1245)XALFA
1245  FORMAT('XALFA = ',F10.7)
      EPSX=H1/TAN(XALFA)
      EPSY=H2*TAN(XALFA)
      EPS=H3/SIN(2*XALFA)+H4*SIN(2*XALFA)
      WRITE(10,1250)EPSX,EPSY,EPS
1250  FORMAT('EPSX = ',F10.8,'   EPSY = ',F10.8,'   EPS = ',F10.8)
C
      IF((EPS-EPSX).LT.0.0.AND.(EPS-EPSY).GT.0.0)THEN
       PRINT *,' Diagonal tension angle cannot be calculated !'
       GOTO 3001
      ELSE IF((EPS-EPSX).GT.0.0.AND.(EPS-EPSY).LT.0.0)THEN
       PRINT *,' Diagonal tension angle cannot be calculated !'
       GOTO 3001
      ELSE IF((EPS-EPSY).EQ.0.0)THEN
       PRINT *,' Diagonal tension angle cannot be calculated !'
       GOTO 3001
      ENDIF
      ALFA=ATAN(((EPS-EPSX)/(EPS-EPSY))**0.5)
      ERR=(ABS(ALFA-XALFA))*100.0
```

```
      WRITE(10,1255)ALFA,ERR
1255  FORMAT('ALFA = ',F10.7,' % ERR = ',F6.3)
      IF(ERR.GE.0.1)THEN
       XALFA=ALFA
       GOTO 1240
      ENDIF
      WRITE(10,1256)I,ALFA*180.0/3.141596
1256  FORMAT(/,'   Diagonal Tension Angle for Panel',I3,' = ',F6.3,
     +' degrees !',/)
C
      SXDT(I)=K*ABS(SXY)/TAN(ALFA)
      SYDT(I)=K*ABS(SXY)*TAN(ALFA)
      WRITE(10,1260)SXDT(I),SYDT(I)
1260  FORMAT('SXDT(I) = ',F18.3,'SYDT(I) = ',F18.3)
C
      EXDT=(-1.0/EP)*(EP*LY*T(I)/2.0/AX/EXB+1)*(SXDT(I)-POISS*SYDT(I))
      EYDT=(-1.0/EP)*(EP*LX*T(I)/AY/EYB+1)*(SYDT(I)-POISS*SXDT(I))
      WRITE(10,1265)EXDT,EYDT
1265  FORMAT('EXDT = ',F13.10,'  EYDT = ',F13.10)
C
      THETA=(1.0/(2*(1.0+POISS)))*(4.0/(SIN(2*ALFA))**2+
     +(EP*LX*T(I)*(TAN(ALFA))**2)/(EYB*(AY+(0.5*LX*T(I))*(1-K)))+
     +(EP*LY*T(I)/(TAN(ALFA))**2)/(EXB*(2.0*AX+(0.5*LY*T(I))*(1-K))))
      EXYDT=K*(THETA-1)*SXY/GP
C
      C1=1+LY*T(I)*EP*BETAX(I)/2.0/AX/EXB
      C2=1+LX*T(I)*EP*BETAY(I)/AY/EYB
      C3=1-BETAX(I)
      C4=1-BETAY(I)
      WRITE(10,1270)THETA,EXYDT,BETAX(I),BETAY(I),C1,C2,C3,C4
1270  FORMAT('THETA = ',F10.6,'   EXYDT = ',F13.10,/,
     +'BETAX(I) = ',F10.6,'   BETAY(I) = ',F10.6,/,
     +'C1 = ',F10.6,'  C2 = ',F10.6,'  C3 = ',F10.6,'  C4 = ',F10.6)
C
      SXC=(C2*C3*(SX-SXCR)*L+BETAX(I)*POISS*C4*(SY-SYCR)*L)/C1/C2
      SYC=(C1*C4*(SY-SYCR)*L+BETAY(I)*POISS*C3*(SX-SXCR)*L)/C1/C2
      WRITE(10,1275)SXC,SYC
1275  FORMAT('SXC = ',F18.3,'   SYC = ',F18.3)
C
      EXC=(-1.0/EP)*(1+LY*T(I)*EP/2.0/AX/EXB)*(SXC-POISS*SYC)*
     +(ABS(BETAX(I)+BETAX(I)*BETAY(I)-1))
      EYC=(-1.0/EP)*(1+LX*T(I)*EP/AY/EYB)*(SYC-POISS*SXC)*
     +(ABS(BETAY(I)+BETAX(I)*BETAY(I)-1))
      WRITE(10,1280)EXC,EYC
1280  FORMAT('EXC = ',F13.10,'   EYC = ',F13.10)
```

```
C
      EXT(I)=EXDT+EXC
      EYT(I)=EYDT+EYC
      EXYT=EXYDT
      WRITE(10,1285)EXT(I),EYT(I),EXYT
1285  FORMAT('EXT(I) = ',F13.10,'   EYT(I) = ',F13.10,
     +'   EXYT = ',F13.10)
C
      SXPS(I)=-EP*(EXT(I)+POISS*EYT(I))/(1-POISS**2)
      SYPS(I)=-EP*(EYT(I)+POISS*EXT(I))/(1-POISS**2)
      SXYPS(I)=-EP*EXYT/(2*(1+POISS))
      WRITE(10,1290)SXPS(I),SYPS(I),SXYPS(I)
1290  FORMAT('SXPS(I) = ',F18.3,'   SYPS(I) = ',F18.3,
     +'   SXYPS = ',F18.3)
C
      IF(ITN.GE.2)THEN
      SXMOD=SXAV-SXDTI(I)
      SYMOD=SYAV-SYDTI(I)
      WRITE(10,1363)SXMOD,SYMOD
1363  FORMAT('SXMOD = ',F18.3,'  SYMOD = ',F18.3)
C
      SX=-1.0*SXMOD
      SY=-1.0*SYMOD
      SXY=SXYAV
      WRITE(10,1360)SX,SY,SXY
1360  FORMAT('SX = ',F18.3,'  SY = ',F18.3,'  SXY = ',F18.3)
C
      IF(SX.LT.0.0.AND.SY.LT.0.0)THEN
      SXYCR=SXYCR0*((1-SX/SXCR0-SY/SYCR0)**0.5)
      ELSE
      BB=((SXYCR0**2)*(SX*SYCR0+SY*SXCR0))/(ABS(SXY)*SXCR0*SYCR0)
      SXYCR1=(-BB+(BB**2+4*SXYCR0**2)**0.5)/2.0
      SXYCR2=(-BB-(BB**2+4*SXYCR0**2)**0.5)/2.0
      WRITE(10,1370)BB,SXYCR1,SXYCR2
1370  FORMAT('BB = ',F18.3,/,'SXYCR1 = ',F18.3,/,
     +'SXYCR2 = ',F18.3)
      IF(ABS(SXYCR1).GE.ABS(SXYCR2))THEN
       SXYCR=SXYCR1
      ELSE
       SXYCR=SXYCR2
      ENDIF
      ENDIF
      SXCR=SX*(ABS(SXYCR)/ABS(SXY))
      SYCR=SY*(ABS(SXYCR)/ABS(SXY))
      WRITE(10,1380)SXYCR,SXCR,SYCR
```

```
1380  FORMAT('SXYCR= ',F18.3,'  SXCR= ',F18.3,'  SYCR= ',F18.3)
C
      BUCKT1=SXCR/SXCR0
      BUCKT2=SYCR/SYCR0
      BUCKT3=(SXYCR/SXYCR0)**2
       WRITE(10,1376)BUCKT1,BUCKT2,BUCKT3
1376   FORMAT('BUCKT1 = 'F20.10,/,'BUCKT2 = 'F20.10,/,
     + 'BUCKT3 = ',F20.10)
      BUCK=SXCR/SXCR0+SYCR/SYCR0+(SXYCR/SXYCR0)**2
      TDB=NINT(BUCK*10000.0)/10000.0
      WRITE(10,1382)BUCK,TDB
1382  FORMAT('Buckling Equation = ',F20.10,/,'TDB = ',F20.4)
C
      IF(TDB.LT.1.0)THEN
       BPL(I)=BPL(I)+1
      WRITE(10,1384)I
1384  FORMAT('    Panel no',I2,' does not buckle !',/)
      GOTO 3002
      ENDIF
C
      K=TANH(0.5*LOG10(ABS(SXY)/ABS(SXYCR)))
C
      H1=(-K*ABS(SXY))/(EXB*(2.0*AX/LY/T(I)+0.5*(1-K)))
      H2=(-K*ABS(SXY))/(EYB*(AY/LX/T(I)+0.5*(1-K)))
      H3=ABS(SXY)*2.0*K/EP
      H4=(1-K+POISS-POISS*K)*(ABS(SXY)/EP)
      WRITE(10,1400)K,H1,H2,H3,H4
1400  FORMAT('K = ',F7.5,/,
     +'H1 = ',F15.11,'  H2 = ',F15.11,'  H3 = ',F15.11,
     +'  H4 = ',F15.11)
C
      XALFA=3.141596/4.0
1450  WRITE(10,1420)XALFA
1420  FORMAT('XALFA = ',F10.7)
      EPSX=H1/TAN(XALFA)
      EPSY=H2*TAN(XALFA)
      EPS=H3/SIN(2*XALFA)+H4*SIN(2*XALFA)
      WRITE(10,1430)EPSX,EPSY,EPS
1430  FORMAT('EPSX = ',F10.8,'  EPSY = ',F10.8,'  EPS = ',F10.8)
C
      IF((EPS-EPSX).LT.0.0.AND.(EPS-EPSY).GT.0.0)THEN
       PRINT *,' Diagonal tension angle cannot be calculated !'
       GOTO 3001
      ELSE IF((EPS-EPSX).GT.0.0.AND.(EPS-EPSY).LT.0.0)THEN
       PRINT *,' Diagonal tension angle cannot be calculated !'
```

```
        GOTO 3001
       ELSE IF((EPS-EPSY).EQ.0.0)THEN
        PRINT *,' Diagonal tension angle cannot be calculated !'
        GOTO 3001
       ENDIF
       ALFA=ATAN(((EPS-EPSX)/(EPS-EPSY))**0.5)
       ERR=(ABS(ALFA-XALFA))*100.0
       WRITE(10,1440)ALFA,ERR
 1440  FORMAT('ALFA = ',F10.7,'  % ERR = ',F6.3)
       IF(ERR.GE.0.1)THEN
       XALFA=ALFA
       GOTO 1450
       ENDIF
       WRITE(10,1460)I,ALFA*180.0/3.141596
 1460  FORMAT(/,'   Diagonal Tension Angle for Panel',I3,' = ',F6.3,
      +' degrees !',/)
C
       SXDT(I)=K*ABS(SXY)/TAN(ALFA)
       SYDT(I)=K*ABS(SXY)*TAN(ALFA)
       WRITE(10,1500)SXDT(I),SYDT(I)
 1500  FORMAT('SXDT(I) = ',F18.3,'   SYDT(I) = ',F18.3)
C
C               Tests 1 to 5 in the Grisham algorithm
C
       BBETAX=0.0
       BBETAY=0.0
C
 2400  EXDT=(-1.0/EP)*(EP*LY*T(I)/2.0/AX/EXB+1)*(SXDT(I)-(1-BBETAX)*
      +SXDTI(I)-POISS*(SYDT(I)-(1-BBETAX)*SYDTI(I)))
       EYDT=(-1.0/EP)*(EP*LX*T(I)/AY/EYB+1)*(SYDT(I)-(1-BBETAY)*
      +SYDTI(I)-POISS*(SXDT(I)-(1-BBETAY)*SXDTI(I)))
       WRITE(10,2540)EXDT,EYDT
 2540  FORMAT('EXDT = ',F13.10,'  EYDT = ',F13.10)
C
       THETA=(1.0/(2*(1.0+POISS)))*(4.0/(SIN(2*ALFA))**2+
      +(EP*LX*T(I)*(TAN(ALFA))**2)/(EYB*(AY+(0.5*LX*T(I))*(1-K)))+
      +(EP*LY*T(I)/(TAN(ALFA))**2)/(EXB*(2.0*AX+(0.5*LY*T(I))*(1-K))))
       EXYDT=K*(THETA-1)*SXY/GP
       WRITE(10,2560)THETA,EXYDT
 2560  FORMAT('THETA = ',F10.6,'  EXYDT = ',F13.10)
C
       C1=1+LY*T(I)*EP*BETAX(I)/2.0/AX/EXB
       C2=1+LX*T(I)*EP*BETAY(I)/AY/EYB
       C3=1-BETAX(I)
       C4=1-BETAY(I)
```

```
      WRITE(10,2580)C1,C2,C3,C4,BETAX(I),BETAY(I)
2580  FORMAT('C1 = ',F10.6,'   C2 = ',F10.6,'   C3 = ',F10.6,
     +'   C4 = ',F10.6,/,'BETAX(I) = ',F10.6,'   BETAY(I) = ',F10.6)
C
      SXC=(C2*C3*(SX-SXCR)*L+BETAX(I)*POISS*C4*(SY-SYCR)*L)
     +/C1/C2
      SYC=(C1*C4*(SY-SYCR)*L+BETAY(I)*POISS*C3*(SX-SXCR)*L)
     +/C1/C2
      WRITE(10,2600)SXC,SYC
2600  FORMAT('SXC = ',F18.3,'  SYC = ',F18.3)
C
      EXC=(-1.0/EP)*(1+LY*T(I)*EP/2.0/AX/EXB)*(SXC-POISS*SYC)*
     +(ABS(BETAX(I)+BETAX(I)*BETAY(I)-1))
      EYC=(-1.0/EP)*(1+LX*T(I)*EP/AY/EYB)*(SYC-POISS*SXC)*
     +(ABS(BETAY(I)+BETAX(I)*BETAY(I)-1))
      WRITE(10,2620)EXC,EYC
2620  FORMAT('EXC = ',F13.10,'   EYC = ',F13.10)
C
      EXT(I)=(1-BBETAX)*EXT(I)+EXDT+EXC
      EYT(I)=(1-BBETAY)*EYT(I)+EYDT+EYC
      EXYT=EXYDT
      WRITE(10,2640)EXT(I),EYT(I),EXYT
2640  FORMAT('EXT = ',F13.10,'   EYT = ',F13.10,'   EXYT = 'F13.10)
C
      SXPS(I)=-EP*(EXT(I)+POISS*EYT(I))/(1-POISS**2)
      SYPS(I)=-EP*(EYT(I)+POISS*EXT(I))/(1-POISS**2)
      SXYPS(I)=-EP*EXYT/(2*(1+POISS))
C
      WRITE(10,2660)SXPS(I),SYPS(I),SXYPS(I)
2660  FORMAT('SXPS(I) = ',F18.3,'   SYPS(I) = ',F18.3,
     +'   SXYPS(I) = ',F18.3)
      ENDIF
C
C         calculate convergence parameters
C
      WRITE(10,2680)SXC,SXCR,SYC,SYCR
2680  FORMAT('SXC = ',F18.3,'   SXCR = ',F18.3,/,
     +'SYC = ',F18.3,'   SYCR = ',F18.3)
      IF(ABS(SXCR).LT.1.0E-5)THEN
      WRITE(21,2682)
2682  FORMAT('  K1X  --> INF ;   K1XP  --> 0')
      ELSE IF(ABS(SXC).LT.1.0E-5)THEN
      WRITE(21,2684)
2684  FORMAT('  K1X  --> 0 ;   K1XP  --> INF')
      ELSE
```

```
      K1X=SXC/SXCR
      K1XP=(K1X-K1XN(I))*100.0/K1X
      K1XPP=(K1XN(I)-K1XNN(I))*100.0/K1XN(I)
      WRITE(10,2687)K1X,K1XP,K1XPP
2687  FORMAT('K1X = ',F10.3,5X,'K1XP = ',F10.3,5X,'K1XPP = ',
     +F10.3)
      ENDIF
C
      IF(ABS(SYCR).LT.1.0E-5)THEN
      WRITE(21,2686)
2686  FORMAT('  K1Y  --> INF ;   K1YP  --> 0')
      ELSE IF(ABS(SYC).LT.1.0E-5)THEN
      WRITE(21,2688)
2688  FORMAT('  K1Y  --> 0 ;   K1YP  --> INF')
      ELSE
      K1Y=SYC/SYCR
      K1YP=(K1Y-K1YN(I))*100.0/K1Y
      K1YPP=(K1YN(I)-K1YNN(I))*100.0/K1YN(I)
      WRITE(10,2689)K1Y,K1YP,K1YPP
2689  FORMAT('K1Y = ',F10.3,5X,'K1YP = ',F10.3,5X,'K1XPP = ',
     +F10.3)
      ENDIF
C
      FN=I+30
      WRITE(FN,2662)IBC,ITN,K1X,SXC,SXCR,SXYCR,SXY,SX-SXCR,SY-SYCR,
     +SX,SXAV,SXDT(I)
2662  FORMAT(2I3,10G12.5)
C
      FN2=I+40
      WRITE(FN2,2663)IBC,ITN,K1Y,SYC,SYCR,SXYCR,SXY,SY-SYCR,SX-SXCR,
     +SY,SYAV,SYDT(I)
2663  FORMAT(2I3,10G12.5)
C
2704  WRITE(10,2700)SXDT(I),SXDTI(I),SYDT(I),SYDTI(I)
2700  FORMAT('SXDT(I) = ',F18.3,'   SXDTI(I) = ',F18.3,/,
     +'SYDT(I) = ',F18.3,'   SYDTI(I) = ',F18.3)
      K2X=SXDT(I)-SXDTI(I)
      K2Y=SYDT(I)-SYDTI(I)
      K2XP=(K2X/SXDT(I))*100.0
      K2YP=(K2Y/SYDT(I))*100.0
      K2XPP=(K2XN(I)/SXDTI(I))*100.0
      K2YPP=(K2YN(I)/SYDTI(I))*100.0
      K2XPPP=(K2XNN(I)/SXDTII(I))*100.0
      K2YPPP=(K2YNN(I)/SYDTII(I))*100.0
      WRITE(10,2710)I,K1X,K1XP,K1Y,K1YP,K1XPP,K1YPP,
```

```
     +K2X,K2XP,K2Y,K2YP,K2XPP,K2YPP,K2XPPP,K2YPPP
      WRITE(21,2710)I,K1X,K1XP,K1Y,K1YP,K1XPP,K1YPP,
     +K2X,K2XP,K2Y,K2YP,K2XPP,K2YPP,K2XPPP,K2YPPP
2710  FORMAT('Panel No ',I2,/,'K1X = ',F18.3,'(',F10.3,' %)',
     +5X,'K1Y = ',F18.3,' (',F10.3,' %)',/,24X,' (',F10.3,' %)',
     +29X,' (',F10.3,' %)',/,
     +'K2X = ',F18.3,'(',F10.3,' %)',5X,'K2Y = ',
     +F18.3,' (',F10.3,' %)',/,24X,' (',F10.3,' %)',29X,
     +' (',F10.3,' %)',/,24X,' (',F10.3,' %)',29X,
     +' (',F10.3,' %)')
C
      SXCRI(I)=SXCR
      SYCRI(I)=SYCR
      SXDTII(I)=SXDTI(I)
      SYDTII(I)=SYDTI(I)
      SXDTI(I)=SXDT(I)
      SYDTI(I)=SYDT(I)
      WRITE(10,2720)SXCRI(I),SYCRI(I),SXDTI(I),SYDTI(I),
     +SXDTII(I),SYDTII(I)
2720  FORMAT('SXCRI(I) = ',F18.3,'     SYCRI(I) = ',F18.3,/,
     +'SXDTI(I) = ',F18.3,'     SYDTI(I) = ',F18.3,/,
     +'SXDTII(I) = ',F18.3,'     SYDTII(I) = ',F18.3)
      K1XNN(I)=K1XN(I)
      K1YNN(I)=K1YN(I)
      K1XN(I)=K1X
      K1YN(I)=K1Y
      K2XNN(I)=K2XN(I)
      K2YNN(I)=K2YN(I)
      K2XN(I)=K2X
      K2YN(I)=K2Y
C
      WRITE(10,2726)K1XNN(I),K1YNN(I),K1XN(I),K1YN(I)
2726  FORMAT('K1XNN(I) = ',F10.3,5X,'K1YNN(I) = ',F10.3,/,
     +'K1XN(I) = ',F10.3,5X,'K1YN(I) = ',F10.3)
      WRITE(10,2728)K2XNN(I),K2YNN(I),K2XN(I),K2YN(I)
2728  FORMAT('K2XNN(I) = ',F18.3,5X,'K2YNN(I) = ',F18.3,/,
     +'K2XN(I) = ',F18.3,5X,'K2YN(I) = ',F18.3)
C
C         Test for convergence:
C
      IF(ITN.GE.3)THEN
      IF(ABS(K1XP).LT.5.0.AND.ABS(K1XPP).LT.5.0)THEN
      WRITE(13,2730)K1X,I,ITN
2730    FORMAT(4X,'K1X converges to',F10.3,' for plate ',I3,
     +  ' !! (ITN=',I3,')')
```

```
          K1XVAL(I)=K1X
        CONA(I)=1
        ELSE IF(CONA(I).NE.1)THEN
         K1XNC(I)=K1X
         K1XID(I)=1
         WRITE(13,2735)K1XNC(I),K1XID(I),I
2735     FORMAT('K1XNC(I) =',F15.7,'    K1XID(I) =',I3,'    I =',I3)
        ENDIF
        IF(ABS(K1YP).LT.5.0.AND.ABS(K1YPP).LT.5.0)THEN
        WRITE(13,2740)K1Y,I,ITN
2740      FORMAT(4X,'K1Y converges to',F10.3,' for plate ',I3,
     +   ' !! (ITN=',I3,')')
          K1YVAL(I)=K1Y
        CONB(I)=1
        ELSE IF(CONB(I).NE.1)THEN
         K1YNC(I)=K1Y
         K1YID(I)=1
         WRITE(13,2745)K1YNC(I),K1YID(I),I
2745     FORMAT('K1YNC(I) =',F15.7,'    K1YID(I) =',I3,'     I =',I3)
        ENDIF
        IF(ABS(K2XP).LT.2.0.AND.ABS(K2XPP).LT.2.0.AND.ABS(K2XPPP).
     +LT.2.0)THEN
        WRITE(13,2780)I,ITN
2780      FORMAT(4X,'K2X converges for plate ',I3,' !! (ITN=',
     +   I3,')')
        CONC(I)=1
        ENDIF
        IF(ABS(K2YP).LT.2.0.AND.ABS(K2YPP).LT.2.0.AND.ABS(K2YPPP).
     +LT.2.0)THEN
        WRITE(13,2790)I,ITN
2790      FORMAT(4X,'K2Y converges for plate ',I3,' !! (ITN=',
     +   I3,')')
        COND(I)=1
        ENDIF
        ENDIF
          CONAT=CONAT+CONA(I)
          CONBT=CONBT+CONB(I)
          CONCT=CONCT+CONC(I)
          CONDT=CONDT+COND(I)
3002    CONTINUE
C
        IF(CONAT+CONBT+CONCT+CONDT.EQ.4*NP)THEN
         NOIT=ITN
         WRITE(13,3003)NOIT
3003     FORMAT('NOIT = ',I3)
```

```
        GOTO 3006
       ELSE
        NOIT=ITN
       ENDIF
C
3000   CONTINUE
C
3006   WRITE(13,3007)ITN
3007   FORMAT('ITN = ',I5)
       BCTOT=0
       DO 3005 I=1,NP
        BCONA(I)=0
        BCONB(I)=0
3005   CONTINUE
C
       IF(IBC.EQ.IBCMAX)THEN
        WRITE(13,3206)IBC
3206    FORMAT('Maximum number of iterations reached - ',I3,' - still',
     + ' no convergence !')
        GOTO 3001
       ENDIF
C
       DO 3051 I=1,NP
C
C           x-BETA values
C
C           modify the BETAs when the solution does not buckle
C
        IF(BPL(I).EQ.NOIT)THEN
        WRITE(13,3011)I,BETAX(I)
3011    FORMAT('Panel no',I3,' converges to an unbuckled state !',/,
     + 'The BETAX value will therefore remain as is:',/,
     + 'BETAX(I) = ',F10.7)
        ELSE IF(BPL(I).LT.NOIT.AND.BPL(I).GT.0.AND.CONA(I).EQ.0)THEN
        WRITE(13,3012)I,BPL(I)
3012     FORMAT('Panel number',I3,'  buckled less than 12 times (BPL=',
     +  I3,');',/,'probably does not converge to an unbuckled state !')
        WRITE(13,3013)I,BETAX(I)
3013     FORMAT('Old BETAX(I) value for panel no',I3,' is:',F10.7)
         BETAX(I)=BETAX(I)+0.02
        IF(BETAX(I).GE.1.0)BETAX(I)=0.001
        WRITE(13,3014)I,BETAX(I)
3014     FORMAT('New BETAX(I) value for panel no',I3,' is:',F10.7)
C
C           modify the BETAs when the solution does not converge
```

```
C
       ELSE IF(K1XID(I).EQ.1.AND.K1XNC(I).LT.1.0)THEN
       WRITE(13,3050)I,BETAX(I)
3050    FORMAT('K1X of panel no',I3,' did not converge !',/,
     +  'Old BETAX(I) value for panel',I3,' is: ',F10.7)
       BETAX(I)=BETAX(I)-0.016
       IF(BETAX(I).LE.0.0)BETAX(I)=0.001
       WRITE(13,3055)I,BETAX(I)
3055    FORMAT('New BETAX(I) value for panel',I3,' is: ',F10.7)
       ELSE IF(K1XID(I).EQ.1.AND.K1XNC(I).GT.1.0)THEN
       WRITE(13,3060)I,I,BETAX(I)
3060    FORMAT('K1X of panel no',I3,' did not converge !',/,
     +  'Old BETAX(I) value for panel',I3,' is: ',F10.7)
       BETAX(I)=BETAX(I)+0.02
       IF(BETAX(I).GE.1.0)BETAX(I)=0.999
       WRITE(13,3065)I,BETAX(I)
3065    FORMAT('New BETAX(I) value for panel',I3,' is: ',F10.7)
C
C          modify the BETAs when the solution converges
C
       ELSE IF(CONA(I).EQ.1.AND.K1XVAL(I).LT.0.8)THEN
       WRITE(13,3015)I,BETAX(I)
3015    FORMAT('Old BETAX(I) value for panel',I3,' is: ',F10.7)
       BETAX(I)=BETAX(I)-0.02
        WRITE(13,3020)I,BETAX(I)
3020    FORMAT('New BETAX(I) value for panel',I3,' is: ',F10.7)
       ELSE IF(CONA(I).EQ.1.AND.K1XVAL(I).GE.0.8.AND.
     + K1XVAL(I).LT.0.95)THEN
       WRITE(13,3021)I,BETAX(I)
3021    FORMAT('Old BETAX(I) value for panel',I3,' is: ',F10.7)
       BETAX(I)=BETAX(I)-0.002
        WRITE(13,3022)I,BETAX(I)
3022    FORMAT('New BETAX(I) value for panel',I3,' is: ',F10.7)
       ELSE IF(CONA(I).EQ.1.AND.K1XVAL(I).GT.1.2)THEN
        WRITE(13,3025)I,BETAX(I)
3025    FORMAT('Old BETAX(I) value for panel',I3,' is: ',F10.7)
       BETAX(I)=BETAX(I)+0.02
        WRITE(13,3030)I,BETAX(I)
3030    FORMAT('New BETAX(I) value for panel',I3,' is: ',F10.7)
       ELSE IF(CONA(I).EQ.1.AND.K1XVAL(I).GT.1.05.AND.
     + K1XVAL(I).LE.1.2)THEN
        WRITE(13,3031)I,BETAX(I)
3031    FORMAT('Old BETAX(I) value for panel',I3,' is: ',F10.7)
       BETAX(I)=BETAX(I)+0.002
        WRITE(13,3032)I,BETAX(I)
```

```
3032    FORMAT('New BETAX(I) value for panel',I3,' is: ',F10.7)
       ELSE IF(CONA(I).EQ.1.AND.K1XVAL(I).GE.0.95.AND.
     + K1XVAL(I).LE.1.05)THEN
        BCONA(I)=1
        WRITE(13,3040)I,BETAX(I),BCONA(I)
3040    FORMAT('BETAX(I) is the correct value for K1X to converge',
     +  ' for plate ',I3,'(BETAX(I) = ',F10.7,')',/,'BCONA = ',I3)
C
C          modify BETAs when none of the above apply
C
       ELSE
        WRITE(13,3041)I,BETAX(I)
3041    FORMAT('Default change - Old BETAX(I) value for panel',I3,
     +  ' is: ',F10.7)
        BETAX(I)=BETAX(I)+0.03
         IF(BETAX(I).GE.1.0)BETAX(I)=0.001
        WRITE(13,3044)I,BETAX(I)
3044    FORMAT('Default change - New BETAX(I) value for panel',I3,
     +  ' is: ',F10.7)
       ENDIF
C
C         y-BETA values
C
C         modify the BETAs when the solution does not buckle
C
       IF(BPL(I).EQ.NOIT)THEN
       WRITE(13,2791)I,BETAY(I)
2791   FORMAT('Panel no',I3,' converges to an unbuckled state !',/,
     + 'The BETAY value will therefore remain as is:',/,
     + 'BETAY(I) = ',F10.7)
        BTOT=BTOT+1
       BPID(I)=1
       ELSE IF(BPL(I).LT.NOIT.AND.BPL(I).GT.0.AND.CONB(I).EQ.0)THEN
        WRITE(13,3128)I,BPL(I)
3128    FORMAT('Panel number',I3,'  buckled less than 12 times (BPL=',
     +  I3,');',/,'probably does not converge to an unbuckled state !')
         WRITE(13,3129)I,BETAY(I)
3129     FORMAT('Old BETAY(I) value for panel no',I3,' is:',F10.7)
       BETAY(I)=BETAY(I)+0.02
       IF(BETAY(I).GE.1.0)BETAY(I)=0.001
         WRITE(13,3131)I,BETAY(I)
3131     FORMAT('New BETAY(I) value for panel no',I3,' is:',F10.7)
C
C         modify BETAs when solution does not converge
C
```

```
      ELSE IF(K1YID(I).EQ.1.AND.K1YNC(I).LT.1.0)THEN
       WRITE(13,3110)I,I,BETAY(I)
3110   FORMAT('K1Y of panel no',I3,' did not converge !',/,
     + 'Old BETAY(I) value for panel',I3,' is: ',F10.7)
      BETAY(I)=BETAY(I)-0.016
      IF(BETAY(I).LE.0.0)BETAY(I)=0.001
       WRITE(13,3115)I,BETAY(I)
3115   FORMAT('New BETAY(I) value for panel',I3,' is: ',F10.7)
      ELSE IF(K1YID(I).EQ.1.AND.K1YNC(I).GT.1.0)THEN
       WRITE(13,3120)I,I,BETAY(I)
3120   FORMAT('K1Y of panel no',I3,' did not converge !',/,
     + 'Old BETAY(I) value for panel',I3,' is: ',F10.7)
      BETAY(I)=BETAY(I)+0.02
      IF(BETAY(I).GE.1.0)BETAY(I)=0.999
       WRITE(13,3125)I,BETAY(I)
3125   FORMAT('New BETAY(I) value for panel',I3,' is: ',F10.7)
C
C         modify BETAs when solution converges
C
      ELSE IF(CONB(I).EQ.1.AND.K1YVAL(I).LT.0.8)THEN
       WRITE(13,3075)I,BETAY(I)
3075   FORMAT('Old BETAY(I) value for panel',I3,' is: ',F10.7)
      BETAY(I)=BETAY(I)-0.02
       WRITE(13,3080)I,BETAY(I)
3080   FORMAT('New BETAY(I) value for panel',I3,' is: ',F10.7)
      ELSE IF(CONB(I).EQ.1.AND.K1YVAL(I).GE.0.8.AND.
     + K1YVAL(I).LT.0.95)THEN
       WRITE(13,3081)I,BETAY(I)
3081   FORMAT('Old BETAY(I) value for panel',I3,' is: ',F10.7)
      BETAY(I)=BETAY(I)-0.002
       WRITE(13,3082)I,BETAY(I)
3082   FORMAT('New BETAY(I) value for panel',I3,' is: ',F10.7)
      ELSE IF(CONB(I).EQ.1.AND.K1YVAL(I).GT.1.2)THEN
       WRITE(13,3085)I,BETAY(I)
3085   FORMAT('Old BETAY(I) value for panel',I3,' is: ',F10.7)
      BETAY(I)=BETAY(I)+0.02
       WRITE(13,3090)I,BETAY(I)
3090   FORMAT('New BETAY(I) value for panel',I3,' is: ',F10.7)
      ELSE IF(CONB(I).EQ.1.AND.K1YVAL(I).GT.1.05.AND.
     + K1YVAL(I).LE.1.2)THEN
       WRITE(13,3091)I,BETAY(I)
3091   FORMAT('Old BETAY(I) value for panel',I3,' is: ',F10.7)
      BETAY(I)=BETAY(I)+0.002
       WRITE(13,3092)I,BETAY(I)
3092   FORMAT('New BETAY(I) value for panel',I3,' is: ',F10.7)
```

```
       ELSE IF(CONB(I).EQ.1.AND.K1YVAL(I).GE.0.95.AND.
     + K1YVAL(I).LE.1.05)THEN
        BCONB(I)=1
        WRITE(13,3100)I,BETAY(I),BCONB(I)
3100    FORMAT('BETAY(I) is the correct value for K1Y to converge',
     +  ' for panel ',I3,'(BETAY(I) = ',F10.7,')',/,'BCONB = ',I3)
C
C          modify BETAs when none of the above apply
C
       ELSE
        WRITE(13,3101)I,BETAY(I)
3101    FORMAT('Default change - Old BETAY(I) value for panel',I3,
     +  ' is: ',F10.7)
        BETAY(I)=BETAY(I)+0.03
         IF(BETAY(I).GE.1.0)BETAY(I)=0.001
        WRITE(13,3104)I,BETAY(I)
3104    FORMAT('Default change - New BETAY(I) value for panel',I3,
     +  ' is: ',F10.7)
       ENDIF
      BCTOT=BCTOT+BCONA(I)+BCONB(I)
      WRITE(13,3180)BTOT,BCTOT
3180  FORMAT('BTOT = ',I4,'      BCTOT = ',I4)
3051  CONTINUE
C
      IF(BTOT.NE.NP.AND.BCTOT.EQ.2*(NP-BTOT))THEN
       WRITE(13,3512)
3512   FORMAT('BETAs adjusted successfully - all no 1 convergence',
     + ' criteria is satisfied !')
       IF(CONCT.EQ.NP)THEN
       WRITE(13,3513)
3513   FORMAT('The Diagonal tension stress requirement in the x-dir',/,
     + 'is satisfied (no 2 requirement)')
       ELSE
       WRITE(13,3514)
3514   FORMAT('Requirement no 2 is not satisfied in x-dir !')
       ENDIF
       IF(CONDT.EQ.NP)THEN
       WRITE(13,3516)
3516   FORMAT('The diagonal tension stress requirement in the y-dir',/,
     +  'is satisfied (no 2 requirement)')
       ELSE
       WRITE(13,3517)
3517   FORMAT('Requirement no 2 is not satisfied in y-dir !')
       ENDIF
       GOTO 3001
```

```
      ELSE IF(BTOT.EQ.NP)THEN
       WRITE(13,3518)
3518   FORMAT('No panels buckled !')
       GOTO 3001
      ENDIF
C
3500  CONTINUE
C
      CALL TIME(SEC)
      TSTOP=SEC
      WRITE(10,3505)TSTOP
3505  FORMAT('TSTOP = ',I20)
C
      CLOSE(10)
      CLOSE(20)
      CLOSE(21)
      CLOSE(13)
      CLOSE(12)
      CLOSE(31)
      CLOSE(32)
      CLOSE(33)
      CLOSE(34)
      CLOSE(35)
      CLOSE(36)
      CLOSE(41)
      CLOSE(42)
      CLOSE(43)
      CLOSE(44)
      CLOSE(45)
      CLOSE(46)
C
      STOP
      END




C*****************************************************************
C
      SUBROUTINE FNODES(NPX,NPY,NEX,NEY,C20,LTOTX,LTOTY,UPRE,
     +SENODE,SNALL,SNCX,SNCY,SNCZ,SNTOT,SNTFLAT)
C
C******************************************************************
C            Subroutine to generate all nodes for the
```

```
C            finite element model
C*****************************************************************
C
      DOUBLE PRECISION LTOTX,LTOTY,LPNX,LPNY,
     +SNCX(1000),SNCY(1000),SNCZ(1000),SENODE(301)
      INTEGER YINC,SNALL(1000),C20,NPX,NPY,NEX,NEY,SNTOT,SNTFLAT
      CHARACTER*(3)UPRE
C
      OPEN(20,FILE='nodedata.dat')
C
      NNXMAX=C20*NPX*NEX+1
      NONX=NNXMAX
      NONY=C20*NPY*NEY+1
      NIPPX=C20*NEX
      NBEPP=2*NEX+2*NEY
      NSEPP=NEX*NEY
      LPNX=LTOTX/(NONX-1)
      LPNY=LTOTY/(NONY-1)
      NINCY=NNXMAX
      NNYMAX=C20*NPY*NEY+1
      NTOTAL=NNXMAX*NNYMAX
      NIPPY=C20*NEY*NINCY
       WRITE(10,30)NNXMAX,NNYMAX,NONX,NONY,NIPPX,NIPPY,
     + NBEPP,NSEPP,LPNX,LPNY
30     FORMAT('NNXMAX = ',I5,/,'NNYMAX = ',I5,/,
     + 'NONX = ',I5,/,'NONY = ',I5,/,'NIPPX = ',I5,/,'NIPPY = ',I5,
     + /,'NBEPP = ',I5,/,'NSEPP = ',I5,/,'LPNX = ',F10.4,/,
     + 'LPNY = ',F10.4)
C
C                 generate nodes in one plane
C
      NCOUNT=0
      YINC=1
       DO 120 J=1,NONY
       DO 100 I=1,NONX
       K=1
         IF(J.EQ.1)THEN
          NCOUNT=NCOUNT+1
          SNALL(NCOUNT)=NCOUNT
          SNCX(NCOUNT)=0.0+LPNX*(I-1)
          SNCY(NCOUNT)=0.0
          SNCZ(NCOUNT)=0.0
          WRITE(20,101)SNALL(NCOUNT),I,J,NCOUNT,YINC,SNCX(NCOUNT),
     +    SNCY(NCOUNT),SNCZ(NCOUNT)
101       FORMAT('SNALL(NCOUNT) = ',I4,'  I = ',I4,'  J = ',I4,
```

```
     +     ' NCOUNT = ',I4,'  YINC = ',I4,'  SNCX(NCOUNT) = ',F10.4,
     +     ' SNCY(NCOUNT) = ',F10.4,'   SNCZ(NCOUNT) = ',F10.4)
           GOTO 100
          ELSE
           NCOUNT=NCOUNT+1
           SNALL(NCOUNT)=NCOUNT
           SNCX(NCOUNT)=0.0+LPNX*(I-1)
           SNCY(NCOUNT)=SNCY(NCOUNT)+(J-1)*LPNY
           SNCZ(NCOUNT)=0.0
           WRITE(20,102)SNALL(NCOUNT),I,J,NCOUNT,YINC,SNCX(NCOUNT),
     +     SNCY(NCOUNT),SNCZ(NCOUNT)
102        FORMAT('SNALL(NCOUNT) = ',I4,'  I = ',I4,'  J = ',I4,
     +     ' NCOUNT = ',I4,'  YINC = ',I4,'  SNCX(NCOUNT) = ',F10.4,
     +     ' SNCY(NCOUNT) = ',F10.4,'   SNCZ(NCOUNT) = ',F10.4)
          ENDIF
100     CONTINUE
        YINC=NINCY
120    CONTINUE
       SNTFLAT=NCOUNT
C
C          generate nodes in second plane for upright eccentricity
C
       IF(UPRE.EQ.'ECC')THEN
       YINC=1
        DO 115 J=1,NONY
         NUPRC=0
        DO 110 I=1,NONX,C20*NEX
           NUPRC=NUPRC+1
          IF(J.EQ.1)THEN
           NCOUNT=NCOUNT+1
           SNALL(NCOUNT)=NCOUNT
           SNCX(NCOUNT)=0.0+LPNX*(I-1)
           SNCY(NCOUNT)=0.0
           SNCZ(NCOUNT)=SENODE(NUPRC)
           WRITE(20,105)SNALL(NCOUNT),I,J,NCOUNT,YINC,SNCX(NCOUNT),
     +     SNCY(NCOUNT),SNCZ(NCOUNT)
105        FORMAT('SNALL(NCOUNT) = ',I4,'  I = ',I4,'  J = ',I4,
     +     ' NCOUNT = ',I4,'  YINC = ',I4,'  SNCX(NCOUNT) = ',F10.4,
     +     ' SNCY(NCOUNT) = ',F10.4,'   SNCZ(NCOUNT) = ',F10.4)
           GOTO 110
          ELSE
           NCOUNT=NCOUNT+1
           SNALL(NCOUNT)=NCOUNT
           SNCX(NCOUNT)=0.0+LPNX*(I-1)
           SNCY(NCOUNT)=SNCY(NCOUNT)+(J-1)*LPNY
```

```
          SNCZ(NCOUNT)=SENODE(NUPRC)
          WRITE(20,106)SNALL(NCOUNT),I,J,NCOUNT,YINC,SNCX(NCOUNT),
     +    SNCY(NCOUNT),SNCZ(NCOUNT)
106       FORMAT('SNALL(NCOUNT) = ',I4,'  I = ',I4,'  J = ',I4,
     +    '  NCOUNT = ',I4,'  YINC = ',I4,'  SNCX(NCOUNT) = ',F10.4,
     +    '  SNCY(NCOUNT) = ',F10.4,'   SNCZ(NCOUNT) = ',F10.4)
         ENDIF
110     CONTINUE
         YINC=NINCY
115     CONTINUE
        SNTOT=NCOUNT
        PRINT *,'SNTOT = ',SNTOT
       ELSE
        SNTOT=NCOUNT
        PRINT *,'SNTOT = ',SNTOT
       ENDIF
C
        CLOSE(20)
        RETURN
        END




C******************************************************************
C
       SUBROUTINE ELEMENTS(NPX,NPY,NEX,NEY,C20,UPRE,NALL,
     +SELS,SELBH,SELBV,SELBVN,SELBHN,SELSN,SPNLSE,
     +ICSS,ICBHS,ICBVS,NTFLAT)
C
C******************************************************************
C           Subroutine to generate all elements for the
C           finite element model
C******************************************************************
C
       INTEGER NALL(1000),NPX,NPY,NEX,NEY,C20,NNXMAX,
     +SELS(1000,1000),SELBH(1000,1000),SELBV(1000,1000),
     +SELBVN(1000),SELBHN(1000),SELSN(1000),SPNLSE(1000,20),
     +ICSS,ICBHS,ICBVS,NTFLAT
       CHARACTER*(3) UPRE
       PRINT *,'NPX =',NPX
       PRINT *,'NPY =',NPY
       PRINT *,'NEX =',NEX
       PRINT *,'NEY =',NEY
```

```
      PRINT *,'C20 =',C20
      PRINT *,'UPRE =',UPRE
C
      OPEN(20,FILE='element-data.dat')
C
      NNXMAX=C20*NPX*NEX+1
      NONX=NNXMAX
      NONY=C20*NPY*NEY+1
C
C           generate the shell elements
C
      ICSS=0
      NINC=0
        DO 169 I=1,NEY*NPY,1
        NINC=(I-1)*NONX*C20
        DO 171 NCOUNT=1+NINC,(NONX-C20)+NINC,C20
          ICSS=ICSS+1
          SELSN(ICSS)=ICSS
          SELS(ICSS,1)=NALL(NCOUNT)
          SELS(ICSS,2)=NALL(NCOUNT+C20)
          SELS(ICSS,3)=NALL(NCOUNT+NONX*C20+C20)
          SELS(ICSS,4)=NALL(NCOUNT+NONX*C20)
          SELS(ICSS,5)=NALL(NCOUNT+C20/2)
          SELS(ICSS,6)=NALL(NCOUNT+NONX+C20)
          SELS(ICSS,7)=NALL(NCOUNT+NONX*C20+C20/2)
          SELS(ICSS,8)=NALL(NCOUNT+NONX)
        WRITE(20,181)ICSS,SELS(ICSS,1),ICSS,SELS(ICSS,2),ICSS,
     +  SELS(ICSS,3),
     +  ICSS,SELS(ICSS,4),ICSS,SELS(ICSS,5),ICSS,SELS(ICSS,6),ICSS,
     +  SELS(ICSS,7),ICSS,SELS(ICSS,8),I
181     FORMAT('SELS(',I3,',1) = ',I5,'  SELS(',I3,',2) = ',I5,
     +  '  SELS(',I3,',3) = ',I5,'  SELS(',I3,',4) = ',I5,/,
     +  'SELS(',I3,',5) = ',I5,'  SELS(',I3,',6) = ',I5,
     +  '  SELS(',I3,',7) = ',I5,'  SELS(',I3,',8) = ',I5,/,
     +  'I = ',I3)
171     CONTINUE
169     CONTINUE
C
C           create element sets (shells) related to panels/sheets
C
      IC4=0
      IC5=0
      WRITE(20,*)'  '
      DO 530 IL=1,NEY
      IC3=0
```

```
      DO 505 I=1,NEY*NPY,NEY
       DO 500 J=1,NEX*NPX,NEX
         IC3=IC3+1
         DO 510 IK=1,NEX
          IC4=IK+NEX*(IL-1)
          IC5=IC5+1
          SPNLSE(IC3,IC4)=SELSN(IC5)
          WRITE(20,520)IC3,IC4,SPNLSE(IC3,IC4)
520       FORMAT(1X,'SPNLSE(',I2,',',I2,') = ',I3)
510      CONTINUE
500    CONTINUE
       IC5=IC5+NEX*NPX*(NEY-1)
505   CONTINUE
      IC5=NEX*NPX*IL
530   CONTINUE
C
C           generate the beam elements - horizontal
C
       NINC=0
       ICBHS=0
       ICB=ICSS
       DO 194 I=1,NPY+1,1
       NINC=(I-1)*NONX*C20*NEY
       DO 192 NCOUNT=1+NINC,NONX-C20+NINC,C20
        ICB=ICB+1
        ICBHS=ICBHS+1
          SELBHN(ICBHS)=ICB
          SELBH(ICBHS,1)=NALL(NCOUNT)
          SELBH(ICBHS,2)=NALL(NCOUNT+C20/2)
          SELBH(ICBHS,3)=NALL(NCOUNT+C20)
          WRITE(20,191)ICBHS,SELBHN(ICBHS),ICBHS,SELBH(ICBHS,1),ICBHS,
     +    SELBH(ICBHS,2),ICBHS,SELBH(ICBHS,3),I,J
191       FORMAT('SELBHN(',I3,') = ',I5,'  SELBH(',I3,',1) = ',I5,
     +    '  SELBH(',I3,',2) = ',I5,'  SELBH(',I3,',3) = ',I5,/,
     +    'I = ',I5,'  J = ',I5)
192    CONTINUE
194    CONTINUE
C
C           generate the beam elements - vertical - no eccentricity
C
      IF(UPRE.EQ.'NOE')THEN
      ICBVS=0
      NINC=0
      DO 210 I=1,NPX+1,1
       NINC=(I-1)*NEX*C20
```

```
      DO 220 NCOUNT=1+NINC,NONX*NEY*C20+NINC,NONX*C20
        ICB=ICB+1
        ICBVS=ICBVS+1
          SELBVN(ICBVS)=ICB
          SELBV(ICBVS,1)=NALL(NCOUNT)
          SELBV(ICBVS,2)=NALL(NCOUNT+NONX)
          SELBV(ICBVS,3)=NALL(NCOUNT+NONX*C20)
          WRITE(20,211)ICBVS,SELBVN(ICBVS),ICBVS,SELBV(ICBVS,1),ICBVS,
     +    SELBV(ICBVS,2),ICBVS,SELBV(ICBVS,3),I,J,ICB
211       FORMAT('SELBVN(',I3,') = ',I5,'  SELBV(',I3,',1) = ',I5,
     +    '  SELBV(',I3,',2) = ',I5,'  SELBV(',I3,',3) = ',I5,/,
     +    'I = ',I5,'  J = ',I5,'  ICB = ',I5)
220   CONTINUE
210   CONTINUE
C
C           generate the beam elements - vertical - with eccentricity
C
      ELSE IF(UPRE.EQ.'ECC')THEN
      ICBVS=0
      NINC=0
      DO 240 I=1,NPX+1,1
       NINC=I-1
      DO 235 NCOUNT=NTFLAT+1+NINC,NTFLAT+NPY*NEY*C20*(NPX+1)+NINC,
     +(NPX+1)*C20
        ICB=ICB+1
        ICBVS=ICBVS+1
          SELBVN(ICBVS)=ICB
          SELBV(ICBVS,1)=NALL(NCOUNT)
          SELBV(ICBVS,2)=NALL(NCOUNT+NPX+1)
          SELBV(ICBVS,3)=NALL(NCOUNT+(NPX+1)*C20)
          WRITE(20,230)ICBVS,SELBVN(ICBVS),ICBVS,SELBV(ICBVS,1),ICBVS,
     +    SELBV(ICBVS,2),ICBVS,SELBV(ICBVS,3),I,J,ICB
230       FORMAT('SELBVN(',I3,') = ',I5,'  SELBV(',I3,',1) = ',I5,
     +    '  SELBV(',I3,',2) = ',I5,'  SELBV(',I3,',3) = ',I5,/,
     +    'I = ',I5,'  J = ',I5,'  ICB = ',I5)
235   CONTINUE
240   CONTINUE
      ENDIF
C
      CLOSE(20)
C
      RETURN
      END
```

```
C************************************************************
C
      SUBROUTINE FEMINP(NPX,NPY,NEX,NEY,C20,ICS,ICBH,ICBV,
     +NTFLAT,NTOT,PNLSE,ITN,UPRE,T,SXPS,SYPS,SXYPS,NALL,
     +NCX,NCY,NCZ,ELS,ELSN,ELBHN,ELBH,ELBVN,ELBV)
C
C************************************************************
C          Subroutine to write the ABAQUS input deck
C************************************************************
C
      DOUBLE PRECISION SXPS(301),SYPS(301),SXYPS(301),
     +NCX(1000),NCY(1000),NCZ(1000),T(301)
      INTEGER ITN,ELS(1000,1000),ELSN(1000),ELBHN(1000),
     +ELBH(1000,1000),ELBVN(1000),ELBV(1000,1000),
     +NALL(1000),PNLSE(1000,20),C20
      CHARACTER*(3)UPRE
      CHARACTER*(1)COMMA
C
      OPEN(30,FILE='femmodel.inp')
C
      COMMA=','
      NP=NPX*NPY
      NSEPP=NEX*NEY
      NONX=C20*NPX*NEX+1
      NONY=C20*NPY*NEY+1
C
      WRITE(30,700)
700   FORMAT('*PREPRINT,HISTORY=NO,ECHO=NO,CONTACT=NO',/,
     +'*RESTART,WRITE,FREQ=1')
      IF(ITN.GE.2)THEN
      WRITE(30,705)
705   FORMAT('*INITIAL CONDITIONS,TYPE=STRESS')
      DO 703 I=1,NP
      WRITE(30,704)I,COMMA,SXPS(I),COMMA,SYPS(I),COMMA,
     +SXYPS(I),COMMA
704   FORMAT('E-PL',I3,A1,G20.12,A1,G20.12,A1,G20.12,A1)
703   CONTINUE
      ENDIF
C
C          write nodal coordinates (includes eccentricity nodes)
C
      WRITE(30,706)
706   FORMAT('*NODE,NSET=N-ALL')
```

```
        DO 266 NCOUNT=1,NTOT
          WRITE(30,FMT=710)NALL(NCOUNT),COMMA,NCX(NCOUNT),COMMA,
     +    NCY(NCOUNT),COMMA,NCZ(NCOUNT)
710       FORMAT(I7,A1,G15.8,A1,G15.8,A1,G15.8)
266     CONTINUE
C
C          generate node sets for BEAM MPC
C
      NINC=0
      WRITE(30,761)
761   FORMAT('*NSET,NSET=N-SUPR')
      DO 764 I=1,NONY,1
       NINC=(I-1)*NONX
      DO 771 NCOUNT=1+NEX*C20+NINC,NONX+NINC,NEX*C20
          WRITE(30,765)NALL(NCOUNT),COMMA
765       FORMAT(I7,A1)
771     CONTINUE
764     CONTINUE
      NINC=0
      WRITE(30,781)
781   FORMAT('*NSET,NSET=N-ECC')
        DO 784 I=1,NONY,1
         NINC=(I-1)*(NPX+1)
        DO 783 NCOUNT=NTFLAT+2+NINC,NTFLAT+NPX+1+NINC,1
        WRITE(30,788)NALL(NCOUNT),COMMA
788       FORMAT(I7,A1)
783     CONTINUE
784     CONTINUE
C
C          write shell elements
C
      WRITE(30,720)
720   FORMAT('*ELEMENT,TYPE=S8R5,ELSET=E-MEMB')
       DO 280 J=1,ICS
         WRITE(30,721)ELSN(J),COMMA,ELS(J,1),COMMA,ELS(J,2),
     +   COMMA,ELS(J,3),COMMA,ELS(J,4),COMMA,ELS(J,5),COMMA,
     +   ELS(J,6),COMMA,ELS(J,7),COMMA,ELS(J,8)
721      FORMAT(I6,A1,I6,A1,I6,A1,I6,A1,I6,A1,I6,A1,I6,A1,I6,A1,I6)
280    CONTINUE
C
C          generate seperate element sets for each panel (shell elements only)
C          seperate panel thicknesses
C          - max no of panels:  999
C
      DO 725 I=1,NP
```

```
      IF(I.LE.9)THEN
       WRITE(30,754)I
754    FORMAT('*ELSET,ELSET=E-PL',I1)
      ELSE IF(I.GE.10.AND.I.LE.99)THEN
       WRITE(30,752)I
752    FORMAT('*ELSET,ELSET=E-PL',I2)
      ELSE
       WRITE(30,753)I
753    FORMAT('*ELSET,ELSET=E-PL',I3)
      ENDIF
       DO 727 J=1,NSEPP
        WRITE(30,728)PNLSE(I,J),COMMA
728     FORMAT(I7,A1)
727    CONTINUE
725   CONTINUE
C
C         create beam elements
C
      WRITE(30,730)
730   FORMAT('*ELEMENT,TYPE=B32,ELSET=E-BHORI')
        DO 290 J=1,ICBH
          WRITE(30,FMT=731)ELBHN(J),COMMA,ELBH(J,1),COMMA,
     +    ELBH(J,2),COMMA,ELBH(J,3)
731       FORMAT(I6,A1,I6,A1,I6,A1,I6)
290     CONTINUE
      WRITE(30,740)
740   FORMAT('*ELEMENT,TYPE=B32,ELSET=E-BVERT')
        DO 300 J=1,ICBV
          WRITE(30,FMT=741)ELBVN(J),COMMA,ELBV(J,1),COMMA,
     +    ELBV(J,2),COMMA,ELBV(J,3)
741       FORMAT(I6,A1,I6,A1,I6,A1,I6)
300     CONTINUE
C
C         create seperate element sets for each upright so that
C         max no of upright sets = 999
C
      BVTOT=0
      NUPRS=ICBV/NEY
      WRITE(10,801)NUPRS
801   FORMAT('NUPRS = ',I3)
      DO 800 I=1,NUPRS
      IF(I.LE.9)THEN
       WRITE(30,810)I
810    FORMAT('*ELSET,ELSET=E-UP',I1)
      ELSE IF(I.GE.10.AND.I.LE.99)THEN
```

```
       WRITE(30,811)I
811    FORMAT('*ELSET,ELSET=E-UP',I2)
      ELSE
       WRITE(30,812)I
812    FORMAT('*ELSET,ELSET=E-UP',I3)
      ENDIF
        DO 805 J=1,NEY
         BVTOT=BVTOT+1
         WRITE(30,815)ELBVN(BVTOT),COMMA
815      FORMAT(I6,A1)
805     CONTINUE
800   CONTINUE
C
      WRITE(30,808)
808   FORMAT('*NSET,NSET=N-ENDC,ELSET=E-UP7')
C
C          create seperate element sets for each part of the flange
C          between two uprights; max no of flange sets = 999
C
      BHTOT=0
      NFLS=ICBH/NEX
      WRITE(10,840)NFLS
840   FORMAT('NFLS = ',I3)
      DO 850 I=1,NFLS
      IF(I.LE.9)THEN
       WRITE(30,845)I
845    FORMAT('*ELSET,ELSET=E-FL',I1)
      ELSE IF(I.GE.10.AND.I.LE.99)THEN
       WRITE(30,846)I
846    FORMAT('*ELSET,ELSET=E-FL',I2)
      ELSE
       WRITE(30,847)I
847    FORMAT('*ELSET,ELSET=E-FL',I3)
      ENDIF
        DO 855 J=1,NEX
         BHTOT=BHTOT+1
         WRITE(30,835)ELBHN(BHTOT),COMMA
835      FORMAT(I6,A1)
855     CONTINUE
850   CONTINUE
C
C          create seperate element sets for the upper and lower flange
C
      WRITE(30,742)
742   FORMAT('*ELSET,ELSET=E-BH1')
```

```
       DO 743 I=1,NEX*NPX
        WRITE(30,745)ELBHN(I),COMMA
745     FORMAT(I6,A1)
743    CONTINUE
      WRITE(30,746)
746   FORMAT('*ELSET,ELSET=E-BH2')
       IF(NEX*NPX.EQ.1)THEN
       IY=2
      ELSE
      IY=NEX*NPX+1
       ENDIF
       DO 747 J=IY,NEX*NPX*(NPY+1)
        WRITE(30,748)ELBHN(J),COMMA
748     FORMAT(I6,A1)
747    CONTINUE
      WRITE(30,813)
813   FORMAT('*NSET,NSET=N-UPFL,ELSET=E-BH2',/,
     +'*NSET,NSET=N-LOWFL,ELSET=E-BH1')
C
C          write shell section properties for all shells
C
      DO 880 I=1,NP
      IF(I.LE.9)THEN
       WRITE(30,882)I,T(I),COMMA
882    FORMAT('*SHELL SECTION,ELSET=E-PL',I1,',MATERIAL=ALU-S',
     +/,G12.7,A1)
      ELSE IF(I.GE.10.AND.I.LE.99)THEN
       WRITE(30,884)I,T(I),COMMA
884    FORMAT('*SHELL SECTION,ELSET=E-PL',I2,',MATERIAL=ALU-S',
     +/,G12.7,A1)
      ELSE
       WRITE(30,886)I,T(I),COMMA
886    FORMAT('*SHELL SECTION,ELSET=E-PL',I3,',MATERIAL=ALU-S',
     +/,G12.7,A1)
      ENDIF
880   CONTINUE
C
      WRITE(30,755)
755   FORMAT('*BEAM SECTION,ELSET=E-BH1,MATERIAL=ALU-B,SECTION=I',/,
     +'0.0055,0.02937,0.0571,0.0,0.00318,0.0,0.00238',/,
     +'0.0,0.0,-1.0',/,
     +'*BEAM SECTION,ELSET=E-BH2,MATERIAL=ALU-B,SECTION=I',/,
     +'0.02952,0.0381,0.0,0.05874,0.0,0.00519,0.00397',/,
     +'0.0,0.0,-1.0',/,
     +'*BEAM SECTION,ELSET=E-BVERT,MATERIAL=ALU-B,SECTION=L',/,
```

```
     +'0.0254,0.0254,0.003175,0.003175',/,
     +'0.0,0.0,-1.0',/,
     +'*MATERIAL,NAME=ALU-B',/,'*ELASTIC',/,'71.0E9,0.3',/,
     +'*MATERIAL,NAME=ALU-S',/,'*ELASTIC',/,'72.4E9,0.3',/,
     +'*BOUNDARY',/,'1,1,6',/,'38,1,6',/,'75,1,6',/,
     +'112,1,6',/,'149,1,6',/,'186,1,6',/,'223,1,6')
C
      IF(UPRE.EQ.'ECC')THEN
       WRITE(30,FMT=759)
759    FORMAT('*MPC',/,'BEAM,N-SUPR,N-ECC')
      ENDIF
C
      WRITE(30,760)
760   FORMAT('*STEP',/,'*STATIC')
C
      WRITE(30,FMT=762)
C
762   FORMAT('*CLOAD',/,'37,2,8578.3',/,'74,2,8578.3',/,
     +'111,2,8578.3',/,'148,2,8578.3',/,'185,2,8578.3',/,
     +'222,2,8578.3',/,'259,2,8578.3')
C
      WRITE(30,FMT=770)
770   FORMAT('*EL FILE',/,'S,SINV',/,'*NODE FILE',/,'COORD')
      WRITE(30,FMT=780)
780   FORMAT('*EL PRINT,TOTALS=YES',/,'S,MISES',/,'SP',/,'*END STEP')
      ENDFILE(UNIT=30)
C
      CLOSE(30)
C
      RETURN
      END
```