2. APDL Input Scripts

Principal Angle Script - run after solution,

```
! - - -
/POST1
SET,1

! defining all my nodes into a subset, total nodes
ALLSEL,ALL
NSEL,S,,,ALL
CM,totalnodes,node

! defining total number of nodes
*GET,totalnodes_count,NODE,0,COUNT

! List nodes into table
*DIM,LIST,ARRAY,totalnodes_count,1
*VGET, LIST(1), NODE, , NLIST

! define table of stress for each node
PRNSOL,S
*DIM,TABLES,ARRAY,totalnodes_count,4
*DO,I,1,totalnodes_count
   *GET,TABLES(I,1),NODE,LIST(I),S,X
   *GET,TABLES(I,2),NODE,LIST(I),S,Y
   *GET,TABLES(I,3),NODE,LIST(I),S,XY
   *AFUN,DEG
   r = TABLES(I,3)
   q = 0.5*(TABLES(I,1)-TABLES(I,2))
   *IF,q,EQ,0,THEN
      q = 0.001
   *ENDIF
   angle = 0.5*ATAN2(r,q)
   TABLES(I,4) = ABS(angle)
*ENDDO
```

```
! use *VEDIT,TABLES to see table of stresses

! PLOTTING

/POST1

/contour,,10, 2,10,88     ! NCONT=10, VMIN=2, VMAX=90 (uniform spacing)
/dscale,,off
SET,LAST
/GRAPHICS,FULL

*DO,I,1,totalnodes_count
    DNSOL,LIST(I),U,X,TABLES(I,4)
*ENDDO

RAPPND,1,1

PLNSOL, U,X, 2,88
! - - -
```

StrainSpace_AllPlane_Validity_Bidirectional_Fibre - run after solution,

```
! - - STRESS STATE LEGEND - -
! PRINCIPAL EVERYWHERE
! XY - YELLOW
! YZ - GREEN
! XZ - TEAL
! MULTIPLE - RED
! - - - - - - - - - - - - - -

/POST1
SET,1

! defining all my nodes into a subset, total nodes
ALLSEL,ALL
NSEL,S,,,ALL
CM,totalnodes,node

! defining total number of nodes
*GET,totalnodes_count,NODE,0,COUNT

! List nodes into table
*DIM,LIST,ARRAY,totalnodes_count,1
```

```
*VGET, LIST(1), NODE, , NLIST

! define table of stress for each node
PRNSOL,S
*DIM,TABLES,ARRAY,totalnodes_count,6
*DO,I,1,totalnodes_count
    *GET,TABLES(I,1),NODE,LIST(I),EPTO,X
    *GET,TABLES(I,2),NODE,LIST(I),EPTO,Y
    *GET,TABLES(I,3),NODE,LIST(I),EPTO,Z
    *GET,TABLES(I,4),NODE,LIST(I),EPTO,XY
    *GET,TABLES(I,5),NODE,LIST(I),EPTO,YZ
    *GET,TABLES(I,6),NODE,LIST(I),EPTO,XZ
*ENDDO

! list of angles
*DIM,TABLEA,ARRAY,totalnodes_count,3

! xy angle
*DO,I,1,totalnodes_count
    *AFUN,DEG
    r = TABLES(I,4)/2
    q = 0.5*(TABLES(I,1)-TABLES(I,2))
    *IF,q,EQ,0,THEN
        q = 0.001
    *ENDIF
    angle = 0.5*ATAN2(r,q)
    TABLEA(I,1) = ABS(angle)
*ENDDO

! yz angle
*DO,I,1,totalnodes_count
    *AFUN,DEG
    r = TABLES(I,5)/2
    q = 0.5*(TABLES(I,2)-TABLES(I,3))
    *IF,q,EQ,0,THEN
        q = 0.001
    *ENDIF
    angle = 0.5*ATAN2(r,q)
    TABLEA(I,2) = ABS(angle)
*ENDDO

! xz angle
*DO,I,1,totalnodes_count
    *AFUN,DEG
    r = TABLES(I,6)/2
    q = 0.5*(TABLES(I,1)-TABLES(I,3))
```

```
    *IF,q,EQ,0,THEN
        q = 0.001
    *ENDIF
    angle = 0.5*ATAN2(r,q)
    TABLEA(I,3) = ABS(angle)
*ENDDO

! index
*DIM,INDEX,ARRAY,totalnodes_count,1
*DO,I,1,totalnodes_count
    COUNT_IND = 0
    INDEX(I,1) = 0
    *IF,TABLEA(I,1),GT,2,AND,TABLEA(I,1),LT,88,THEN ! XY - YELLOW
        INDEX(I,1) = 0.8
        COUNT_IND = COUNT_IND + 1
    *ENDIF
    *IF,TABLEA(I,2),GT,2,AND,TABLEA(I,2),LT,88,THEN ! YZ - GREEN
        INDEX(I,1) = 0.55
        COUNT_IND = COUNT_IND + 1
    *ENDIF
    *IF,TABLEA(I,3),GT,2,AND,TABLEA(I,3),LT,88,THEN !XZ - TEAL
        INDEX(I,1) = 0.30
        COUNT_IND = COUNT_IND + 1
    *ENDIF
    *IF,COUNT_IND,GT,2,THEN ! MULTIPLE - RED
        INDEX(I,1) = 1
    *ENDIF
*ENDDO

! use *VEDIT,TABLES to see table of stresses

! PLOTTING

/POST1

/contour,,10, 0,1,1     ! NCONT=10, VMIN=2, VMAX=90 (uniform spacing)
/dscale,,off
SET,LAST
/GRAPHICS,FULL

*DO,I,1,totalnodes_count
    DNSOL,LIST(I),U,X,INDEX(I,1)
*ENDDO

RAPPND,1,1

PLNSOL, U,X, 0,1
```

/GLINE,1,-1

/REPLOT
! - - -

StrainSpace_AllPlane_Validity_Unidirectional_Fibre - run after solution,

! - - -

/POST1
SET,1

! defining all my nodes into a subset, total nodes
ALLSEL,ALL
NSEL,S,,,ALL
CM,totalnodes,node

! defining total number of nodes
*GET,totalnodes_count,NODE,0,COUNT

! List nodes into table
*DIM,LIST,ARRAY,totalnodes_count,1
*VGET, LIST(1), NODE, , NLIST

! define table of stress for each node
PRNSOL,S
*DIM,TABLES,ARRAY,totalnodes_count,6
*DO,I,1,totalnodes_count
    *GET,TABLES(I,1),NODE,LIST(I),EPTO,X
    *GET,TABLES(I,2),NODE,LIST(I),EPTO,Y
    *GET,TABLES(I,3),NODE,LIST(I),EPTO,Z
    *GET,TABLES(I,4),NODE,LIST(I),EPTO,XY
    *GET,TABLES(I,5),NODE,LIST(I),EPTO,YZ
    *GET,TABLES(I,6),NODE,LIST(I),EPTO,XZ
*ENDDO

! list of angles
*DIM,TABLEA,ARRAY,totalnodes_count,3

! xy angle
*DO,I,1,totalnodes_count

```
    *AFUN,DEG
    r = TABLES(I,4)/2
    q = 0.5*(TABLES(I,1)-TABLES(I,2))
    *IF,q,EQ,0,THEN
        q = 0.001
    *ENDIF
    angle = 0.5*ATAN2(r,q)
    TABLEA(I,1) = ABS(angle)
*ENDDO

! yz angle
*DO,I,1,totalnodes_count
    *AFUN,DEG
    r = TABLES(I,5)/2
    q = 0.5*(TABLES(I,2)-TABLES(I,3))
    *IF,q,EQ,0,THEN
        q = 0.001
    *ENDIF
    angle = 0.5*ATAN2(r,q)
    TABLEA(I,2) = ABS(angle)
*ENDDO

! xz angle
*DO,I,1,totalnodes_count
    *AFUN,DEG
    r = TABLES(I,6)/2
    q = 0.5*(TABLES(I,1)-TABLES(I,3))
    *IF,q,EQ,0,THEN
        q = 0.001
    *ENDIF
    angle = 0.5*ATAN2(r,q)
    TABLEA(I,3) = ABS(angle)
*ENDDO

! index
*DIM,INDEX,ARRAY,totalnodes_count,1
*DO,I,1,totalnodes_count
    COUNT_IND = 0
    INDEX(I,1) = 0
    *IF,TABLEA(I,1),GT,2,THEN ! XY
        INDEX(I,1) = 0.8
        COUNT_IND = COUNT_IND + 1
    *ENDIF
    *IF,TABLEA(I,2),GT,2,THEN ! YZ
        INDEX(I,1) = 0.55
        COUNT_IND = COUNT_IND + 1
```

```
   *ENDIF
   *IF,TABLEA(I,3),GT,2,THEN !XZ
      INDEX(I,1) = 0.30
      COUNT_IND = COUNT_IND + 1
   *ENDIF
   *IF,COUNT_IND,GT,2,THEN
      INDEX(I,1) = 1
   *ENDIF
*ENDDO

! use *VEDIT,TABLES to see table of stresses

! PLOTTING

/POST1

/contour,,10, 0,1,1     ! NCONT=10, VMIN=2, VMAX=90 (uniform spacing)
/dscale,,off
SET,LAST
/GRAPHICS,FULL

*DO,I,1,totalnodes_count
   DNSOL,LIST(I),U,X,INDEX(I,1)
*ENDDO

RAPPND,1,1

PLNSOL, U,X, 0,1

/GLINE,1,-1

/REPLOT
```

U_Star_Load_Path_Analysis- run after ajusting "***" indicated lines,

```
/CLE
*DEL,ALL
! Analyses input starts here
/NOPR   ! Suppress printing of UNDO process
FINISH  ! Make sure we are at BEGIN level
/PREP7 ! Enter the preprocessor
```

```
/OUTPUT,debug,txt   ! send all messages to debug.txt (optional)

file_name = 'FILENAUM' ! Change the
filename*************************************************************************
NODES_PER_ELEMENT = 20
DOF_PER_NODE = 2 ! 2 - 2D problem ; 3 - 3D problem
!read in the file. the file should contain nodal entity setsl

CDREAD,DB,'C:\Users\bragg\OneDrive\Desktop\Load_Path\RedoOfCode\Load_Path-
main\g23plate_fixed',cdb ! ******************************************
EPLOT

!-----------COOKING------------
! SELECT BC
NSEL,U,,,ALL
NSEL,S,D,U,0
CM,fixnodes,NODE
! SELECT FORCES
NSEL,U,,,ALL
NSEL,S,F,F,-1000 ! SAY WHAT MAGNITUDE OF LOAD YOU ARE
INPUTTING**********************************
CM,LOADNODES,NODE

!---------------------- GET FREENODES & TOTALNODES ----------------------
ALLSELL,ALL
nsel,all
nsel,u,,,fixnodes
nsel,u,,,loadnodes
CM,freenodes,node
nsel,all
cm,totalnodes,node
!----------------------- Solve the problem ----------------------
/SOLU            ! Enter the solution
SOLVE

!---------------------- COUNT THE NODES ----------------------
NSEL,S,,,fixnodes
*GET,fixnodes_count,NODE,0,COUNT
NSEL,S,,,loadnodes
*GET,loadnodes_count,NODE,0,COUNT
NSEL,S,,,freenodes
*GET,freenodes_count,NODE,0,COUNT
NSEL,S,,,totalnodes
*GET,totalnodes_count,NODE,0,COUNT
```

```
!---------------------- GET THE NODE NUMBERS ----------------------

NSEL,S,,,loadnodes
*VGET, loadnodes_no, NODE, 0, nlist
NSEL,S,,,fixnodes
*VGET, fixnodes_no, NODE, 0, nlist
NSEL,S,,,freenodes
*VGET, freenodes_no, NODE, 0, nlist

ALLSEL
NSEL,S,,,freenodes            ! select freenodes
*GET,min_node_no,NODE,0,NUM,MIN        ! Get and assign the least node number of freenodes in
the varable "min_node_no"

*DIM,loaded_nodes_lc,ARRAY,2,freenodes_count

!---------------- Apply original boundary conditions ----------------

/PREP7          ! Enter the preprocessor
ALLSEL
D,fixnodes,ALL,0      ! All dofs of the fix nodes to zero (Change according to the problem)
F,loadnodes,Fy,-1000    ! Apply force (Change according to the
problem)************************************


!---------------------- Calculate the strain Energy ----------------------
/POST1
SET,
/POST26
ENERSOL,2,SENE,,STRAINENERGY

FILE,'%file_name%','rst','.'
/UI,COLL,1
NUMVAR,200
SOLU,191,NCMIT
STORE,MERGE
FILLDATA,191,,,,1,1
REALVAR,191,191

*GET,U_VALUE,VARI,2, REAL,1  ! Store the Strain energy value

!---------------------- Calculate the load node displacement ----------------------

*DIM, LOADNODE_DISP,ARRAY,DOF_PER_NODE,loadnodes_count

*IF,DOF_PER_NODE,EQ,2,THEN
    *DO, I, 1,loadnodes_count
```

```
        *GET,LOADNODE_DISP(1,I),NODE,loadnodes_no(I),U,X
        *GET,LOADNODE_DISP(2,I),NODE,loadnodes_no(I),U,Y
   *ENDDO
*ELSEIF,DOF_PER_NODE,EQ,3,THEN
   *DO, I, 1,loadnodes_count
        *GET,LOADNODE_DISP(1,I),NODE,loadnodes_no(I),U,X
        *GET,LOADNODE_DISP(2,I),NODE,loadnodes_no(I),U,Y
        *GET,LOADNODE_DISP(3,I),NODE,loadnodes_no(I),U,Z
   *ENDDO
*ENDIF


!---------------------- Do the looping ----------------------

/PREP7 ! Enter the processorer

*DO,I,1,freenodes_count
   DDELE,ALL
   FDELE,ALL
   ALLSEL
   !Apply Normal boundary Condition
   D,fixnodes,ALL,0              ! All dofs of the fix nodes to zero
   *IF,DOF_PER_NODE,EQ,2,THEN
      *DO,J,1,loadnodes_count
           D,loadnodes_no(J),UX,LOADNODE_DISP(1,J)! assigning enforced X-Displacement at the
load nodes
           D,loadnodes_no(J),UY,LOADNODE_DISP(2,J)! assigning enforced Y-displacement at the
load nodes
      *ENDDO
   *ELSEIF,DOF_PER_NODE,EQ,3,THEN
      *DO,J,1,loadnodes_count
           D,loadnodes_no(J),UX,LOADNODE_DISP(1,J)! assigning enforced X-Displacement at the
load nodes
           D,loadnodes_no(J),UY,LOADNODE_DISP(2,J)! assigning enforced Y-displacement at the
load nodes
           D,loadnodes_no(J),UZ,LOADNODE_DISP(3,J)! assigning enforced Z-displacement at the
load nodes
      *ENDDO
   *ENDIF
   NSEL,S,,,freenodes
   loaded_nodes_lc(1,I)=min_node_no
   min_node_no=NDNEXT(min_node_no)     ! min_node_no(i)= min_node_no(i+1)
   D,loaded_nodes_lc(1,I),ALL,ALL
   ALLSEL
   LSWRITE,I             ! Write loadstep
   DDELE,ALL              ! delete all constrained dofs
```

```
*ENDDO
DDELE,ALL



!-------------------- SOLVE ALL THE SETS -------------------------
*GET,TBEFORE,ACTIVE,,TIME,CPU
/SOLU
ALLSEL
LSSOLVE,1,freenodes_count
*GET,TAFTER,ACTIVE,,TIME,CPU
SOLUTION_TIME = (TAFTER-TBEFORE) ! record the computational CPU time


!-------------------- CALCULATE STRAIN ENERGY FOR ALL SET ----------------------

/POST26
! output needs to be redirected to get the list of energies over all load steps
/OUTPUT,energy_list,txt
ENERSOL,3,SENE,,U_PRIME
!PRVAR,3
/OUTPUT          ! send the output back again to usual file


!---------------------- ASSIGN U* VALUES TO THE NODES ----------------------

FILE,'%file_name%','rst','.'
/UI,COLL,1
NUMVAR,200
SOLU,191,NCMIT
STORE,MERGE
FILLDATA,191,,,,1,1
REALVAR,191,191

*DO,J,1,freenodes_count            ! RECORDING U' PRIME VALUES FOR THE FREENODE
   *GET,loaded_nodes_lc(2,J),VARI,3, REAL,J
*ENDDO

*DIM,U_STAR,ARRAY,1,freenodes_count


*DO,K,1,freenodes_count
   U_STAR(1,K)= 1 - (U_VALUE/loaded_nodes_lc(2,K))  ! ORIGINAL RATIO
*ENDDO

!---------------------- PLOTTING ----------------------
```

```
/POST1
SET,LAST
/GRAPHICS,FULL

*DO,I,1,freenodes_count
DNSOL,freenodes_no(I),U,X,U_STAR(1,I)
*ENDDO

*DO,I,1,loadnodes_count
DNSOL,loadnodes_no(I),U,X,1
*ENDDO

*DO,I,1,fixnodes_count
DNSOL,fixnodes_no(I),U,X,0
*ENDDO
RAPPND,freenodes_count+1,freenodes_count+1

PLNSOL, U,X, 0,1.0

!------------------------------ DATA PROCESSING ---------------------


ALLSEL
*GET,ELEM_NO_MIN,ELEM,0,NUM,MIN
*GET,ELEM_NO_MAX,ELEM,0,NUM,MAX
*GET,ELEM_COUNT,ELEM,0,COUNT


ALLSEL
*GET,ELEM_NO_MIN,ELEM,0,NUM,MIN
*GET,ELEM_NO_MAX,ELEM,0,NUM,MAX
*GET,ELEM_COUNT,ELEM,0,COUNT

*DIM,ELEMENT_TABLE,ARRAY,ELEM_COUNT,1+NODES_PER_ELEMENT ! 20 NODES
*DO,I,ELEM_NO_MIN,ELEM_NO_MAX
   *SET,ELEMENT_TABLE(I,1),I
   *DO,J,1,NODES_PER_ELEMENT
      *SET,ELEMENT_TABLE(I,J+1),NELEM(I,J)
   *ENDDO
*ENDDO


*DIM,TABLE_RES,ARRAY,totalnodes_count,5
*IF,DOF_PER_NODE,EQ,2,THEN
   *DO,I,1,totalnodes_count
      *SET,TABLE_RES(I,1),I            ! Node Number
      *GET,TABLE_RES(I,2),NODE,I,LOC,X    ! X- coordinate
```

```
        *GET,TABLE_RES(I,3),NODE,I,LOC,Y    ! Y- coordinate
        *SET,TABLE_RES(I,4),0           ! Z- coordinate
    *ENDDO
*ELSEIF,DOF_PER_NODE,EQ,3,THEN
    *DO,I,1,totalnodes_count
        *SET,TABLE_RES(I,1),I           ! Node Number
        *GET,TABLE_RES(I,2),NODE,I,LOC,X    ! X- coordinate
        *GET,TABLE_RES(I,3),NODE,I,LOC,Y    ! Y- coordinate
        *GET,TABLE_RES(I,4),NODE,I,LOC,Z    ! Z- coordinate
    *ENDDO
*ENDIF


*DO,J,1,freenodes_count
        *SET,TABLE_RES(freenodes_no(J),5),U_STAR(1,J)
*ENDDO

*DO,J,1,loadnodes_count
        *SET,TABLE_RES(loadnodes_no(J),5),1
*ENDDO

*DO,J,1,fixnodes_count
        *SET,TABLE_RES(fixnodes_no(J),5),0
*ENDDO


!-------------------- EXPORTING DATA ----------------------
*cfopen,element_type.txt
! list element types
*GET,net,ETYP,1,NUM,COUNT
*DO,i,1,net
    *GET,ETname,ETYP,i,ATTR,ENAM
*VWRITE,'ET ',i,' ',ETname
(A3, F4.0, A1, F4.0)
*ENDDO
*cfclose

*MWRITE,ELEMENT_TABLE,element_table,txt
(F10.0,F10.0,F10.0,F10.0,F10.0,F10.0,F10.0,F10.0,F10.0,F10.0,F10.0,F10.0,F10.0,F10.0,F10.0,F10.0
,F10.0,F10.0,F10.0,F10.0,F10.0)

*MWRITE,TABLE_RES,result_table,txt
(F10.0,F10.6,F10.6,F10.6,F10.6)
!6 digit precision. if you want more precision, increase'x' in F10.x
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! format output for nodes, elements and stress listings
/PAGE, 1E9,, 1E9,,  ! disable headers
/FORMAT, , ,14,5, , ! fix floating point format
/HEADER, off, off, off, off, on, off ! disable summaries

!/POST26
!/OUTPUT,NLIST,txt
!NLIST,,,,COORD        ! print nodes w. coordinates
!/OUTPUT

!/OUTPUT,ELIST,txt
!ELIST                 ! print element connectivity table
!/OUTPUT


!/POST1
!/OUTPUT,DISP,txt
!PRNSOL,U              ! print all displacements
!/OUTPUT
```

Revision #6
Created 2025-12-05 02:12:12 UTC by Bragg Farmer
Updated 2026-03-10 15:50:33 UTC by Trevor Bendik