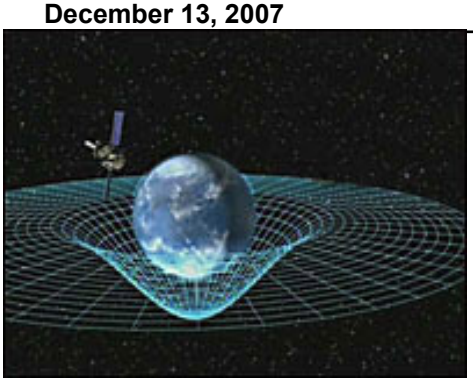
The Focus

PADT

December 13, 2007 | A Publication for ANSYS Users | Issue 61



Mass Units in ANSYS

Did you know you were working in Slinches?

By Rod Scholl

Well this could just be a one line article right?

"Ansys is unitless, just ensure self-consistency within your model."

If only that were enough… I know it wasn't when I first started using English units with ANSYS – and occasionally this subject comes up and words like slugs, poundals, lbm vs. lbf get bantered about, and I wish I had a cheat sheet to hand to someone so they could pour over it in private. So let's make one.

With F=M*A we get:

$$Mass = \frac{Force}{Acceleration}$$

And in English units, we commonly have material modulus in psi, which means our geometry is likely in inches, our outputs stresses will be in psi, and our loads will be in pounds. What kind of lbs? Lbf. (more on that to come).

A 170 lb person's 170 lbm exerts 170 lbf on the scale. So it's not that 170lbm isn't a valid mass unit, it's that it isn't consistent with an acceleration in units of *inch/sec*^{2}.

Thus given these input units of lbf, and inches per second squared, we can derive the units necessary for our mass to maintain unit consistency:

$$[Mass] = \frac{[lbf]}{[inch/\sec^2]} = \frac{[lbf]*[\sec]^2}{inch}$$

What is the name of this unit? This elusive "pound force second squared per inch"? NASA called it the **Slinch**, (aka a mug or a snail*) And come to think of it, wouldn't a better article title have been something like "*The Slinch that Stole Christ'mass*" – ahh missed opportunities…

(Cont. on pg. 2)

* Surprisingly there is not much information on the Internet about the slinch. The best link can be found at: www.unc.edu/~rowlett/units/dictS.html

CAD & Workbench: Sorting Out Options

By Eric Miller

If you ask most people what they like best about workbench, 30% will tell you the robust meshing and 30% will say the CAD connections*. Getting your geometry into Design Modeler, Simulation or CFX Mesh is so much easier in Workbench than in almost any product that is always surprises us when users are unaware of how this works and how great it is. So we thought we would cover a few key facts about this killer capability. †

	Type	Solids	Surfaces	Curves	Attributes	Names	Materials
CATIA V5 (Capri Gateway)	Plugin	Yes					
DesignModeler	PlugIn	Yes	Yes	Yes		Yes	Yes
Inventor	PlugIn	Yes			Yes	Yes	Yes
Mechanical Desktop	PlugIn	Yes			Yes	Yes	
OneSpace Designer	PlugIn	Yes	Yes				
Pro/ENGINEER	PlugIn	Yes	Yes		Yes	Yes	Yes
Solid Edge	PlugIn	Yes	Yes		Yes	Yes	
SolidWorks	PlugIn	Yes	Yes		Yes	Yes	
Unigraphics	PlugIn	Yes	Yes		Yes	Yes	Yes
ACIS	Reader	Yes	Yes				
CATIA V4/V5	Reader	Yes	Yes				
IGES	Reader	Yes	Yes				
Parasolid	Reader	Yes	Yes				
STEP	Reader	Yes	Yes				

Supported Formats

The first thing to know about formats is that they fall into two classes: Readers and PlugIns. Readers simply translate from the CAD format into Workbench's internal format. A plugin actually uses software from the CAD vendor and opens up the geometry in the native format and gives Workbench the information it needs in the native format. We sometimes refer to reader geometry as "dumb" and plugin geometry as smart because the plugin geometry is associative back to the CAD files (see below).

(Cont. on pg. 2)

*These percentages fall in the SWAG category, but are probably pretty close. What about the other 40%, they fall in that "other" category and mention stuff like automatic report, the model tree, ease of use, fast graphics, etc…

† This article was originally published in the September '07 issue of the SWAU Report. It was so popular that we decided to reprint it here.

Table of Contents

(Units, cont...)

So how do we convert our 170lb person to our desired units of slinches? 170 lbs is also written 170 lbm. Both these terms assume earth's gravitational field, and thus 32.17 ft/sec^2 or 386.09 in/sec^2.

$1lbf = 1lbm*1g$

1 g = 386.09 in/sec^2

$$1lbm = \frac{1lbf}{1g} = \frac{1lbf}{386.09\frac{inch}{\sec^2}} = \frac{1lbf*\sec^2}{386.09inch}$$

So we divide the mass in "pounds" by 386.09 and get our mass in slinches. Density:

Similarly one divides the density by 386.09 to keep unit consistency. If you look at many ANSYS APDL macros that define material properties, you will often see the tell-tale den=den/(32.17*12). I would call this a "Slinch per cubic inch" because its just plain fun to say…

Density:

$$\frac{lbm}{inch^3} = \frac{1lbf*\sec^2}{386.04inch} \Big/ inch^3 = \frac{1lbf*\sec^2}{386.09*inch^4}$$

The Poundal Unit of Force

1 poundal = 1lbm*1ft/sec^2

If we enter our mass in lbm, and our lengths in feet, we could interpret our forces in poundals:

$$1poundal = \frac{1lbf*\sec^2}{386.09inch} * \frac{1ft}{\sec^2}$$
$$= \frac{1lbf*ft}{386.09inch} = \frac{1lbf}{32.17}$$

Thus if we employ the poundal we can use the lbm unit, and feet for length… yet now our modulus and stress are in poundals/feet^2 .

Why Not Slugs

Units of slugs are close to what one would want... But a slug uses acceleration in ft/sec^2 not in/sec^2 - this shakes down like such:

$$1lbf = 1slug*1\frac{ft}{\sec^2}$$

With some manipulation one can see that:

$$1slug = \frac{lbf*\sec^2}{foot}$$

Which makes it a factor of 12 off from the Slinch. Of course having geometry in feet would be pretty well accepted, but this also means your modulus and stress are in lbf/ft^2, a kinda non-standard unit.

A Worse Unit System

If you went ahead and used the mass in lbm and the force in lbf… acceleration would have to be in g's! And if acceleration is in g's and we leave time as seconds, then what would your length units be for consistency? It would be :

$$g*\sec^2$$

… and modulus/stress would have to be in:

$$\frac{lbf}{g^2*\sec^4}$$

It'll work… but expect some rotten fruit during that design review.

Metric Is Easy

F = M*A

Force is in units of Newtons, or:

$$\frac{[kg]*[m]}{[s]^2}$$

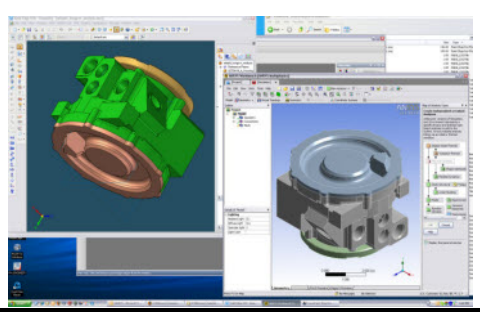
This goes nicely with mass being in kg and acceleration being in m/s^2

(CAD Import, cont...)

The table above lists the formats supported by Workbench at V11. It also gives a brief summary of what does and does not come over from the geometry file into Simulation. Pretty much all types of geometry come into DesignModeler, so if you need surfaces or curves from an unsupported format, go through DM. Attributes and Names are important because these are pieces of information attached to geometry, like loads from a motion simulation or names on entities that can be used to make your simulation job much easier.

How it Works

The readers work in a fairly simple way. They take the geometry file and parse through the description of the entities and topology (how things are connected) and build a geometric model in Simulation or DM. For the non-CAD formats (IGES, STEP, etc…) it often converts to the Parasolid format first, then to the Workbench format, so don't be surprised if you see something about Parasolids during import when you don't have a parasolid file.



The PlugIn's are bit more complicated, but deliver a lot more power. What the PlugIn's do is actually start up the CAD package that the native file comes from in a batch mode. So you need to have the CAD tool loaded on your machine and you need a license for the CAD tool for things to work right. If you have your file already open in your CAD tool, it will just use that session.

If you are bringing in the file for the first time, it builds the geometry and topology and stores any parameters that you have asked to be transferred (this is done by specify a prefix on the parameter names) in

(Cont. on pg. 3)

the parameter manager. If you are doing an update, it can be much faster since it is just updating geometry, topology and parameters.

One quick hint: Set your options in Workbench to release the CAD license when you are done. If you don't do this, you will hold on to your CAD seat even if Workbench is not using it.

Bi-Directional Associativity

The big deal with using PlugIn's is the bi-directional associativety that it allows. If you change a geometry parameter either in Workbench or in the CAD system itself, you can ask for an update and the workbench model will update the geometry - and this is the big deal - any mesh, loads and boundary conditions that you assigned to the geometry. This can literally save days of remeshing.

If you make a topology change, that is add a new surface or delete a surface, things may not update 100%. That depends on the magnitude of the change and the CAD system. But even if it isn't 100%, most of the model updates and you may need to redo a few loads and BC's.

As you can imagine, this makes "what-if", optimization and probabilistic studies very easy.

Try a PlugIn Today

So, if you are using ANSYS or reading dumb geometry into Workbench now, give your salesperson a call and ask for a temp key on a PlugIn for your CAD tool. Give it a shot, it will pay for itself very quickly.

Efficiency with Instancing

By Eric Miller

Did you know that Workbench takes advantage of geometry instancing? Are you even aware of what instancing is? I didn't and wasn't until I sent an e-mail in to ANSYS, Inc. asking about adding mesh copying, and they said they are working on it but in most cases, instancing works even better for what you want. "Oh yea, of course" I said, then quickly tried to figure out what the heck they were talking about. A little research showed a useful set of features that has been there in V11 without much notice.

Instancing

Instancing is a term from the CAD world. It refers to how an assembly treats a part that is already in the assembly. It can read in and store all the surfaces/edges/vertices for that part or it can just point to the original part and apply a transformation. This saves memory and disk space. If you look at the flexure in Figure 1 you see seven volumes made of two parts: two bases on the top and bottom and 5 flexure springs in-between. I modeled these in SolidEdge, but most CAD systems display an instance in the same way; they append some sort of number to the part name in the assembly tree (Figure 2). Note that even though the base1 parts are placed by hand and the flexures use a patterned feature, they both show up as the same part with multiple copies.

Once you import your part into simulation, you will see the instancing in there as well, and it shows up as shown in Figure 3. This works for all of the supported CAD packages but not with the "dumb" geometry files like ACIS, SAT, IGES and STEP, because they do not support instancing. For the same reason, you can not use this feature with DesignModeler, although development is looking into adding support.

Leveraging Instances

Once you have an assembly with repeated parts in Workbench Simulation, you can start to take advantage of it without any special settings or commands. It is completely automatic. The first and most important advantage is that the mesher will automatically recognize that the parts are copies and only mesh the first instance, then copy the mesh for the remainder. This not only saves time, as in this example where the mesher only had to mesh two parts instead of seven, but it also gives an identical mesh on each part. If you actually read the little dialog box that shows progress during meshing, you will notice that it only meshes two parts.

We picked this flexure model as an example because this geometry is very sensitive to mesh variation between pillars. Each mesh must be identical for the model to perform

(Cont. on pg. 4)

Fig. 1: CAD Geometry

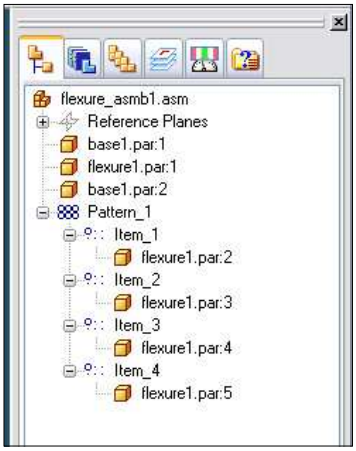


Fig. 2: CAD Model Tree

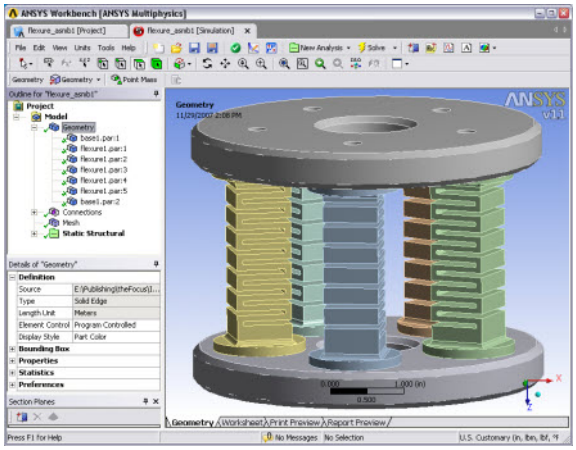


Fig. 3: Model in Simulation with Instances

properly, and as you can see in Figure 4, each has an identical mesh.

The second place where you can take advantage of instancing is with a Beta feature under Extended Selection (to turn on Beta features go to Tools->Options->Common Settings->User Interface->Menus/Toolbars and set "Show Beta Options" to "Yes"). If you select a vertex/edge/surface on a part that is involved in an instance, you can choose the "Extend to Instances" option and it will select the same feature on every instance. This is a huge time saver if you want to apply the same load to each part, or involve the same surface on each part in a contact. Figure 5 shows an example where one of the faces on a flexure can be chosen on each part with only two clicks: pick the part, pick "Extend to Instance."

Observations

This simple little capability reminded me of many important things. First off, there is a lot of stuff in Workbench that I am unaware of (embarrassing!). Even though I read the release notes and even teach update seminars, there is always something I miss or do not remember. Second, the integration between CAD and Simulation is a lot stronger than most of us realize. If Workbench treated a CAD file as a bunch of NURBS and points things would be a lot less efficient, so I'm glad they don't. And lastly, the developers at ANSYS, Inc. are really making a push towards providing the tools needed for modeling bigger and more complex assemblies. Imagine modeling a bunch of bolts or a repeated part in some machine. Instead of meshing and copying (the APDL way) you just need to mesh the one and use "Extend to Instance" to load it.

Note for "The Instance" graphic:

Turns out "The Instance" is some sort of World of Warcraft thing that has tons of references on the web. This was a cool picture so we used it instead of another boring plot from ANSYS...

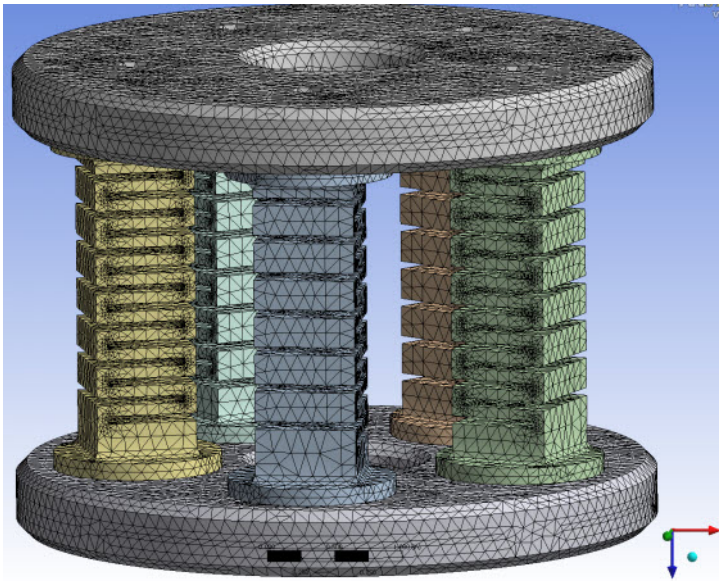


Fig. 4: Identical Meshes on Parts

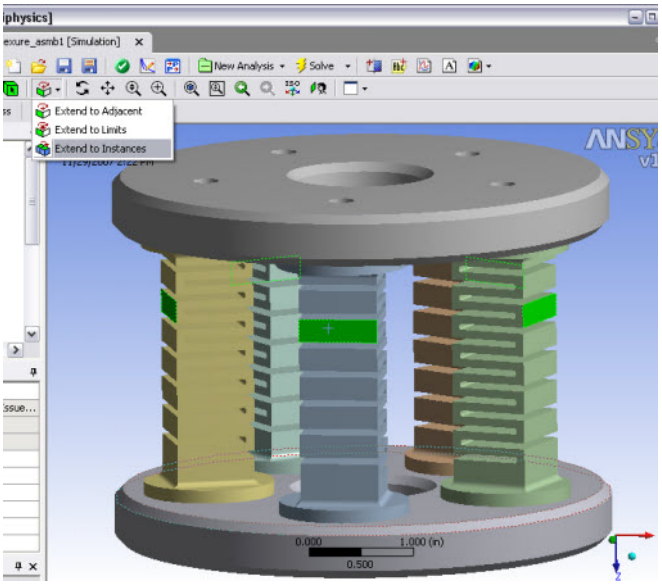


Fig. 5: Instance Selection

Finding Close Areas: Using ACON.EXE

By Eric Miller

Sometimes, old tools never die, they just become undocumented and have limited features. One such tool in ANSYS is something that PADT wrote far too many years ago: a set of tools to do automatic contact detection in PREP7. This was actually part of the MechanicalToolbar and it was almost working well when Workbench came along and made it completely obsolete. But even though many of the pieces are no longer around, the executable that actually found areas that were close to each other is still there. And we use it here at PADT all the time to do all sorts of things.

How ACON.EXE Works

ACON.EXE is a brute force tool that basically takes a faceted representation of your ANSYS areas and finds which facets are close to each other. Because it was part of a more complex set of tools, it does not do things in a general purpose way. But with a little bit of scripting you can overcome or ignore these issues.

Using ACON.EXE

There are three steps to have ACON.EXE figure out which areas are close or touching:

1) Select the areas you want to output and plot them with faceting turned on (/FACE,NORML). Then write out the facets using the undocumented command AGWRITE,Anum1,Anum2,Incr,Filename,Ext. Anum1,Anum2,Inc define the areas you want to write out and Filename,Ext defines the file to put the faceted info into. By default (no arguments) it writes all the areas to jobname.afw

2) Next, you need to run acon.exe with the arguments: filename relTol absTol. You can do this with a /sys command from within ANSYS or from a command window. You have to supply the faceted area file name and a gap size. The second

(Cont. on pg. 5)

argument is actually a relative gap size. We recommend you put 0 there and specify an explicit gap as the third argument. So acon test.afw 0 .005 will look in test.afw for areas that are 0.005 apart or closer.

3) When ACON.EXE is done, it writes two files: ccon.mac and areas.txt. ccon.mac has a bunch of ASEL commands that select all the pairs. It doesn't put them in components or anything. What we do is replace asel,s and asel,a with the name of a macro that does whatever we want to happen between areas that are close, like define radiation or contact. If you want to get real fancy, you can use area.txt, which lists the areas that are close as two numbers separated by a column. Your script could sort, combine, check, etc this data and do more sophisticated tasks.

Figure 1 is a macro to make a simple test case, write facets, and run acon.exe. It also reads the ccon.mac file and parses out the area numbers to build components for each pair. Figure 2 shows output for that model in ccon.mac and figure 4 shows what would be in area.txt. Figure 3 shows the very exciting model.

If you use Workbench, this isn't going to be much use to you, but if you are dealing with large assemblies in ANSYS, with a little bit of scripting you might find it useful.

Figure 1: Example Macro

```
!Setup a blank model
finish
/clear
/file,acontst
/prep7
!Build a flat plate with four cubes
! on top with various gaps
blc4,-1,-1,2,2,-.25
blc4,-.5,-.5,.25,.25,.25
blc4,.5,-.5,.25,.25,.25
wpoff,,,0.05
blc4,-.5,.5,.25,.25,.25
wpoff,,,0.15
blc4,.5,.5,.25,.25,.25
!Turn on faceting and force their
!  creation with an aplot
/face,norml
aplot
!Write facets to file
agwrite
!Execute the acon.exe command
/sys,acon acontst.afw 0 .005
!Read and parse the contents of
! ccon.mac to create components for
! each pair
*sread,lns,ccon.mac
*get,nlns,PARM,lns,dim,2
*do,ii,1,nlns-1
  aaa=strsub(lns(1,ii),14,20)
  c1=strpos(aaa,',')
  aa1 = valchr(strsub(aaa,1,c1-1))
  aaa = strsub(aaa,c1+1,20)
  c1=strpos(aaa,',')
  aa2 = valchr(strsub(aaa,1,c1-1))
  asel,s,area,,aa1
  cm,acn_%ii%a,area
  asel,s,area,,aa2
  cm,acn_%ii%b,area
  cmsel,a,acn_%ii%a
  cmgrp,acn_%ii%,acn_%ii%a,acn_%ii%b
*enddo
```

Figure 2: CCON.MAC Example

```
asel,s,area,,2,7,5
asel,a,area,,2,13,11
aplot
```

Figure 3: AREAS.TXT Example

```
2,7
2,13
```

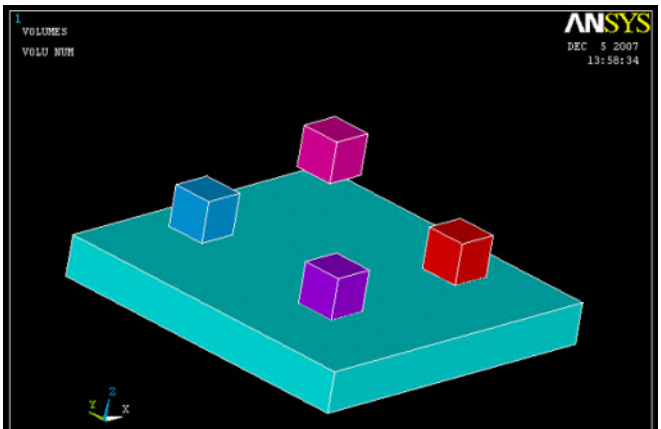


Figure 4: Example Geometry

The Author's sons battling Darth Vadar at Legoland... Hey it is a better space filler then another picture of Doug on top of a mountain!

Awesome APDL: Parsing a Text File

If you actually read the previous article on finding areas that are close together in ANSYS using ACON.EXE, and you actually looked at the example macro, you will have noticed some fancy APDL at the end that parses a text file. So, using one stone to to kill two birds, we thought we would use the same macro for this issue's Awesome APDL.

Yes, I know, I should use python to parse the file. Python is the greatest tool invented by man (or perhaps given to us by aliens) for dealing with text files. But sometimes you want to keep everything in one tool so it is portable. And APDL has most of what you need.

In the example we have a file that contains ASEL commands that select area pairs we want to place in components. So we want to strip off the ASEL, get the two area numbers, and build some componetns.

The first command that you need to learn well is *SREAD. Introduced about 3 or 4 years ago, *SREAD reads a text file and sticks each line in the file into a text string. Basically, it makes an array of the text file:
*SREAD,StrArray,Fname,Ext,,nChar,nSkip,nRead

One of the nice things about the command is that you don't have to *DIM the array up front, it creates it for you. Check out the manual page for details on the command.

So in our example, *SREAD reads the whole file and stores it in the string array lns. Next, a *get is used to figure out how many lines were read. That is used in a do loop, minus the last line because it contains an aplot. For each line in the file, we want to chop off the first fourteen characters (asel,s,area,) then get the two numbers that follow it.

So for the next bit we use the string functions in APDL, documented at the bottom of Appendix B in the ANSYS Parametric Design Language Manual (do a search on strpos in help to find it). We use STRSUB to grab characters 14 through 20. Then STRPOS to count characters to the comma that separates the numbers (c1). VALCHR is used with an embedded STRSUB to convert the string into a number. Then we use STRSUB again to grab the second number on, find the next comma with STRPOS and use VALCHR again to get the second number.

The rest of the macro is a fairly standard use of parameter substitution to select values and create components.

Example Input:

```
asel,s,area,,2,7,5
asel,a,area,,2,13,11
Aplot
```

Example Code:

```
*sread,lns,ccon.mac
*get,nlns,PARM,lns,dim,2
*do,ii,1,nlns-1
  aaa=strsub(lns(1,ii),14,20)
  c1=strpos(aaa,',')
  aa1 = valchr(strsub(aaa,1,c1-1))
  aaa = strsub(aaa,c1+1,20)
  c1=strpos(aaa,',')
  aa2 = valchr(strsub(aaa,1,c1-1))
  asel,s,area,,aa1
  cm,acn_%ii%a,area
  asel,s,area,,aa2
  cm,acn_%ii%b,area
  cmsel,a,acn_%ii%a
  cmgrp,acn_%ii%,acn_%ii%a,acn_%ii%b
*enddo
```

News - Links - Info

- ANSYS and LSTC renew agreement and keep LS-DYNA in ANSYS alive. <link>
- Zuken and ANSYS agree to work together on better ECAD integration. <link>
- ANSYS and Network Analysis team up and release Workbench connection for SINDA/G <link>
- The latest ANSYS Advantage magazine is out and is a real nice publication. Much more technical than in the past. Read and/or subscribe here: <link>
- Need material testing, try: Datapoint Labs, Matereality or Axel Products

Upcoming Training Classes

Month	Start	End	#	Title	Location
Jan '08	1/14	1/16	101	Introduction to ANSYS, Part I	Tempe, AZ
	1/17	1/18	100	Engineering with FEA	Tempe, AZ
	1/24	1/25	801	ANSYS Customization with APDL	Tempe, AZ
	1/28	1/29	104	ANSYS WB Simulation – Intro	ALBQ, NM
	1/30	1/31	207	WB – Structural Nonlinearities	ALBQ, NM
	1/31	2/1	301	Heat Transfer	Tempe, AZ
Feb '08	2/4	2/5	107	ANSYS WB DesignModeler	Tempe, AZ
	2/6	2/6	411	WB Simulation Electromagnetics	Tempe, AZ
	2/7	2/8	205	WB Simulation Dynamics	Tempe, AZ
	2/21	2/21	206	WB Rigid & Flexible Dynamics	Tempe, AZ
	2/25	2/26	202	Advanced Structural NL	Tempe, AZ
Mar '08	3/3	3/4	104	ANSYS WB Simulation – Intro	Tempe, AZ
	3/5	3/6	207	WB – Structural Nonlinearities	Tempe, AZ
	3/10	3/11	203	Dynamics	Tempe, AZ
	3/17	3/18	501	ANSYS/LS-DYNA	Tempe, AZ
	3/26	3/28	902	Multiphysics Simulation for MEMS	Tempe, AZ

The Focus is a periodic publication of Phoenix Analysis & Design Technologies (PADT). Its goal is to educate and entertain the worldwide ANSYS user community. More information on this publication can be found at: http://www.padtinc.com/epubs/focus/about

The Shameless Advertising Page