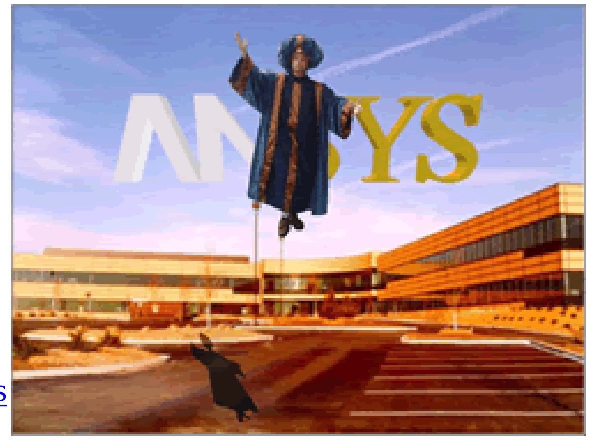
The Focus PADT

A Publication for ANSYS Users

Contents



The Focus PADT
A Publication for ANSYS Users

Post Processing Tips & Hints: Animation in ANSYS

by Eric Miller, PADT

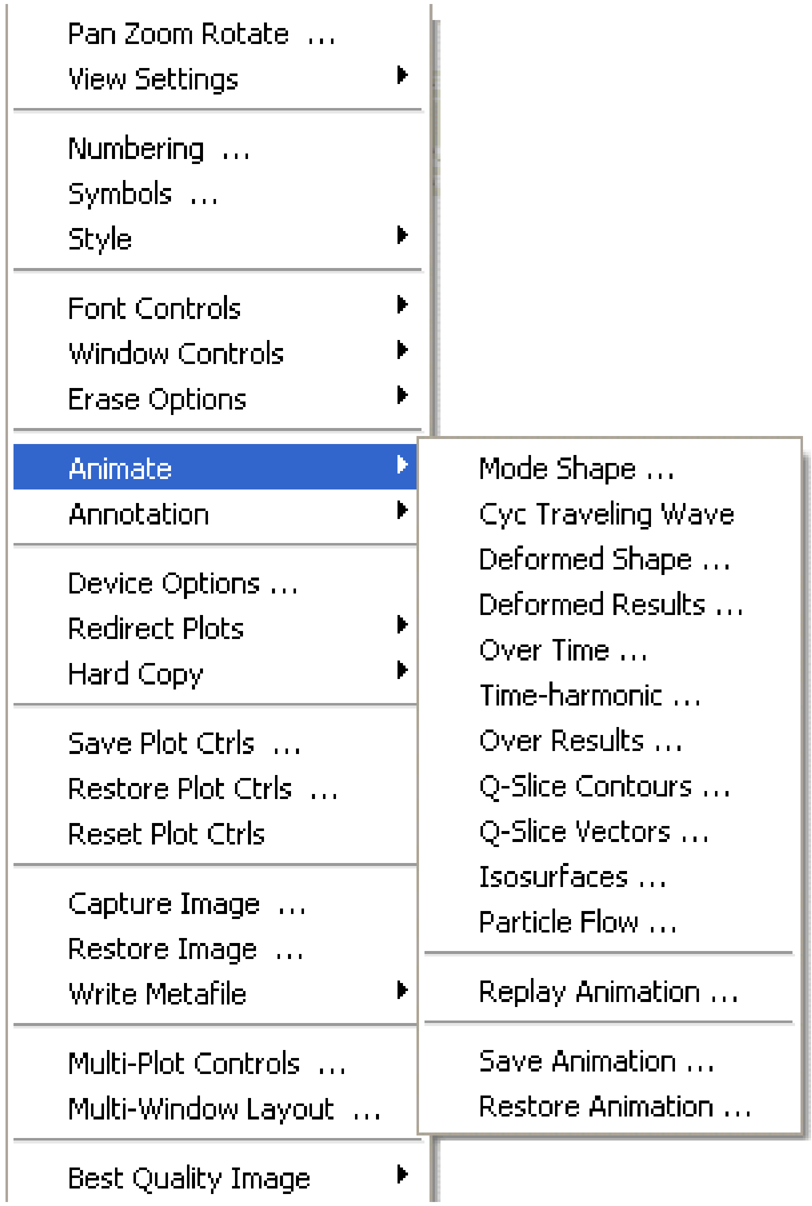
Post processing is an oft-maligned part of the analysis process. Real analysts focus on meshing and running their models. But the fact is, getting meaningful answers is the whole point of doing a simulation. So spending a little bit of time to learn how to produce more concise and digestible results is a good investment. To that end, we will be covering a couple of key ways to produce higher quality results using POST1 and POST26 including listing, graphing, sections, and for our fourth article, creating animations in ANSYS.

If a picture is “worth a thousand words,” then an animation can be worth 100,000. Because an animation is simply a series of single images, called frames, played rapidly in succession, the key to getting the most out of animation in ANSYS is understanding how to create clear images, automate what is seen, and use the built-in tools to capture a series of pictures. Of course, it is important you have

A Publication for ANSYS Users

something that changes incrementally; otherwise, creating animations is a waste of time.

Another often over-looked application is to use animation files as a “folder” for multiple images. Instead of running them as an animation, you step through them one at a time to see different aspects. This can sometimes be easier then making a PowerPoint or shipping 15 JPEGs to someone.



Animation Basics

Before beginning it is always a good idea to stop and determine the goal of an animation. What value are you trying to show and how do you want to show it. The three fundamental ways to show change in a FE animation are to show distortion of the shape, change colors, or create an annotation and move or change its value. All of these are accomplished through APDL scripts in ANSYS. Example 1 shows a simple macro for animating the displacement with contours of a mode shape and Figure 2 is the resulting animation.

```
/reset
/annot,dele
/sss,1
```

The Focus PADT
A Publication for ANSYS Users

```
*if,abs(arg1),lt,.1,return

md = arg1
udir = arg2
set,1,md
*get,frq,active,,set,freq

/win,1,-1,.33,.15,1
/win,2,.33,1.67,.15,1
/win,3,-1,.33,-.7,.15
/win,4,.33,1.67,-.7,.15
/win,5,-.2,1.67,-1,-.7

/view,1,0,0,1
/vup,1,-x
/view,2,0,1,0
/vup,2,x
/view,3,1,0,0
/vup,3,z
/view,4,1,1,1
/vup,4,z

/focus,5,100,100,100
/dist,all,1.1
/dist,4,1.4
/triad,off
/plopt,minm,off
/plopt,info,3
/plopt,leg1,off
*do,i,1,4
  /udoc,i,cntr,off
  /udoc,i,date,off
*enddo
/udoc,5,date,off

/pspec,0,0,1
/poly,4,-1,-1,-.2,-1,-.2,-.7,-1,-.7
/txtre,file,66,padt_logo,bmp
/lsym,-.9,-.9,0,66,0,1
```

A Publication for ANSYS Users

```
/tspec,15,1,1,0,0
/tlab,-.5,-.77,Mode %md%
/tlab,-.5,-.83,%frq% Hz
/tlab,-.5,-.89,U%udir%
/title,Modal Animation

/contour,all,,auto
plnsol,u,udir
*get,dsc0,graph,1,dscale,dmult
*get,vmn,graph,5,contour,vmin
*get,ncnt,graph,5,contour,ncont
*get,vnc,graph,5,contour,vinc
vmx = vmn + (ncnt*vnc)
*if,abs(vmn),gt,abs(vmx),then
  aa = vmn
  vmn = -1*abs(vmn)
  vmx = abs(aa)
*else
  vmn = -1*vmx
*endif
/contour,all,ncnt,vmn,,vmx
/seg,delete
/seg,multi,anm1,.1
nn = 20
*do,i,1,nn
  scl = sin(2*3.14159*(i-1)/(nn-1))
  dsc = dsc0*scl
  *if,dsc,eq,0,then
    dsc = 1e-7
  *endif
  /dscale,all,dsc
  /sss,scl
  plnsol,u,udir
*enddo
/seg,off
/sss,1
/contour,all,,auto
anim,5,1,.4
```

The Focus
A Publication for ANSYS Users

Example 1.

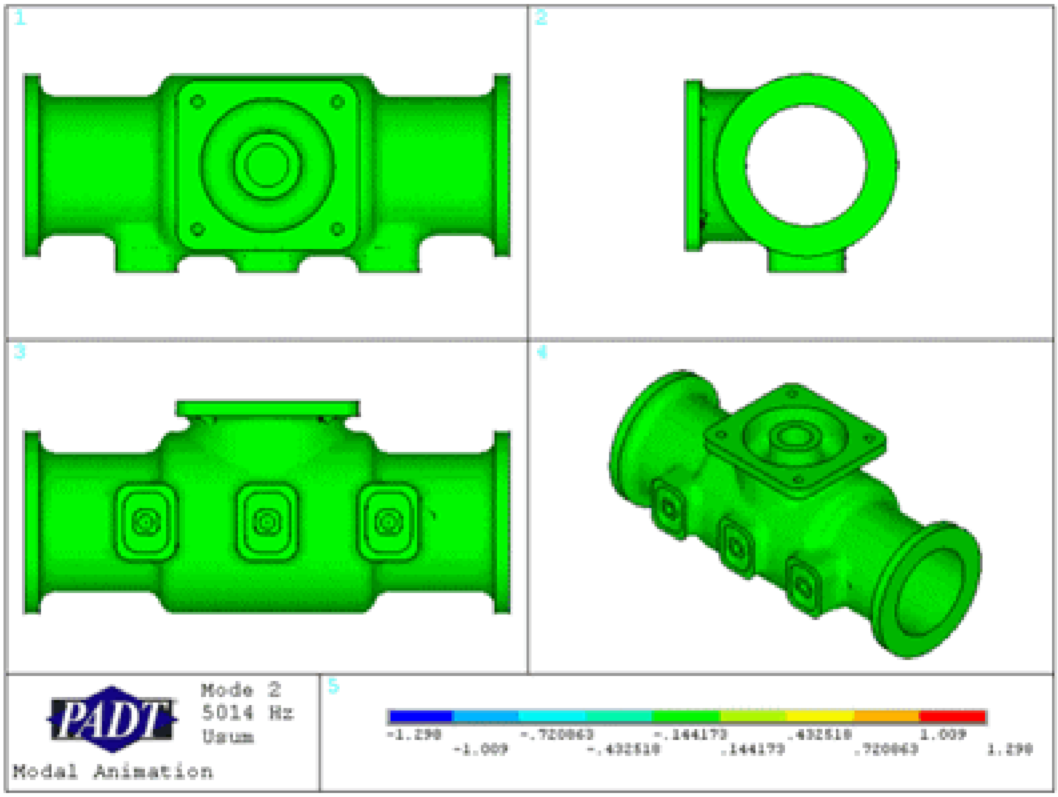


Figure 2.

Although this macro may seem intensive to build, use it as a template and make minor changes as necessary. Also, ANSYS comes with a fairly complete suite of animation macros exposed under the PlotCtls->Animation menu, and also available in the installtion directory. For interactive review of results, one of these tools should work for most cases. The macros set contours, scaling, and other values, and change them over the user specified range of the animation. If more user control is needed, or something different is required, then user-created APDL scripts can be used to modify the existing macros (or create new ones).

Under the Hood

ANSYS uses a command called /SEG to create and manage animations. ANSYS basically copies the pixels in the screen bitmap to memory, called a segment (see previous article on plotting to review how ANSYS creates bitmap images). The command is used to turn storage on, close segment storage and write the animation file, and to delete what is stored in memory. You can also use the

The Focus PADT
A Publication for ANSYS Users

command to specify the animation file name, the pause between frames, and to show the status of the /SEG settings. To create animation files in ANSYS, you simply use /SEG in a macro. On Windows platforms, this file is a Microsoft AVI format.

The Files

Since this file contains frames of the graphics bitmap, they can get very large. The size of the frames is determined by the size of your graphics window. PADT recommends that you shrink down your graphics window before creating AVIs to speed things up and create manageable files. You can also get a variety of tools that convert AVI files to other formats, if size is an issue.

To view your animation, ANSYS provides a tool that lets you play animation files interactively. It is pretty basic so we recommend that you use Media Player or some other high-quality AVI player.

Getting a Good Image

The key to getting good images is to create a clean and clear bitmap image. Since you are only going to view it on a screen, there is no reason to go to the extremes discussed in the plotting article for getting good printed images. For example, there no need to use precise hidden line algorithms or really large image sizes. The best thing to do is set your graphics window — not your screen — to something around 600 to 1000 pixels wide, and use your view controls to get a good look at your parts.

In addition, you should take some time to clean up the non-FE information on your screen, such as the legend, contours, and color. Get a nice static image on the screen, and then capture the settings used to create it in an APDL script to make sure your animation looks clean. This is a good idea even if you are using the ANSYS-provided macros, because using a setup macro ensures a consistent look and feel from one animation to another.

Also note that if auto screen sizing is activated (/AUTO,1), the perspective might zoom in and out with deflection, or move the “camera” with a moving part rather than remaining stationary. In these cases, you might want to deactivate it (/USER,1).

The Focus PADT
A Publication for ANSYS Users

Since your animation often “stands on its own,” it also pays to take the time to add some annotation that describes what is being seen. A future article will talk more about annotation, but as you can see in the example script, you can add some nice content that improves the amount of information captured in your animation.

Last but not least, it is recommended that you change your colors to reverse video. This is because the white lines/text on a black background look fuzzy in an animation, and the black-on-white scheme seems to give a clearer image.

Automating the Scene Creation

Once you are ready to create the individual frames, you will need to use APDL to change the scene from one frame to another. This is usually something that works well in a *DO loop. The most common method is to establish a constant up front that will be the number of frames you want to capture. Then you can divide other values by the loop counter to get the increments that are needed to change values over the animation range. How you change deflected shape or result information depends on if you are animating a single result set or animating over a set of results. In a modal or static analysis, you have one set of results (displacements, stresses, temperatures, flux, etc.) and you will want to show how those values change realative the zero value. To do that, you scale the values with /DSCALE for the element distortion and /SSS for the contour values. Example 1 shown above demonstrates this for the displacement scaling of the plot (/DSCALE) and the contour result values (/SSS) as a sinusoidal function of the maximum deflections.

If you have a set of results that you want to “walk” through, then you need to find your extreme result values and set your /DSCALE and /CONTOUR to capture the maximums, and then keep them constant as you use the SET command to resume a result set and plot it for each frame. The do-loop in Example 2 shows this usage.

One thing that people always want to change in an animation is the orientation between the camera and the object. In most cases this is a bad idea for an analysis plot. If it is for marketing, have fun, but for information capture and transfer you should only change what you are studying. It is far better to create an animation like the example with multiple views if you need to show different sides of the object.

Lastly, if you are creating an animation where discrete values you care about are

The Focus PADT
A Publication for ANSYS Users

changing, changing the annotation over time is a great way to highlight results. It is not uncommon for people to use a blank window in the frame to list the key inputs and outputs with annotation and change them for each frame as the actual values change. This is simply accomplished by placing *GETs and annotation commands with %val% substitution in your do-loop. A similar method is to place a graph in a window as is done in Example 2.

Saving the Images

Most of the time, you will use the /SEG command, as explained earlier, to save the frames to an AVI file. /SEG,multi,fname.avi specifies that you want to start saving and this is the file, and /SEG,off writes the data to the file.

However, if you want a higher quality image or if you are doing something that /SEG does not support (like GPLOT or /ERASE, /NOERASE), then you need to save your frames to a plotting file using /SHOW just as you normally would. If you do a /SHOW,fname,ext, then the information is stored in ANSYS DISPLAY program format and you can create an AVI from with the DISPLAY program. If you specify some bitmap format, then you need to use a 3rd party program (PADT uses Jasc Animation Shop to convert the frames into an animation format.

The most common format that animations are saved in are AVI files. Unfortunately, these can be very large (50-100 MB), so it is nice to have other options. MPEGs are nice and are much more common these days. PADT has found that creating animated GIFs is also a nice option. These can be displayed in PowerPoints without having to carry around the AVI file, and they are also great for web pages.

Thoughts on Animation

When used properly, a good animation can really help the viewer understand the physical response of a system. If it is kept simple and automated with APDL, the creation of high-quality animations is also very easy. As with everything in ANSYS, using a library of scripts can certainly make the process more efficient.

If you have animation questions or are looking for clever ways to make different animations, post your question to XANSYS and you will get lots of responses on this topic.

The Focus
PADT
A Publication for ANSYS Users

```
/reset
/annot,dele
/win,1,-1,.33,.15,1
/win,2,.33,1.67,.15,1
/win,3,-1,.33,-.7,.15
/win,4,.33,1.67,-.7,.15
/win,5,-.2,1.67,-1,-.7

/RGB,INDEX,100,100,100, 0
/RGB,INDEX, 80, 80, 80,13
/RGB,INDEX, 60, 60, 60,14
/RGB,INDEX,  0,  0,  0,15

nstp = 10

/view,1,1,1,1
/vup,1,y
/view,2,0,1,0
/vup,2,-z
/view,3,1,0,0
/vup,3,y
/focus,5,10000,0,10000

/dscale,1,1
/dscale,2,1
/dscale,3,1

/triad,off
/plopt,minm,off
/plopt,info,3
/plopt,leg1,off
*do,i,1,4
  /udoc,i,cntr,off
  /udoc,i,date,off
*enddo
/udoc,5,date,off

/pspec,0,0,1
/poly,4,-1,-1,-.2,-1,-.2,-.7,-1,-.7
```

The Focus PADT
A Publication for ANSYS Users

```
/txtre,file,66,padt_logo,bmp
/lsym,-.9,-.9,0,66,0,1
/tspec,15,1,1,0,0
/tlab,-.5,-.77,LS-DYNA
/tlab,-.5,-.83,Example
/tlab,-.5,-.89,USUM

set,last
plnsol,u,sum

/focus,1,17.36,10,-31.14
/dist,1,30.44
/focus,2,17.36,10,-31.14
/dist,2,47.18
/focus,3,17.36,10,-31.14
/dist,3,35.39

zmx1 = uz(n1)
zmx2 = uz(n2)
zmx = zmx1
*if,zmx2,gt,zmx1,then
  zmx = zmx2
*endif

*dim,dt1,table,nstp,2,3

dt = .1
/nstp
tt = 0
/xrange,0,.1
/yrange,0,zmx2
plnsol,u,sum
/contour,5,,user

/show,lsexp.apf
/gtype,all,node,0
/gtype,all,line,0
/gtype,all,area,0
/gtype,all,keyp,0
```

The Focus
PADT
A Publication for ANSYS Users

```
/gtype,all,volu,0
/gcmd,1,plnsol,u,sum
/gcmd,2,plnsol,u,sum
/gcmd,3,plnsol,u,sum
/gcmd,5,plnsol,u,sum
/gcmd,4,*vplot,dt1(1,0),dt1(1,1),2
*do,i,1,nstp
  dt1(i,0) = tt
  set,,,,,tt
  /title,Over Time Anim
  dt1(i,1) = uz(n1)
  dt1(i,2) = uz(n2)
  tt = tt + dt
  *vlen,i
  gplot
*enddo
dt1=

/show,close
/show,term
```

Example 2.

The Focus
PADT
A Publication for ANSYS Users

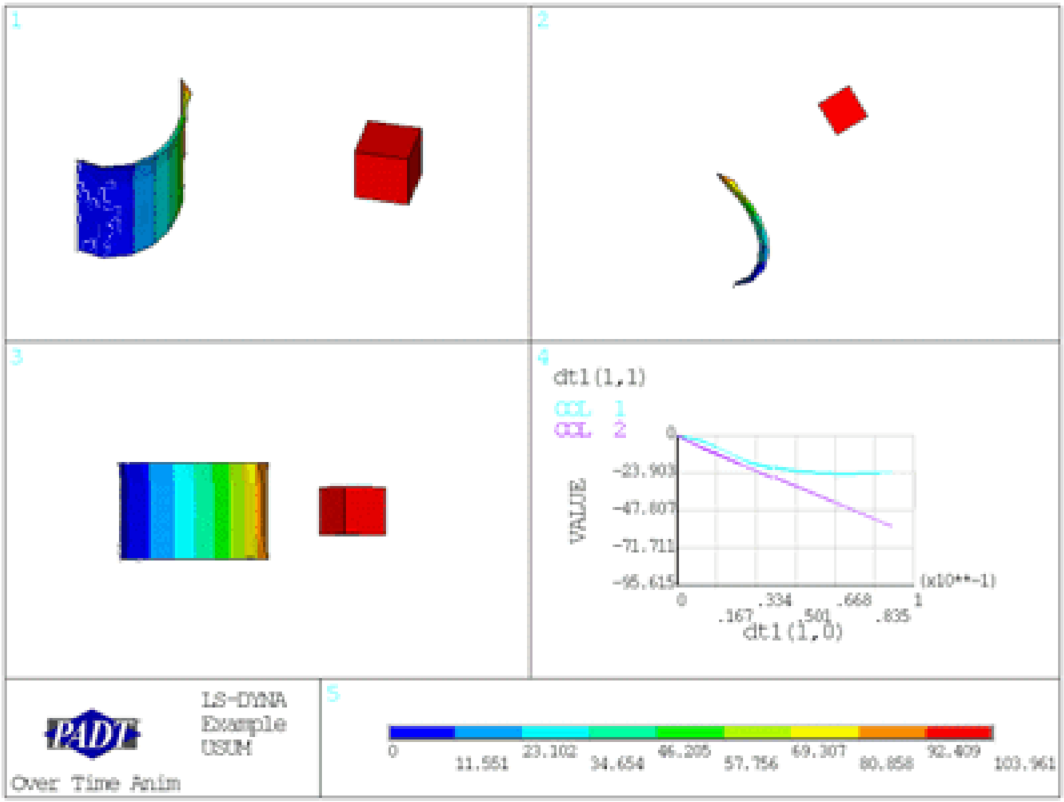
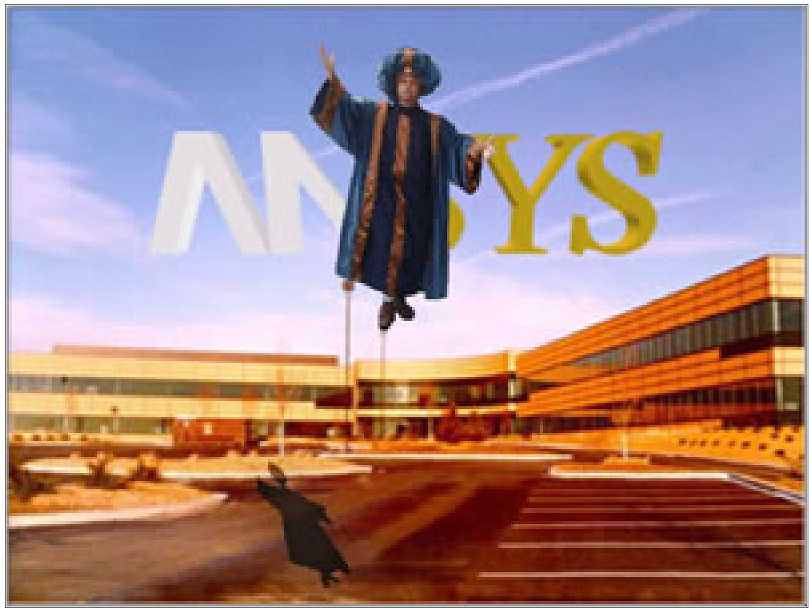


Figure 3.

The Focus PADT
A Publication for ANSYS Users

Are You an APDL Guru?

by Matt Sutton, PADT



Finally, a chance for you, a loyal *Focus* reader, to show the world that you are an APDL Guru! Test your neural noggin against the scenario below, and send your answers to apdl.guru@padtinc.com.

The first person with the correct answer gets… well, I suppose nothing, other than the satisfaction that they are truly sharper than everyone else. And isn't that why we got into this gig in the first place? At long last, all those dark hours of intensive honing of intellect will reach fruition and you can take your honorary seat at the mountain top. That is, until next month's edition of "Are You an APDL Guru?", when you slide back to the depths of anonymity.

It's 4:30 and you've got three analyses you want to run overnight. Unfortunately, you only have one ANSYS license, so you need to run the jobs sequentially. Flexing your APDL might, you effortlessly arrive at the following solution: First you create a directory called my_runs. Under that directory, you create three subdirectories called run_1, run_2, run_3. In each subdirectory, you place the databases for the three runs. Finally, you create the following little script:

```
fini
/cle
*do,i,1,3
parsav,scalar,foo,par
/cwd,./run_%i%
resume
```

The Focus PADT
A Publication for ANSYS Users

```
/solu
solve
fini
/cle
parres,new,foo,par
*enddo
```

You save the script in my_runs, launch ANSYS, and execute the script. Everything seems to be working according to plan, so you check out for the day. When you arrive back in the morning, you see problems. What are they, and what are the smallest changes you can make to the above script that will fix them?

Submit your answer.

The Focus PADT
A Publication for ANSYS Users

*Varying Your *VWRITE

by Jeff Strain, PADT

Imagine that you've written a macro to store an array of nodal displacements over time. Now you want to write out those displacements to a text file to be imported into Excel. If you select the same number of nodes each time, it's pretty easy. Let's say the array stores displacements for the five nodes exhibiting the highest deflections. Your export block of script would look something like this:

```
*CFOPEN,nodedisps,txt
*VWRITE,ndisps(1,1),ndisps(1,2),ndisps(1,3),ndisps(1,4),ndisps(1,5)
(5F10.4)
*CFCLOSE
```

But what if you want to write out data based on the number of selected nodes? Maybe you select three nodes during one run, seven nodes the next, four nodes after that, and so on. One trick is to use *MWRITE instead of *VWRITE, since *MWRITE writes out entire planes, or matrices, to a file. You probably could have saved yourself some trouble by using *MWRITE originally, but that's okay. It's water under the bridge now.

The real issue is the format descriptor. How do we vary the number of format descriptors for each run? The trick is to write a separate macro from the original macro using *VWRITE and then execute it. When writing this macro, you will build up the descriptor lines by taking advantage of the $, which in FORTRAN suppresses the carriage return. You could also build up a string array and write that out to the macro, but thanks to Gary Betts' suggestion on XANSYS regarding the dollar sign, I've learned that it's much easier than that.

The following demonstrates the writing-a-macro-from-a-macro approach:

```
*CFOPEN,ndispwrite,mac
*VWRITE
('*MWRITE,ndisps(1,1),nodedisps,txt')
*DO,index,1,ndinqr(0,13)
  !ndinqr(0,13) returns number of selected nodes
*VWRITE
('%10.4F '$)
*ENDDO
```

The Focus PADT
A Publication for ANSYS Users

```
*CFCLOSE
NDISPWRITE
/DEL,ndispwrite,mac
```

Note that the format descriptors in the ndispwrite.mac file are based on C instead of FORTRAN. Since the C descriptors don't require parentheses, it's much easier to set up from the original macro.

This example represents a fairly simple case. Now suppose you want to make things more complicated. Perhaps you've performed a random vibration analysis and have an array containing frequency values in the first column and output PSD values in the remaining columns. More than likely, you'll want a different format descriptor for the frequency column. The way to handle this is to simply write out the first portion of the descriptor string, followed by the $, just prior to executing the *DO loop. For example:

```
*CFOPEN,psdwrite,mac
*VWRITE
('*MWRITE,psdvals(1,1),psdcurves,txt')
*VWRITE
('%6.0F '$)
*DO,index,1,ndinqr(0,13)
*VWRITE
('%10.4F '$)
*ENDDO
*CFCLOSE
PSDWRITE
/DEL,psdwrite,mac
```

Using the above tricks, you can create and call a macro to give your text output just about any format you desire, including headings. If you wish to continue writing the macro beyond the format descriptors, you can place a final format descriptor, minus the dollar sign, just after the *DO loop or use the APPEND option on the *CFOPEN command. If you're skipping array columns or using *VLEN to limit the number of rows that get written, you can use the $ to write out the appropriate *VWRITE command strings as well. If you would like to learn more about creating arrays and exporting and importing data, consider attending the Introduction to APDL course taught by PADT. I've heard the instructor is outstanding.

The Focus PADT

A Publication for ANSYS Users

About *The Focus*

The Focus is a periodic electronic publication published by PADT, aimed at the general ANSYS user. The goal of the feature articles is to inform users of the capabilities ANSYS offers and to provide useful tips and hints on using these products more effectively. *The Focus* may be freely redistributed in its entirety. For administrative questions, please contact Rod Scholl at PADT.

The Focus Library

All past issues of *The Focus* are maintained in an online library, which can be searched in a variety of different ways.

Contributor Information

Please don t hesitate to send in a contribution! Articles and information helpful to ANSYS users are very much welcomed and appreciated. We encourage you to send your contributions via e-mail to Rod Scholl.

Subscribe / Unsubscribe

To subscribe to or unsubscribe from *The Focus*, please visit the PADT e-Publication subscriptions management page.

Legal Disclaimer