

The Focus



A Publication for ANSYS Users

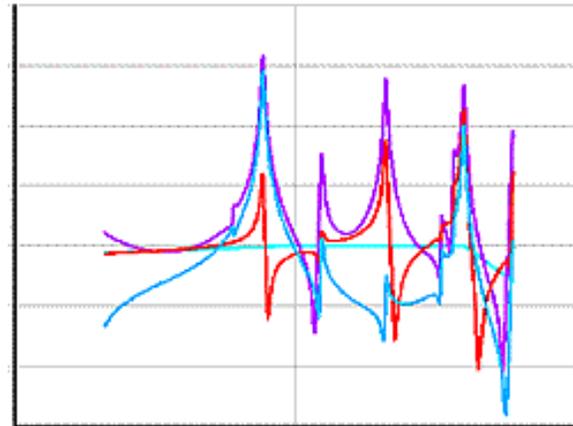
Contents

Feature Articles

- [Post-Processing: The Case for Load Case Operations](#)
- [PSD Output Curves](#)
- [Are You an APDL Guru?](#)

On the Web

- [Excellent Demos of ANSYS Meshing Products](#)
- [Belcan ANSYS Support Site](#)
- [PADT: ANSYS Training & Baseball Spring Training](#)



ANSYS
Support Distributor

Resources

- [PADT Support: How can we help?](#)
- [Upcoming Training at PADT](#)
- [About *The Focus*](#)
 - *The Focus* Library
 - Contributor Information
 - Subscribe / Unsubscribe
 - Legal Disclaimer

The Focus



A Publication for ANSYS Users

Post Processing Tips & Hints: The Case for Load Case Operations

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 that getting meaningful answers is the whole point of doing a simulation and 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, animation, and for our sixth article, how to use the often overlooked load case utilities in POST1.

One of the advantages of doing any type of linear analysis is that the results created from a given set of loads can be combined with another result set to create a combined solution, scaled to give a new results set, or operated on in several other ways. This can be used to your advantage to reduce the number solves, conduct complex modal superposition analysis, and calculate derived results values very quickly. Another overlooked use is as a way to compare the results from multiple load steps by finding minimums, maximums, or differences.

Before we talk about LCs (I'm going to use LCs because I'm already tired of typing "load case"), it is important to go back to the concept of the "database" in ANSYS. The database is not the *.db file or the result file, it is the name of all the data that ANSYS currently stores in memory. When post processing, the results from the last solution solved or from the last SET command are the results in the database. This is important because many of the LC commands operate on the results information stored in the database, and the results of most operations overwrite the contents of the database with the new calculated values. Think about this and make sure you have your head around it before we move on:

If you do a SET,2, then the database contains the results from load step 2 on the result file. If you then use LCOPER,SQRT, then the database will contain the square root of the results that are stored on load step 2 of your results file. If you then do a SET,3, then those results in memory are overwritten by the results stores on load step 3 in the *.rst file. There is no way to tell what results are currently in

The Focus



A Publication for ANSYS Users

the database so you need to keep track.

Most of the Load Case commands begin with LC and are summarized in Table 1. A simple example of combining two bending loads in a beam is shown in Listing 1.

Table 1. Load Case Commands.

Command Description

LCABS	Specifies absolute values for load case operations.
LCASE	Reads a load case into the database.
LCDEF	Creates a load case from a set of results in a results file. Use LCDEF,STAT to see how your load cases are defined.
LCFACT	Defines scale factors for load case operations.
LCFILE	Creates a load case from an existing load case file.
LCOPER	Performs load case operations.
LCSEL	Selects a subset of load cases.
LCSUM	Specifies absolute whether to process nonsummable items in load case operations.
LCWRITE	Creates a load case by writing results to a load case file.
LCZERO	Zeroes the result portion of the database.
RAPPND	Appends the results from the database to the results file.

Listing 1. Simple Bending Load Case Combination Example.

```
!Model Setup Stuff
finish
/clear
/file,lctst
/prep7
! Build simple beam 10 elements tall
et,1,45
n,1,-.5,-.5
n,2, .5,-.5
n,3, .5, .5
n,4,-.5, .5
ngen,2,10,all,,0,0,1
e,1,2,3,4,11,12,13,14,15
ngen,11,10,all,,0,0,1
egen,10,10,all
ex,1,10e6
nuxy,1,.23
! Fix one End
```

The Focus



A Publication for ANSYS Users

```

d,1,all
d,2,all
d,3,all
d,4,all
!-----Write Load Steps
/solu
! Load Step 1: Down 20 lbs
f,102,fx,-10
f,103,fx,-10
/title,FX -20
lswrite,1
fdel,all,all
! Load Step 2: Side 20 lbs
f,102,fy,10
f,103,fy,10
/title,FY 10
lswrite,2
! Load Step 2: Down and Side 20 lbs
f,102,fx,-10
f,103,fx,-10
/title,FX -10, FY 10
lswrite,3
!-----Solve
lssolve,1,3
finish
/post1

!!!!!!!!!!!! LC STUFF !!!!!
set,1      ! Get First Load Step
lcwrite,1  ! Write it as Load Case 1
lcdef,2,2  ! Define Load Step 2 as Load Case 2
lcooper,add,2 ! Add Load Case 2 to current results
lcwrite,4  ! Save current (1 + 2) results as Load Case 4
rappnd,4   ! Save current (1 + 2) to results as Load Step 4
!!!!!!!!!!!! LC STUFF !!!!!

set,1
usum1 = sqrt(ux(101)*ux(101)+uy(101)*uy(101)+uz(101)*uz(101))
set,2
  
```

The Focus



A Publication for ANSYS Users

```
usum2 = sqrt(ux(101)*ux(101)+uy(101)*uy(101)+uz(101)*uz(101))
set,3
usum3 = sqrt(ux(101)*ux(101)+uy(101)*uy(101)+uz(101)*uz(101))
set,4
usum4 = sqrt(ux(101)*ux(101)+uy(101)*uy(101)+uz(101)*uz(101))
*msg,,usum1
  LS 1: %g
*msg,,usum2
  LS 2: %g
*msg,,usum3
  LS 3: %g
*msg,,usum4
  LS 4: %g
```

What's a Load Case?

A load case really is a single results set. When one issues the SET command, one reads in a single result, or Load Case, which is pretty trivial. However, if one wants to add a couple results in an .rst file (such as thermal and mechanical loads), one defines each of the results as a Load Case, so they can be operated on. Said simply, a set of results becomes a Load Case when you label it as one using LCWRITE.

Tagging Load Cases

The first thing to do when working with load cases is to define your load cases. Load cases are identified by an integer from 1 to 99 that is assigned by the user. The LC must point to either a result file or to a special file called, you guessed it, a load case file. LCDEF is used to point to a result file set and LCFEIL is used to point to an existing LC file. If you want to make a LC file from the current results in the database, use LCWRITE. This last one is important because this is how you save the results of an operation as something you can get back to.

Operating on Load Cases

Once you have identified load cases you can start doing calculations using LCOPER. This is one of those commands that can be tricky and you should study

The Focus



A Publication for ANSYS Users

the manual page before using it. The basic syntax is:
 LCOPER,Oper1,Lcase1,Oper2,Lcase2. When executed the command does:
Database = Database Oper1 (Lcase1 Oper2 Lcase2). Note that, if specified, the
 Oper2 part is done first, followed by Oper1 part. Let's go over each Oper1
 operation in Table 2:

Table 2. LCOPER Oper1 Options.

Option	Description
ZERO	Sets database result values all to 0. Rest of command is ignored.
SQUA, SQRT	Square and square root of the database result values. Other values ignored.
LPRIN	Recalculates line element principal stresses. Stress are shown for the NMISC items of the ETABLE command for the specific line element type.
ADD, SUB	Add or subtract Lcase1 values from database values.
SRSS	Square root of the sum of the squares (RSS) of database and Lcase 1.
MIN, MAX	Compare Lcase 1 and the database, then save the smaller (min) or larger (max) value in the database.
ABMN, ABMX	Compare the absolute value of Lcase1 and the absolute value of the database, then save the smaller (abmn) or larger (abmx) value in the database, with the sign before the absolute value was taken.

Oper2 can only currently be used to multiply numbers, and is in fact the only way to multiply two LCs. But, you can't have a blank Oper1 value, so you need to use two LCOPER commands:

```
LCOPER,zero      !Zeros out the database
LCOPER,add,1,mult,2 !Multiplies LC1 and LC2, then adds
                  !to the database, which is zero
```

Just as with the vector commands in APDL, you can specify scale factors (LCFACT) or absolute values (LCABS) be applied to load cases prior to their use in an LCOPER or LCASE command. Specifying them does not change the values stored in the LCs, it just temporarily scales and/or absolute values the results right before an operation.

LCFACT and LCABS can't point towards the database, so if you want to scale a value in the database you need to apply LCFACT or LCABS to the LC before you make it the current result set with LCASE.

The Focus



A Publication for ANSYS Users

Doing Multiple Operations

Sometimes you want to apply an operation to many LCs. You can put LCOPER commands in for each one, or you can use the LCSEL command to specify a group of LCs to work with, then use ALL instead of a number in the Lcase1 variable. As an example, if you have 8 load cases and you want to find the maximum stress found in LC 1 through 5 and 7, you could do:

```
lcase,1  
lcooper,max,2  
lcooper,max,3  
lcooper,max,4  
lcooper,max,5  
lcooper,max,7
```

or

```
lcsel,s,1,5  
lcsel,a,7  
lcase,1  
lcooper,max,all
```

After execution, the database will contain the maximum result value across the selected range.

Saving Your Work

Since the results of an operation are only stored in memory, you need some way to save the calculated results set. One way to do this is to save it as a new LC using LCWRITE. This is a good idea but may be confusing to others down the road. What you likely want to do (unless you're concerned about .rst size) is save your results to a result file. RAPPND writes the results in the database to the end of the current result file. One thing to be careful of is that RAPPND always appends, and it prompts you for a Load Step and Time to label the result set with. If you give it an existing load step number, then you will not be able to read it. The best thing to do is label the appended sets sequentially.

The Focus



A Publication for ANSYS Users

Conclusions, Comments, and Tips

As you can see, many complex analyses and even more complex utilities written in APDL can be replaced with the use of this simple set of commands. Here are some comments based upon PADT's usage of this part of ANSYS:

1. When doing a linear static analysis with more than one load, run each load separately and combine them later in POST1 as LCs. This allows you to do two things: View the contribution of each load separately, and also combine the loads with scale factors if your loads change.
2. As always, use APDL to implement your use of LCs. This allows you to keep track of what you are doing.
3. You can also use RAPPND to move results sets from one *.rst file to another. Use the FILE command to change the active *.rst file:

```
file,res1 !Set active file to "res1.rst"  
set,2    !Read the second load step into the database  
file,res2 !Set active file to "res2.rst"  
rappnd,5 !Append results to end of RST file as load step 5
```

4. If you are doing calculations using vectors on result values, look at LC commands. They are a *lot* faster.

The Focus



A Publication for ANSYS Users

PSD Output Curves

by Jeff Strain, PADT

Creating a complex graph such as a PSD plot can be challenging. Frequently, ANSYS users resort to exporting the data to an Excel spreadsheet and plotting the data there. However, becoming familiar with the behavior of the Time History Postprocessor with regard to PSD data and taking advantage of a couple of APDL commands can save you a lot of time and effort down the road.

In order to plot output PSD curves, you have to write out acceleration results using PSDRES before solving. Once the solution is complete, you'll go to the Time History Postprocessor to retrieve your PSD data at specific nodes.

Upon entering POST26, you'll first store the PSD results. Next, compute the response PSD at the node(s) of interest in acceleration²/Hz. You may wish to convert the PSD in acceleration²/Hz to G²/Hz. Batch input is as follows:

```
/POST26  
RESET  
STORE,PSD,5, !Store PSD values  
NSOL,2,,U,Y, !Store DOF results in var2 (y-direction here)  
RPSD,3,2,,3,1, !Compute PSD results from var2, store in var3  
PROD,4,3,,,,,1/(386.4**2),1,1, !Convert to G^2/Hz
```

As an example, consider a wing with base excitation at the root. The input PSD spectrum is listed in Table 1. [This is the random vibration example from the Dynamics class, in case it looks familiar to you.]

The Focus



A Publication for ANSYS Users

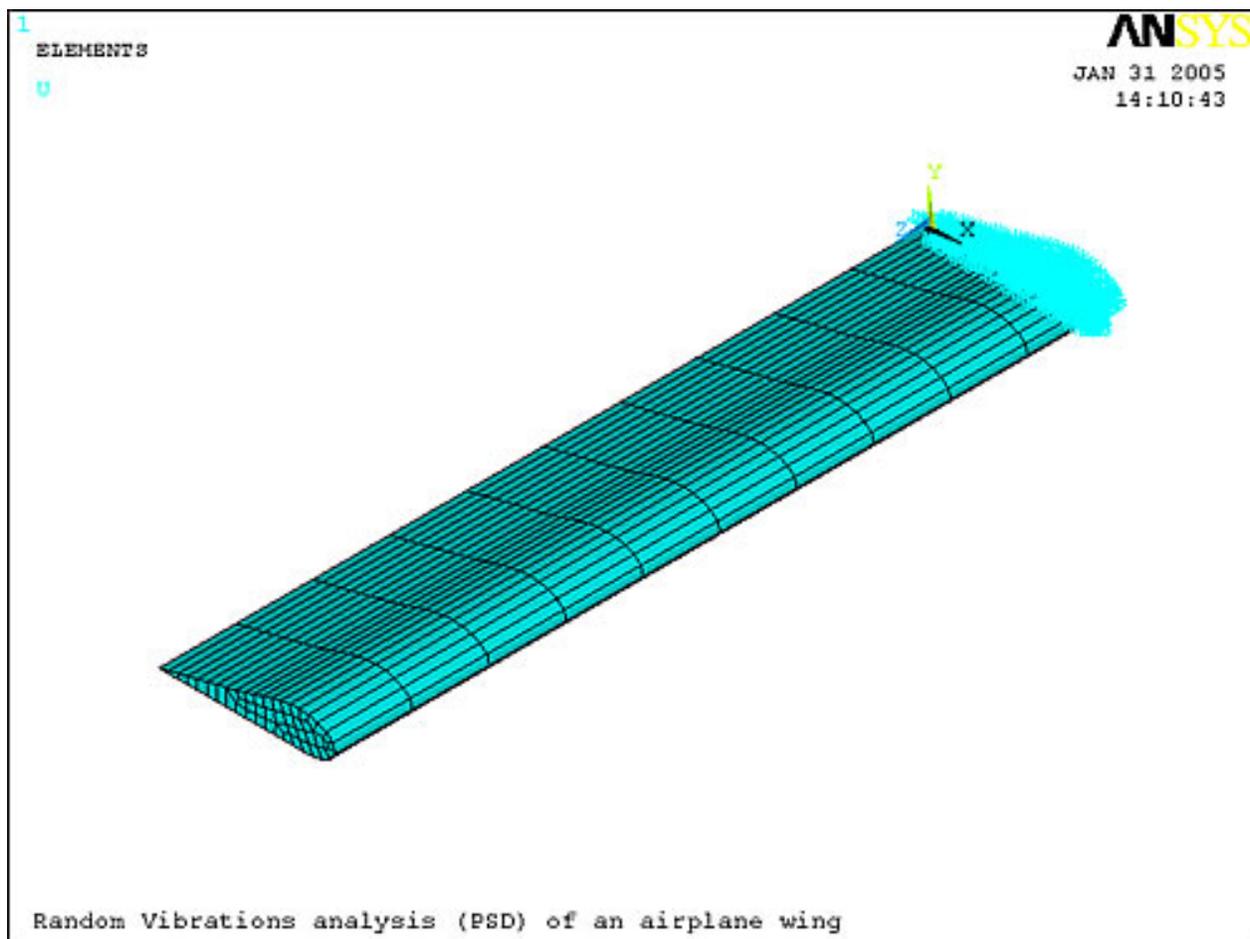


Figure 1. Wing PSD Model.

Table 1. Wing PSD Input Spectrum.

Freq. (Hz)	PSD (G^2/Hz)
20	0.075
100	0.1
400	0.1
600	0.025

If you use the method discussed above, and change the axis labels and convert to log scale via Utility Menu > PlotCtrls > Style > Graphs > Modify Axes, you get an output PSD curve at the tip as shown in Figure 2.

The Focus



A Publication for ANSYS Users

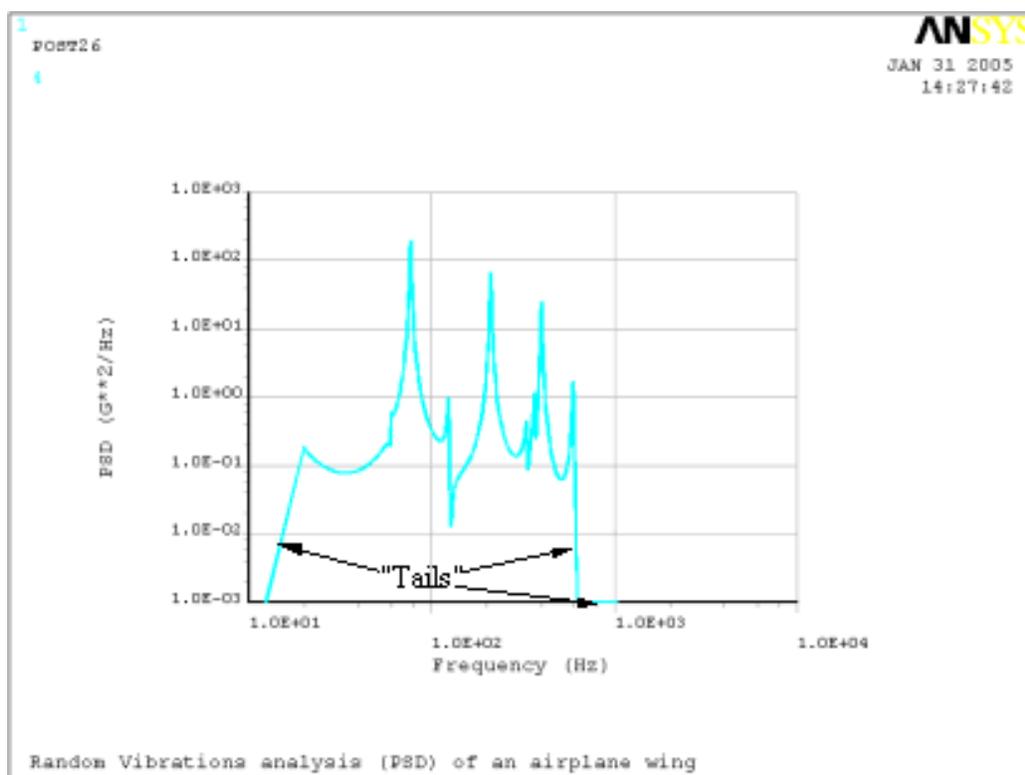


Figure 2. Wing Tip Output PSD.

If you look closely at the PSD plot, you'll notice a slight problem (actually, I labeled the problem, so you shouldn't have to look *that* closely, but I digress). Below the first frequency break point and above the last frequency breakpoint are what I've labeled as "tails" — points beyond the spectrum envelope that go to zero G²/Hz. A listing of the PSD values vs. frequency makes the problem more apparent.

The Focus



A Publication for ANSYS Users

TIME	4 PROD
12.403	0.00000
20.000	0.182871
20.622	0.164270
20.919	0.156752
21.541	0.143280
22.162	0.132269
22.680	0.124583
598.41	0.793975
599.33	0.641171
600.00	0.547824
624.72	0.00000
807.35	0.00000
811.22	0.00000
864.29	0.00000
1033.0	0.00000
0.0000	0.00000
0.0000	0.00000
0.0000	0.00000
0.0000	0.00000

Figure 3. Wing Tip Output PSD Listing.

As you see, for frequencies of 12.403, 624.72, 807.35, 811.22, 864.29, and 1033 Hz, we have PSD levels of zero. These zeros represent modes that were calculated during the modal run, but which lie outside the input spectrum. There are also four points where both the frequency and PSD values are zero. These correspond to loadsteps 2 through 5. The number of these zero points may vary depending on how you use PSDRES.

Obviously, it is in our best interest to remove these zeros from our PSD plots. The best way I found to do so was to first store the frequencies and PSD output values to an array using VGET. The next thing to do is to compress out the zero values via *VMASK and *VFUN,...,COMP using the output PSD array as the masking array, since it contains only zeros and positive values. For example:

```
GET,size,VARI,,NSETS !Retrieve the number of POST26 data sets
*DIM,psds,,size,2 !Define temporary array of PSD data
vget,psds(1,1),1 !Store frequencies in first column
vget,psds(1,2),4 !Store PSD levels in second column
```

The Focus



A Publication for ANSYS Users

```
!Count number of non-zero PSD values
n=
*do,n,1,size
*if,psds(n,2),ne,0,then
ndatpt=ndatpt+1
*endif
*enddo
!End count
*dim,psddata,table,ndatpt,2 !Def. permanent array of PSD data
*vmask,psds(1,2) !Use PSD output as masking array
*vfun,psddata(1,1),comp,psds(1,1) !Compress frequencies
*vmask,psds(1,2)
*vfun,psddata(1,2),comp,psds(1,2) !Compress output PSD levels
```

Note that *psddata* is a table array. The reason will become apparent in the next step.

The next step is to plot the array using **VPLOT*. If *psddata* were a regular array, then **VPLOT* would plot the data as a column chart, which would be a bit of a mess. Since *psddata* is a table array, then **VPLOT,PSDDATA(1,1),PSDDATA(1,2)* will graph the data as a continuous curve.

The Focus



A Publication for ANSYS Users

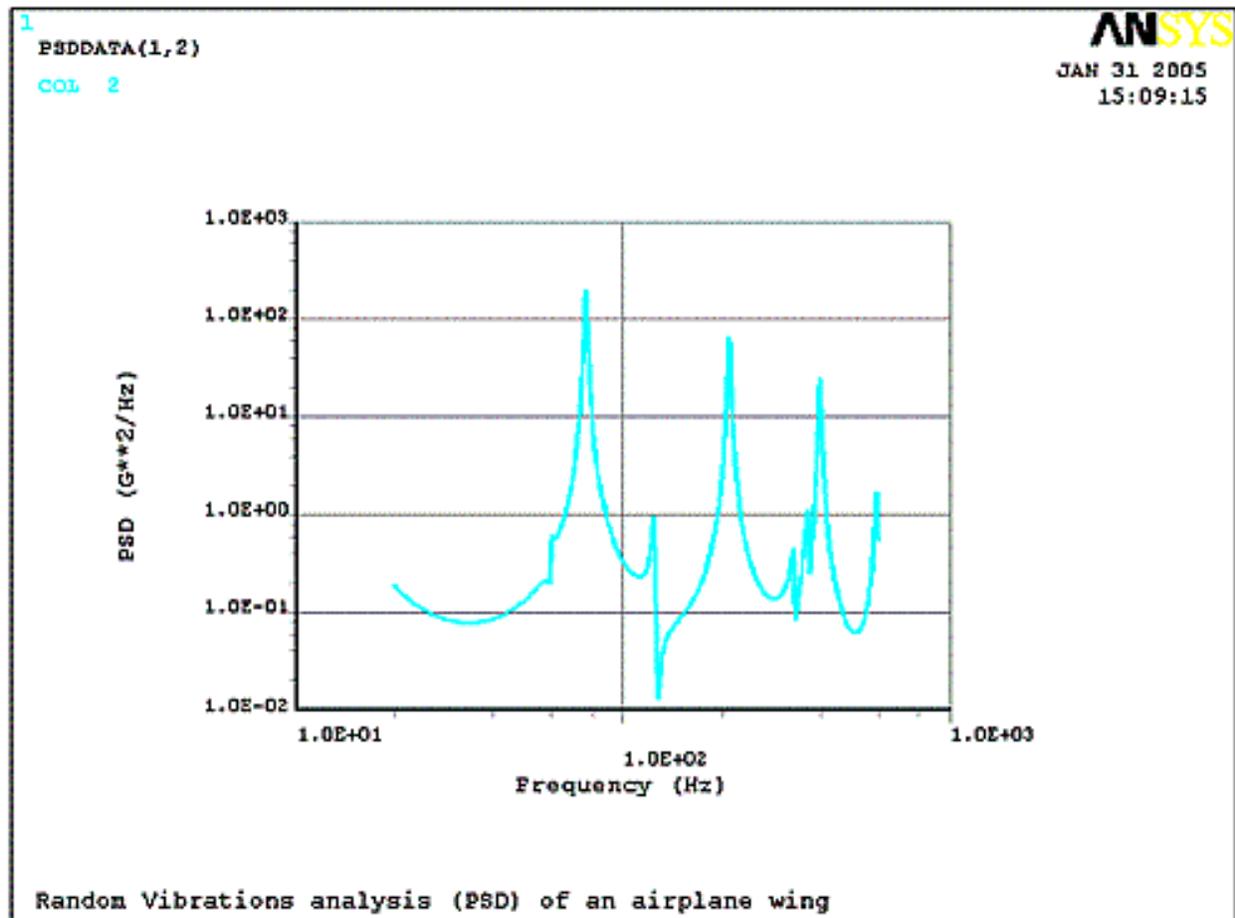


Figure 4. Wing Tip Output PSD Curve (Improved).

You can further improve the plot using [annotations](#) and the /GCOLUMN and /UDOC commands.

As with other repetitive, time consuming tasks, the best way to obtain PSD output curves quickly and consistently is to write a script. Fortunately I've saved you some trouble in this case. The attached macro, [psdout.mac](#), retrieves output PSD data (in G²/Hz) and output GRMS for picked nodes and stores them in Time History variables with the options to plot and/or export them to a text file. The plotting function performs everything discussed here and more as shown in Figure 5. To execute the macro, type in PSDOUT,dir,plot,list where

dir = x, y, or z direction (defaults to x)
plot = 1 to plot (defaults to 0)
list = 1 to export data (defaults to 0)

More detail may be found at the top of the file. Note that the macro creates both a regular array and table array of PSD data. The reason is that *VWRITE references array

The Focus



A Publication for ANSYS Users

index values (i.e., row and column integers) whereas tables don't use this sort of designation.

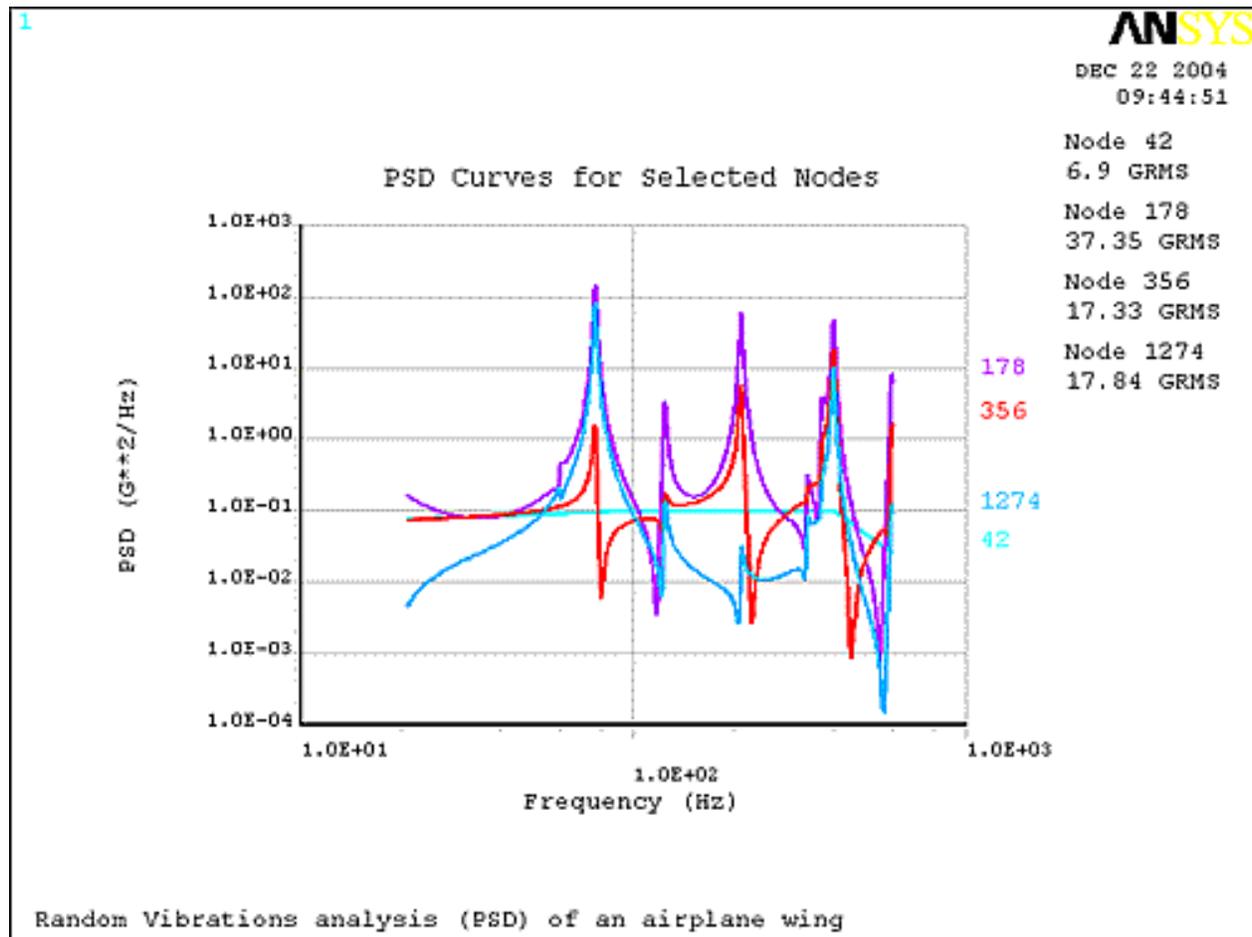


Figure 5. PSD Plot Generated with psdout.mac.

The Focus

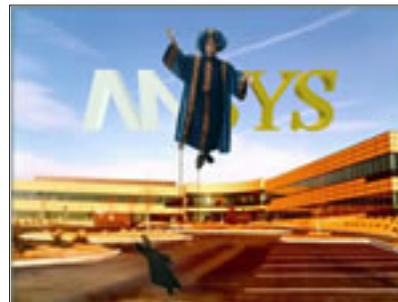


A Publication for ANSYS Users

Are You an APDL Guru?

Ahh, so you thought the Guru was deceased? No, just in deep meditation. It's not simple choosing what vessel to next inhabit.

Last time, we threw out the ambitious request for a "Trim Line" macro with the idea that we might as well make the submissions useful to the community. The results were terrific. We received four good macros. They each had their strengths and weaknesses in terms of robustness, cleaning up after themselves, etc. In fact, I vacillated a bit on which I liked most, eventually choosing the slowest of the bunch. Although I hate "click-and-wait" software, in this case I made an exception. I'd like to say they are all winners because they scored differently depending on the situation — attachment to solids, colinearity, etc. — but (rumbling sound of thunder) "There can be only one!"



So this month's venerated APDL Guru is [Navaneeth Bashyam](#). All hail his excellency Navaneenth! We are not worthy! Throw his [macro](#) in your own directory and make drawing a bit simpler.

Some ideas for improvements:

- After LPTN of intersecting lines, allow users to reselect portions of lines to delete (LSEL,R). Using IMMED,OFF might facilitate this.
- Make the check for coplanarity more than a check for Z1=Z2 to allow 3D functionality.
- Super slow. Maybe use GOTO rather than *create and /NOPR rather than /OUT to speed things up.

If anyone wants to run with these improvements and especially speed it up, we will publish the improved macro and credit you in *The Focus* and on XANSYS.

I have uploaded all four macros, plus one of our own that includes many of the desired features (but fails on a common type of trim extends, unfortunately), to [Sheldon's Site](#). You may find that you like one of the others better. At first, others were my favorites, but failed in some scenarios.

And until next time, peace and may all your analyses be on primitive geometries.

The Focus



A Publication for ANSYS Users

Upcoming Training at PADT

A complete schedule of the training courses for the next three months is shown below. [Learn more](#) about how the **Training Services** offered by PADT can save you time and money. Or, feel free to drop an e-mail to our training coordinator, [Ted Harris](#)!

Month	Start	End	Course	Description	Location
Mar '05	3/7	3/9	101	Introduction to ANSYS, Part I	Tempe, AZ
	3/10	3/11	102	Introduction to ANSYS, Part II	Tempe, AZ
	3/17	3/18	203	Dynamics	Tempe, AZ
	3/22	3/23	501	ANSYS/LS-DYNA	Irvine, CA
	3/28	3/30	902	Multiphysics Simulation for MEMS	Tempe, AZ
Apr '05	4/4	4/6	104	ANSYS Workbench Simulation - Intro	Tempe, AZ
	4/11	4/13	401	Introduction to Electromagnetics	Tempe, AZ
	4/14	4/15	604	Introduction to CFX	Tempe, AZ
	4/19	4/20	201	Basic Structural Nonlinearities	Tempe, AZ
	4/21	4/22	204	Advanced Contact and Bolt Pretension	Tempe, AZ
	4/25	4/27	152	AI*Environment	Tempe, AZ
May '05	5/2	5/4	101	Introduction to ANSYS, Part I	Irvine, CA
	5/5	5/6	102	Introduction to ANSYS, Part II	Irvine, CA
	5/9	5/10	100	Engineering with Finite Element Analysis	Tempe, AZ
	5/12	5/13	803	Tcl/Tk for ANSYS	Tempe, AZ
	5/16	5/17	604	Introduction to CFX	Irvine, CA
	5/19	5/20	203	Dynamics	Irvine, CA
	5/23	5/23	106	ANSYS Workbench DesignXplorer	Tempe, AZ

Whether it s one of our regularly-scheduled classes at our training facility, or a customized class tailored to your specifications and location, you can be assured that the training you receive will have immediate, positive results on your capabilities in design, analysis, and product and process improvement.



The Focus



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

Phoenix Analysis and Design Technologies (PADT) makes no representations about the suitability of the information contained in these documents and related graphics for any purpose. All such document and related graphics are provided as is without warranty of any kind and are subject to change without notice. The entire risk arising out of their use remains with the recipient. In no event, including inaccurate information, shall PADT be liable for any direct, consequential, incidental, special, punitive or other damages whatsoever (including without limitation, damages for loss of business information), even if PADT has been advised of the possibility of such damages.

The views expressed in *The Focus* are solely those of PADT and are not necessarily those of ANSYS, Inc.