Working Wonders with APDL MATH - Ep 04

Author: Nicolas Jobert (Alma Consulting)
Revision: 1.0

Introduction

This document contains the scripts developped for the test case of a simple, symetric, tee-shape junction submitted to a thermal shock.

The code is organized as follows:

1. thermal model setup (element, material propoerties)
2. obtain thermal transient solution
3. compress thermal solution snapshots using SVD, obtain POD vectors
4. store POD vectors into .RTH files
5. obtain structural solution using thermal POD vectors as inputs
6. obtain final structural solution, combining solutions obtained in the previous step

At revision 1.0, the POD reduction is only applied to the structural pass. Tested using ANSYS rev 19.2.

Annexe: ANSYS APDL input files

Step1: thermal FEM setup

```
! MODELE DE PIPE + TEE
! POUR ESTIMATION POD
fini
/clear
resu,Te_48x6,db
/units,SI
nbNodes=ndinqr(0,14)
/title,Cas Test - %nbNodes% noeuds
/gresu,vue_iso.gsav
*use,revVideo.MAC
eplo
/show,png
/replo
/show,close
/PREP7
! SOLID 185 => SOLID70
et,1,SOLID70
cmsel,s,N_INT
cmsel,u,N_INT2
cm,N_INT3,NODE

! Elements pour l'affichage
csys,0
nsel,s,loc,y,-1e3,0
nsel,r,loc,x,0,1e3
esln,s,1
esel,inve
cm,El_DISP,ELEM
eall $ nall

/COM ***************************
/COM *** PROPRIETES MATERIAUX ***
/COM ***************************
mp,ex,1,200e9     ! Pa
mp,alpx,1,15e-6     ! K**-1
mp,kxx,1,15       ! W/m/K
mp,dens,1,7850      ! kg/m3
mp,c,1,500        ! J/kg/K

/show,png,REV
plnsol,eplo
/show,close
*get,currentJobname,ACTIVE,0,jobnam
/RENAME,%currentJobname%000,png,,FEM_Iso,png

save,step1,db
```

step1_modele.inp

```
FINISH
resu,step1,db
GenerationJobname='RUN_THERMAL_FULL'
flag_export_png=1 ! 1=harcopy temperature snapshots to PNG files
/filnam,%GenerationJobname%

!temps_total=1000
!temps_montee=10            ! s
!temps_plateau=temps_total-temps_montee ! s
!temp_ini=20               ! degC
!temp_fin=120            ! degC

! Chargement thermique
!*dim,temp_Fluide,table,3,1,,TIME
!temp_fluide(1,0)=0,temps_montee,temps_montee+temps_plateau
!temp_fluide(1,1)=temp_ini,temp_fin,temp_fin

! Instants de rÃ©initialisation du TS
!*dim,ts_res,array,4,1
!ts_res(1)=2,5,7

*dim,temp_Fluide,table,7,1,,TIME
temp_fluide(1,0)=0,1   ,2      ,5   ,7      ,10  ,50
temp_fluide(1,1)=0,-200  ,-200  ,0    ,-100  ,0    ,0
temp_ini=0
temps_total=50
!
/COM, *** Obtain solution ***
!
/SOLU
/TITLE,FULL THERMAL TRANSIENT SOLVE
ANTYPE,TRANSIENT,NEW !
AUTOTS,ON ! Automatic time stepping
!CNVTOL, Lab, VALUE, TOLER, NORM, MINREF
!CNVTOL,heat,,1e-4,2,1
!CNVTOL,temp,,1e-3,0,1

! pas de temps mini
! cf. Thermal Analysis Guide Â§3.4.3.2
diff=15/(7850*500)   ! m**2/s
l_min=.2e-3       ! m
dt_min=l_min**2/(4*diff)   ! condition suffisante pour Ã©viter les oscillations

!DELTIM,DT_INI,DT_MIN,DT_MAX,CARRY
DELTIM,dt_min,dt_min,5.,ON
!TSRES,%ts_res%

!TINTP, GAMMA, ALPHA, DELTA, THETA   , OSLM, TOL, --, --, AVSMOOTH, ALPHAF,
    ALPHAM
TINTP,       ,       ,       ,1.0     , 0.5
!OPNCONTROL, Lab, VALUE, NUMSTEP
OPNCONTROL,OPENUPFACTOR,2

NROPT,AUTO ! Program-chosen Newton-Raphson option
TUNIF,temp_ini ! Uniform starting temperature at all nodes
TREF,temp_ini
eall
```

```
SF,N_INT2,CONV,25000,%temp_fluide% ! Convection load at all nodes
SF,N_INT3,CONV,1000,%temp_fluide% ! Convection load at all nodes
ALLSEL
/PBC,TEMP,,1 ! Temperature b.c. symbols on
/PSF,CONV,,2 ! Convection symbols on
/TITLE,Boundary conditions
NPLOT

CSYS,0
/COM *** CONTROLE DES SORTIES ***
OUTRES,ALL,NONE
OUTRES,NSOL,ALL

KBC,1 ! STEPPED loading (default)

! On place des points d'arrets
! Ã  des instants clÃ©s du transitoire thermique
time,1
solve

time,2
solve

time,5
solve

time,10
solve

TIME,temps_total
SOLVE
FINISH
save,step2,db


*if,flag_export_png,eq,1,then
! EXPORT TO PNG
/POST1
*use,revVideo.MAC
! Elements pour l'affichage
csys,0
nsel,s,loc,y,-1e3,0
nsel,r,loc,x,0,1e3
esln,s,1
esel,inve
cm,El_DISP,ELEM
eall $ nall

set,last
!*GET,Par, ACTIVE, 0, Item1, IT1NUM, Item2, IT2NUM
*get,nbResults,ACTIVE,0,SET,NSET

! Ici on peut utiliser tout ou partie des rÃ©sultats disponibles
nbSnapshots=nbResults
cmsel,s,EL_DISP
file,%GenerationJobname%,rth
*do,ind_step,1,nbSnapshots
  !setTime=instants_List(ind_step)
```

```
115    set,next
116    *get,currTime,ACTIVE,0,SET,TIME
117    *if,currTime,LE,1,THEN
118    currTime=NINT(currTime*1000)/1000
119    *else
120    currTime=NINT(currTime*10)/10
121    *endif
122
123
124    /title,FULL estimation of temperature @t=%currTime%s
125    /show,png,REV
126    plnsol,temp
127    /show,close
128    *get,currentJobname,ACTIVE,0,jobnam
129    /RENAME,%currentJobname%000,png,,RUN_THERMAL_FULL_%ind_step%,png
130 *enddo
131
132 *endif
```

step2_solve_transitoire.inp

```
fini
resu,step2,db
/filnam,%GenerationJobname%
/POST1
nbNodes=ndinqr(0,13)
maxNode=ndinqr(0,14)
set,last,last

! Get total number of sets in result file
!*GET,Par, ACTIVE, 0, Item1, IT1NUM, Item2, IT2NUM
*get,nbResults,ACTIVE,0,SET,NSET

! Ici on peut utiliser tout ou partie des rÃ©sultats disponibles
nbSnapshots=nbResults

/POST26
solu,2,DTIME
store
/title,Pas de temps solution thermique
plvar,2
*dim,instants_List,array,nbResults
vget,instants_List(1),1
fini


!/POST1
!file,%GenerationJobname%,RTH
!set,list
!*dim,instants_List,array,nbSnapshots
!*do,ind,1,nbSnapshots
!set,1,ind
!*get,toto,ACTIVE,0,SET,TIME
!instants_List(ind)=toto
!*enddo
!fini

! 1a METHODE APDL
!*dim,snapshots,array,MaxNode,nbSnapshots
!*do,ind_snap,1,nbSnapshots
!  set,1,ind_snap
!  *vget,snapshots(1,ind_snap),NODE,1,TEMP
!*enddo

/POST1

! 1b METHOD APDL MATH
/COM Importing snapshot matrix into APDL Math workspace...
*get,currCPUTime,ACTIVE,0,TIME,CPU
time_TIC=currCPUTime
/COM Performing SVD compression...
!*DMAT,Matrix,Type,Method,Val1,Val2,Val3,Val4,Val5
*DMAT,X,D,IMPORT,RST,%GenerationJobname%.RTH,1,nbSnapshots
*get,currCPUTime,ACTIVE,0,TIME,CPU
time_TOC=currCPUTime
CPU_IMPORT=time_TOC-time_TIC


```

```
!/sys,del mySnapShots_%nbNodes%x%nbSnapshots%.txt
*PRINT,X,mySnapShots_%nbNodes%x%nbSnapshots%.txt

! 2. SVD decomposition
! Perform Singular Value Decomposition
*DMAT,U,D,COPY,X

! M=U SIGMA V*
threshold=1e-3
*get,currCPUTime,ACTIVE,0,TIME,CPU $ time_TIC=currCPUTime
*COMPRESS,U,SVD,threshold,SigmaVec,Vconj
*get,currCPUTime,ACTIVE,0,TIME,CPU $ time_TOC=currCPUTime
CPU_SVD=time_TOC-time_TIC

! Number of Singular values above threshold
nbSingularValues=U_coldim

! CHECK: REGENERATE ORIGINAL MATRIX FROM COMPRESSED
! Step1: Generate SigmaMat (diagonal matrix) from vector SigmaVec
*SMAT,SigmaMat,D,ALLOC,DIAG,nbSingularValues
*do,ind,1,nbSingularValues
  SigmaMat(ind,ind)=SigmaVec(ind)
*enddo

! STEP2: GENERATE TRUNCATED RIGHT VECTORS MATRICE
*DMAT,resizedVconj,D,COPY,Vconj
*DMAT,resizedVconj,D,RESIZE,nbSingularValues,nbSnapshots

!*MULT, M1, T1, M2, T2, M3 (M3=M1(T1)*M2(T2))
*MULT,SigmaMat,,Vconj,,SigmaVconj
*MULT,U,,SigmaVconj,,shouldBeX

! ESTIMATE ERROR
*DMAT,errorX,D,COPY,X
*AXPY,-1,0.,ShouldBeX,1.,0.,errorX
! L2 Norm / Linf norm
*NRM,errorX,NRM2,errorX_L2
*NRM,errorX,NRMINF,errorX_Linf

*cfopen,toto,MAC
*vwrite,nbNodes,nbSingularValues
/sys,del myPODLeftVectors_%ix%i.txt
*vwrite,nbSingularValues,nbSnapshots
/sys,del myPODAmplitudes_%ix%i.txt
*cfclose
/inpu,toto.MAC
/sys,del toto.MAC

*PRINT,U,myPODLeftVectors_%nbNodes%x%nbSingularValues%.txt
*PRINT,SigmaVconj,myPODAmplitudes_%nbSingularValues%x%nbSnapshots%.txt

save,step3,db,,MODEL   ! ATTENTION LES OBJETS APDL MATH NE SONT PAS SAUVES
```

step3_reduce_snapshots.inp

```
! CrÃ©e un fichier RTH contenant les champs de tempÃ©rature
! correspondant Ã  chaque vecteur fini
resu,step3,db
/inpu,vue_iso.gsav
!/EXPAND,2,RECT,HALF,,,0.00001
*use,revVideo.MAC

/sys,copy RUN_THERMAL_FULL.RTH RUN_THERMAL_POD_VECTORS.RTH
!/filnam,RUN_THERMAL_POD,OFF
nall
*vget,NodeList,NODE,,NLIST
!*SMAT,Nod2Bcs,D,IMPORT,FULL,strcat(GenerationJobname,'.full'),NOD2BCS
!*VEC,MapForward,I,IMPORT,FULL,strcat(GenerationJobname,'.full'),FORWARD
*VEC,MapBack,I,IMPORT,FULL,strcat(GenerationJobname,'.full'),BACK

! Store into .RTH file
/POST1
file,RUN_THERMAL_POD_VECTORS,rth
set,first
lcdef,erase

*dim,nodeUsrNumbering,array,nbNodes
! A vectoriser avec un *voper, gather
*do,ind_node,1,nbNodes
  nodeUsrNumbering(ind_node)=MapBack(ind_node)
*enddo

! Vecteur dans l'espace APDL (pas APDL math!)
*dim,currTempVecIntAPDL,array,nbNodes

*do,ind_step,1,nbSingularValues !nbSnapshots
  *VEC,currTempVecInt,D,LINK,U,ind_step
  !*EXPORT, Matrix, Format, Fname, Val1, Val2, Val3
  *EXPORT,currTempVecInt,APDL,currTempVecIntAPDL
  /NOPR
  dnsol,nodeUsrNumbering(1:nbNodes),TEMP,,currTempVecIntAPDL(1:nbNodes)
  /GOPR
  /title,POD vector #%ind_step%
  plnsol,temp
  !RAPPND, LSTEP, TIME
  rappnd,2,1000+ind_step
*enddo
FINISH

! EXPORT TO PNG
/POST1

file,RUN_THERMAL_POD_VECTORS,rth
cmsel,s,EL_DISP
*do,ind_step,1,nbSingularValues
  setTime=instants_List(ind_step)
  set,,,,,1000+ind_step
/show,png,REV
  plnsol,temp
/show,close
*get,currentJobname,ACTIVE,0,jobnam

```

```
/RENAME,%currentJobname%000,png,,POD_VECTOR_%ind_step%_THERMAL,png
*enddo

```

step4_store_POD_vectors_into_RTH.inp

```
fini
resu,step3,db
/inpu,vue_iso.gsav
!/EXPAND,2,RECT,HALF,,,0.00001
*use,revVideo.MAC

/title, Structural analysis - response for POD vectors
/filnam,Run_Structural_POD_Vectors,OFF
/sys, del Run_Structural_POD_Vectors,rst

/COM ********************************************************************************
/COM *** ETAPE 1: CONVERT MODEL FROM THERMAL TO STRUCTURAL ***
/COM ********************************************************************************
/prep7
! get rid of thermal coupling equations
cedele,all

! Change Thermal to Structural
etchg,tts

! Elements pour l'affichage
csys,0
nsel,s,loc,y,-1e3,0
nsel,r,loc,x,0,1e3
esln,s,1
esel,inve
cm,El_DISP_STRUC,ELEM
eall $ nall

! Boundary conditions
csys,0
nsel,s,loc,z, 0.86000E-001
cm,ND_UZ,NODE
d,ND_UZ,UZ

cmsel,s,ND_UZ ! stabilisation mais expansion thermique libre
nsel,r,loc,x,0
d,all,UX

cmsel,s,ND_UZ
nsel,r,loc,y,0
d,all,UY

nall

matID=1
!mp,ex,matID,170e3     ! MPa
!mp,nuxy,matID,.3
!mp,alpx,matID,15e-6
mp,reft,matID,0      ! Expansion thermique en RELATIF

nall $ eall
eplo
save,model_structural,db,,MODEL
```

```
55
56 /COM
       ********************************************************************************
57 /COM *** ETAPE 2: RESOLUTION STRUCTURALE                                        ***
58 /COM
       ********************************************************************************
59 fini
60
61 /SOLU
62 antype,static,new
63 outres,erase
64 outres,all,none
65 outres,nsol,all
66 outres,esol,all,EL_DISP_STRUC
67
68 ! Allocate matrices to separately store real and imaginary part of solution
69 ! row: frequency
70 ! col: DOF
71 ! plane: node
72
73 *dim,UsolR,array,F_nb_steps,NbDOFPerNode,NbOutputNodes
74 *dim,UsolI,array,F_nb_steps,NbDOFPerNode,NbOutputNodes
75 autots,off
76 nsubst,1
77 kuse,1   ! re-use triangularized stiffness matrix
78 *DO,ind_step,1, NBSINGULARVALUES        ! LOOP OVER POD VECTORS
79   ldread,TEMP,,,1000+ind_step,0,Run_Thermal_POD_Vectors,RTH
80   time,ind_step
81   /title,Structural response to POD vector %ind_step%
82   solve
83   ! *do,ind_n,1,NbOutputNodes
84   ! NdOutput=ListNdOutput(ind_n)
85     ! UsolR(ind_F,1,ind_n)=1e3*UX(NdOutput)
86   !UsolR(ind_F,2,ind_n)=1e3*UY(NdOutput)
87   ! UsolR(ind_F,2,ind_n)=1e3*UZ(NdOutput)
88   ! *enddo
89 *ENDDO
90 fini
91 save,step5,db,,MODEL
92
93 /COM
       ********************************************************************************
94 /COM *** ETAPE 4: EXPORTATION FICHIER TEXTE
       ***
95 /COM
       ********************************************************************************
96
97 /COM CREATE A GLOBAL MATRIX CONTAINING REAL AND IMAGINARY PARTS OF SOLUTION
98 ! *dim,UsolC,array,F_nb_steps,1+2*NbOutputNodes*NbDOFPerNode
99
100 /COM row: frequency
101 /COM col: NodeI DOF 1/ Node I/ DOF2 etc...
102 /COM so that we can output a text file with
103 /COM column 1= frequency
104 /COM column 2/3/4= X/Y/Z of Node 1
105 /COM column 5/6/7= X/Y/Z of Node 2
106 /COM etc
107
```

```
/COM Fill 1st colum (frequency)
! *do,ind_f,1,F_nb_steps
! UsolC(ind_f,1)=F_table(ind_f)
! *enddo

! *do,ind_node,1,NbOutputNodes
  ! *do,ind_DOF,1,NbDOFPerNode

    ! *do,ind_f,1,F_nb_steps
      ! ind_r=2+2*NbDOFPerNode*((ind_node-1))+2*(ind_DOF-1)
      ! ind_i=ind_r+1
      ! UsolC(ind_f,ind_r)=UsolR(ind_f,ind_DOF,ind_node)
      ! UsolC(ind_f,ind_i)=UsolI(ind_f,ind_DOF,ind_node)
  ! *enddo
  ! *enddo
! *enddo

! *cfopen,command_file,inp
! *vwrite,ExcType,NbOutputNodes,NbDOFPerNode
! *mwrite,UsolC(1,1),Run_Struct_FRF_ExcType=%c_NbNd=%i_NbDOF=%i,txt
! *vwrite,'(',1+2*NbOutputNodes*NbDOFPerNode
! %c F8.1,X,%i(F12.8))
! *cfclose

! /inpu,command_file,inp
!/sys,del command_file,inp

/COM
    ********************************************************************************
/COM *** ETAPE 3: EXPORTATION PNG                                                ***
/COM
    ********************************************************************************

! EXPORT TO PNG
/POST1

!file,RUN_THERMAL_POD,rth
*get,currentJobname,ACTIVE,0,jobnam
*do,ind_step,1,nbSingularValues

  set,ind_step
/show,png,REV
  plnsol,u,sum
/show,close

/RENAME,%currentJobname%000,png,,POD_VECTOR_%ind_step%_STRUCTURAL,png
*enddo
```

step5_solve_POD_vectors_structural.inp

```
fini
!resu,model_structural,db
/filnam,RUN_STRUCTURAL_POD,OFF

/POST1
reset
file,Run_Structural_POD_Vectors,rst

! IDEE 1: on essaie d'utiliser les "load case"
! pour conserver toutes les possibilitÃ©s de post-traitement
! Ref: Basic Analysis Guide - Â§7.3

! COMBINAISON
*do,ind_step,1,nbSnapshots

lczero
lcdef,erase

*do,ind,1,nbSingularValues
! LCDEF, LCNO, LSTEP, SBSTEP, KIMG
lcdef,ind,ind,1
*enddo


! On a rangÃ© les coordonnÃ©s
! dans sigmaVconj (nbVecxnbInstants)
lcfact,1,SigmaVconj(1,ind_step)
lcase,1

*do,indVec,2,nbSingularValues
lcfact,indVec,SigmaVconj(indVec,ind_step)
lcoper,add,indVec
*enddo

!LCOPER, Oper, LCASE1, Oper2, LCASE2,SweepANG
!lcase,1
!*do,ind,2,nbSingularValues
!lcoper,add,ind
!*enddo
ct=NINT(instants_List(ind_step)*100)/100
/title,Structural response using POD (snapshot #%ind_step% t=%ct%s)
cmsel,s,EL_DISP_STRUC
plnsol,u,sum

/show,png
/replo
/show,close
/RENAME,RUN_STRUCTURAL_POD000,png,,RUN_STRUCTURAL_POD_DISP_%ind_step%,png

plnsol,s,eqv
/show,png
/replo
/show,close
/RENAME,RUN_STRUCTURAL_POD000,png,,RUN_STRUCTURAL_POD_SINT_%ind_step%,png
*enddo

!lcwrite,11,toto,ext
```

step6_structural_solution_using_POD.inp