A Publication for ANSYS Users

Contents

Feature Articles

On the Web

Resources

The Focus

Hex Meshing Tools

by Joe Woodward, PADT

Anyone who has done hex meshing in ANSYS has learned that complex volumes can be sliced into simpler sweepable volumes. For example, take a look at the simple block in Figure 1. With a fillet and a hole, this volume cannot be directly hex-meshed. But, we can cut it horizontally to make two or three sweepable volumes. ANSYS can then free mesh an area and sweep the mesh though the volume. Adjacent volumes can be swept in different directions, resulting in a hex mesh. If you are going to use LS-Dyna, or some other solver that needs an all-hex mesh, then it sometimes takes a little extra work and luck to avoid any wedge elements. With this method, and sometimes a lot of chopping, ANSYS can hex mesh just about anything, right? Well, not exactly.
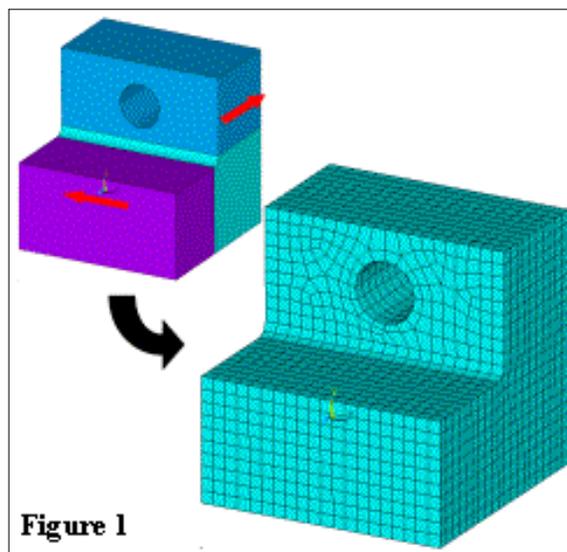

Figure 1

To continue with the same simple case, Case 1, what happens if the fillet is large enough to touch the hole (Case 2), or if the hole is tangent to a wall (Case 3)? These are cases that turn up often when dealing with designers and real geometry. What you get when you try to slice such geometry is the dreaded "topological degeneracy" error. You can also get this error when intersecting slices do not line up correctly, causing sliver volumes. So if you're not careful, all of your hard work may cause problems of its own. But take heart; if you can't change the geometry, there are options to get the job done.

The first option is ANSYS Workbench with Design Modeler. In Design Modeler you can slice the geometry, just as in ANSYS, only more robustly. In Case 2, the hole-fillet tangency, we move the slice down slightly so that it cuts into the fillet and not tangent to the hole. Even if the same could have been done in ANSYS Prep7, the resulting volume would not have been sweepable. There would have been two areas on one side of the volume, and ANSYS will not use concatenated areas as the source or target for a swept mesh. This is where Workbench has another advantage.
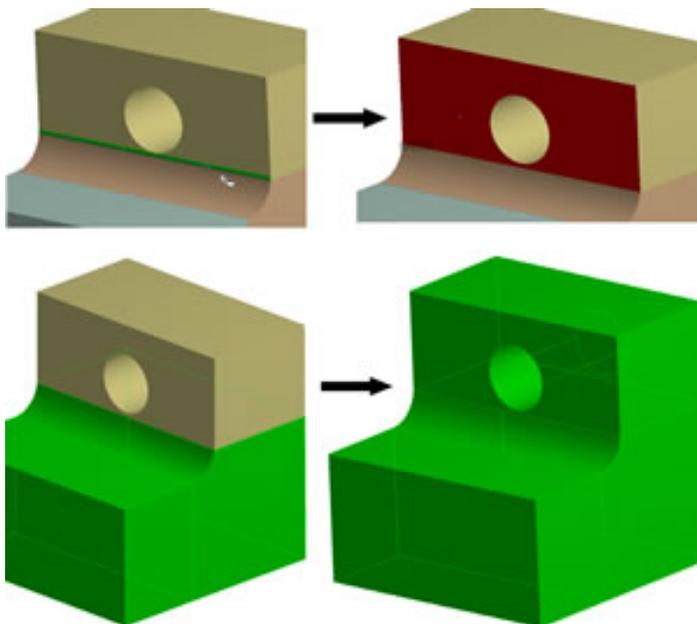
Figure 2

Workbench allows the concatenation of surfaces into 'virtual cells', which can be used in the sweeping process. The top of Figure 2 shows the original surfaces and the resulting virtual cell. The 'Preview Sweep' view highlights all the sweepable volumes in green, as in the bottom of Figure 2. This is very handy since you can see what needs to be changed before you actually try to mesh the model.
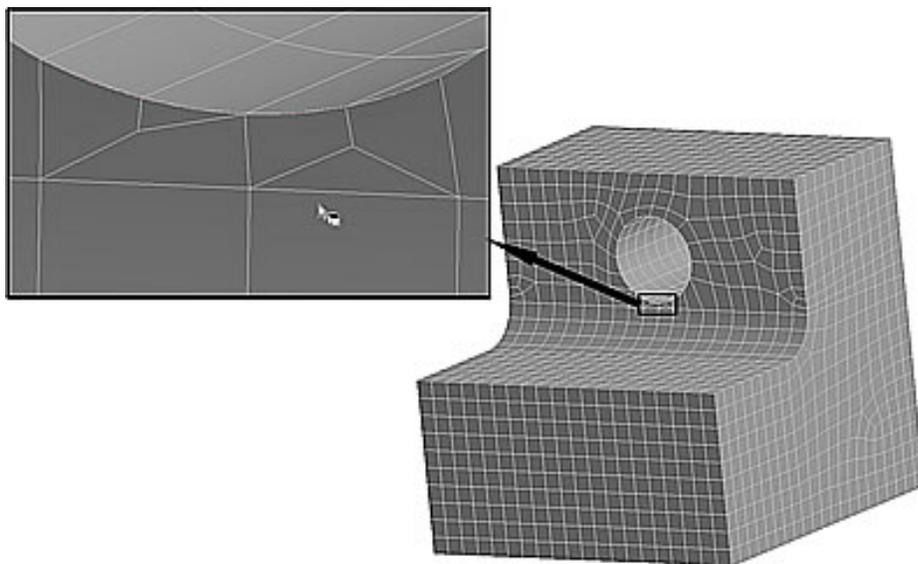

Figure 3

Now that the volumes are sweepable, it does not mean that they will get meshed with all hexes. The first time I meshed this geometry, Workbench put wedges in around the hole and nothing that I tried would get them out. You can specify a mapped mesh on an area, but that did not work because of the hole. I also tried specifying the element size in the hole, but that still did not

The Focus

PADT

A Publication for ANSYS Users

change the mesh. I finally had to go back to Design Modeler and increase the distance between the hole and the slice that I made. The good news here is that Workbench and Design Modeler are parametric and fully integrated. I just changed the parameter for the slice location, and the simulation updated with a new all-hex mesh.

Another option for hex-meshing stubborn geometry is ANSYS AI*Environment. AI*Environment, which was originally ICEM CFD, gives you exact control when hex meshing. Instead of slicing the geometry, the user slices, stretches, and morphs a blocking topology that gets projected onto the geometry. The nodes are then projected onto the surfaces and curves of the geometry. Your geometry does not need to be as clean as with other meshers. Gaps and overhangs in the surfaces are ignored as long as the nodes can be projected to avoid the problems. This also limits the amount of defeaturing necessary since the user controls the nodal projections and can choose not to project to certain features.
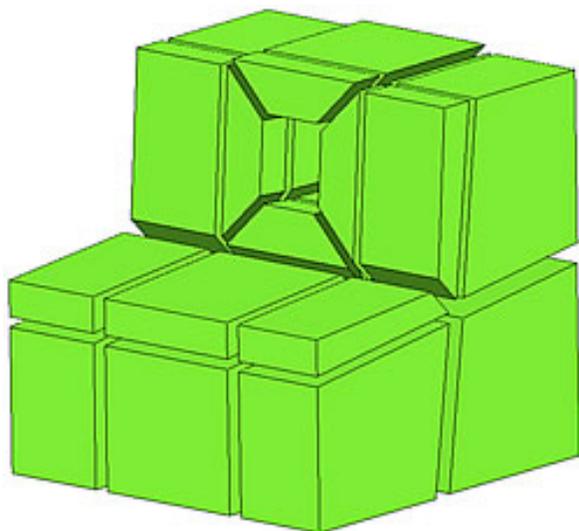
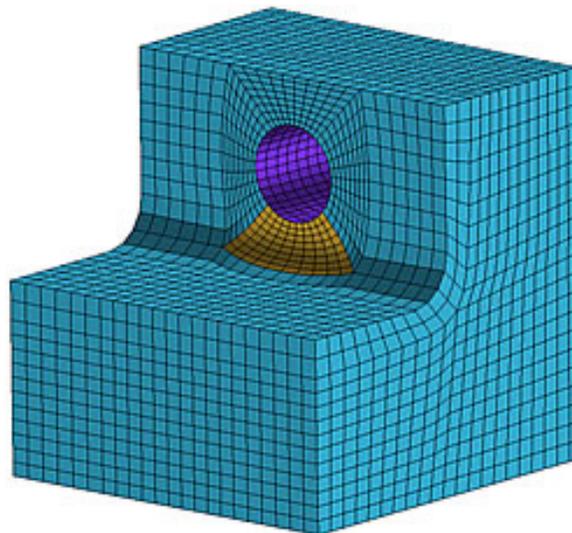Figure 4   Figure 5

Figure 4 shows the final blocking structure for the first geometry case. This structure took four slices and two special features called O-grids. The X-shaped cuts around the hole are created with a single O-grid command, and allow the mesh to be perpendicular around the hole. The second O-grid follows the L-shape of the block, and creates orthogonal elements along the fillet. The element sizes can be taken from the geometry, or controlled specifically on each edge of the blocks. The structured mesh format ensures that the mesh is consistent throughout the model. The blocking structure is saved to a separate file, which can then be used again on other parts with similar topology. This can be a big time savings for companies that make similar components for different systems.

The Focus

Figure 6

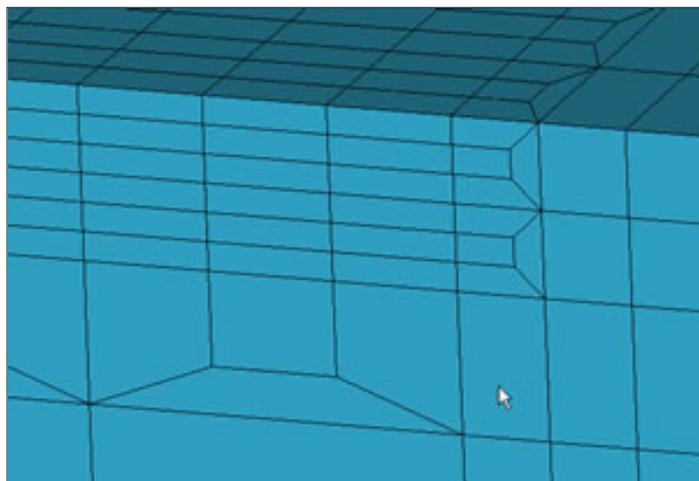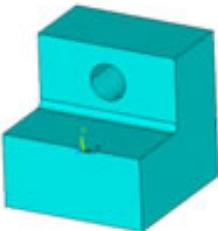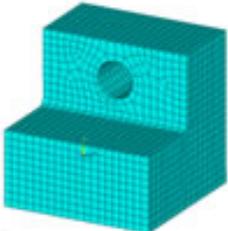AI*Environment also has an abundance of mesh editing tools and quality checks. You can delete elements, merge or move nodes, smooth the mesh, and even build automatic transitions from a fine to coarse mesh as shown in Figure 6. The first three of these, and many more, can be done on any mesh imported from one of more than ten available formats. [I brought in the original Workbench mesh and removed the wedges, just because I could.] When your mesh is ready, you can output to more that one hundred different solvers.

Below is a head-to-head comparison of the three geometry cases meshed on each of the different ANSYS packages. For a mesh quality comparison, all the meshes were brought into AI*Environment for quality checks, with 1.0 being a perfect orthogonal hex element. In each case, a global element size of .1" was used for the 1" x 1" inch block, along with the default settings for everything else. I recorded the time it took to get an all-hex mesh, since ANSYS and Workbench may produce wedges.
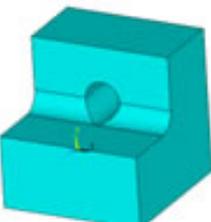
The Focus

A Publication for ANSYS Users

Geometry	ANSYS Prep7	ANSYS WorkBench	ANSYS AI*Environment	Notes
Case 1	5 min / .28	9.5 min / .36	13 min / .47	Numbers are time from ACIS import and the minimum element quality.
Case 2	Error — No hex mesh	10 min / .35	5.5 min / .40	AI*Env Blocking from Case 1 was reused for Case 2.
Case 3	Error — No hex mesh	10 min / No hex mesh	14 min / .40	10 min. in Workbench to try options to get all hex mesh. Not successful.
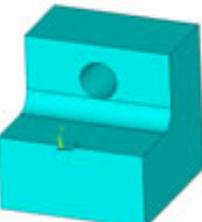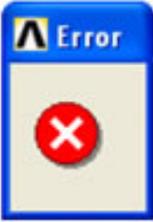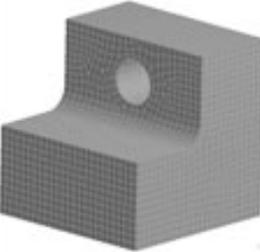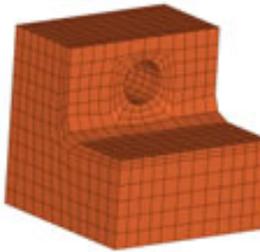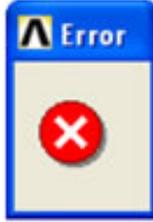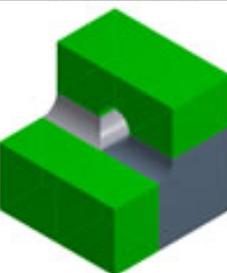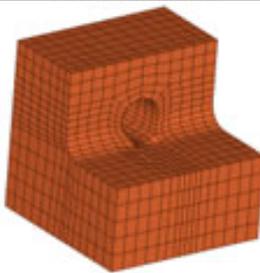
As you can see, only AI*Environment was able to mesh all three cases easily. Although it took longer to mesh Case 1 in AI*Environment than in Workbench, the time savings is recovered by using the same blocking file for Case 2, and the quality is also higher with AI*Environment in all cases. The blocking only changed slightly in Case 3, since a change in the O-grid configuration was needed to accommodate the tangency to the wall.

The Focus

A Publication for ANSYS Users

You should always consider your tools depending on the job that you need to do. If you think of the job of hex meshing as a screw, you can think of the three packages as screwdrivers. ANSYS is the standard flat-head screwdriver. It works well for most projects, but sometimes you just need something else. ANSYS Workbench would be the handy reversible screwdriver that has both flat and Phillips heads when one just won't do. And finally, there is AI*Environment. You can compare it to the little screwdriver that comes on the giant Swiss Army knife you see in the window at the knife shop. It may seem little cumbersome at first, but once you get used to it, you also have all those other tools to help take care of any job.

The Focus

PADT

A Publication for ANSYS Users

Memory Usage Guidelines for Windows

by Gene Poole, Technical Fellow, ANSYS Inc.

The following guidelines are intended to help ANSYS users maximize ANSYS performance on Windows workstations running IA32 processors. Memory usage is the Achilles' heel of 32-bit operating systems given the capability of modern processors for speed and the capability of ANSYS software to easily generate very large detailed models for analysis. The memory usage problems in ANSYS are often most evident in the solution step where a memory allocation for a large contiguous block of memory may fail. This document will explain how to minimize this memory failure and give guidelines for determining the memory usage of ANSYS jobs.

Not all ANSYS solvers require contiguous space for memory, but most do. The important exception is the PCG and JCG solvers. These iterative solvers allocate memory for the large assembled matrices in blocks. The blocks do not have to be contiguous. The PCG solver can assemble matrices into one single memory block or as a set of 50 MB blocks, and will work equally well with either approach. Because the memory does not have to be in one contiguous block, the PCG solver can use a total memory size in excess of 2 GB if the Windows /3Gb switch is in use. The sparse solver cannot do this. This solver manages memory efficiently, but because of the design of the solver and the algorithmic differences between an iterative solver and a direct solver, a large contiguous block of memory is required. For PCG many people have good luck just using the default memory setting. ANSYS grows the memory as required and things go just fine. In fact, Workbench currently gets a fairly large space for the –db memory and then sets –m to be just 50 MB larger than the –db setting. This works fine for a lot of cases, including the sparse solver. But when jobs get larger, this approach is not optimal and it is often possible to solve a large model just by changing the initial memory settings.

It's too complicated to explain all the system interactions with memory that go on, but users should be aware that the Windows OS presents multiple challenges for large models in ANSYS. There is a fundamental limitation of 32-bit operating systems that limits the maximum contiguous block of memory to 2 GB. The total Windows memory for a single user application can be extended to 3 GB using the

The Focus

A Publication for ANSYS Users

the input file before the solve command so that you get sparse solver memory requirements and a look at the performance summary. Look at the memory available to the sparse solver and the memory required to run in-core. If the in-core memory requirement is within the capability of your machine's resources, you may be able to get it to run in-core just by increasing the –m number. That number, remember, is always total memory to ANSYS. You must subtract the –db space from that number, and anything else in use, before the sparse solver starts up to see what is left for the beginning of the sparse solver. Also, beware that we have memory heuristics built into our sparse solver that try to maximize performance while balancing memory limitations. Certain problems never run in-core no matter what — mostly problems which use pivoting for numerical stability. ANSYS also will not run in-core if the attempted memory allocation fails.

4. After all the preliminaries, I always try to run comfortably within the system memory resources available. If you try to hog them all you will probably pay in wall time. Especially on Windows! It is better to run in optimal out-of-core than to run in-core if the cost of running incore is to use the total physical memory of the system. Many ANSYS models in the 100,000 to 200,000 size range can run in-core using 600 to 800 MB of memory. This is where one should run in-core on Windows machines.

You should have 2 GB of real memory, period. Bare minimum. Then set the system page file for 3 GB. Use the /3Gb switch but be careful. Don't just add the /3Gb switch to the boot.ini file. Add the /3Gb switch to a separate copied line at the end of the boot.ini file so that you always have the option of booting Windows in normal mode or in the /3Gb mode. If the /3Gb mode fails, as it does on some XP releases, there are fixes available and you can always recover by rebooting and selecting the normal mode of operation. See Microsoft Support Article 328269 for a description of this problem.

Furthermore, you should get around 100 GB of disk space to run your ANSYS jobs, and don't put everything on your C drive. The ANSYS jobs directory should regularly be defragged and even reformatted to keep it clean. Move permanent files somewhere else after the jobs run. If you can get RAID 0 striping, that is great. It is available on home systems now for just a few hundred dollars. This allows users to have two drives sharing the ANSYS work directory, and the large file I/O can be done in parallel. SCSI drives are faster than IDE but cost more and typically are not as large. New Serial ATA drives configured for RAID 0 look to be a very attractive, cost-effective way to improve I/O for Windows workstations.

A Publication for ANSYS Users

The ultimate solution, of course, is to go to a 64-bit operating system. Windows64 is already available as a Preview, although it is still a very new OS and not all Windows drivers and applications are available. The highest performance systems are currently Intel Itanium 64-bit processors offered in Windows, Linux, and Unix OS machines from SGI and HP. IBM also has very high performance 64-bit workstation and server systems that run large models very well. The new AMD Opteron processor also offers excellent performance with 64-bit address space and very low cost.

*TAXIS! Improved Definition of Table Headings

by Jeff Strain, PADT

While teaching an Intro class last week, I came across a new APDL command called *TAXIS. "Wow!" I exclaimed, "An ANSYS command that calls a cab for you! I can't wait until they come out with *WASHDISHES and *MOWLAWN." This was met with some angry glares from the students, who were trying to do their workshops.

As it turns out, my initial interpretation was incorrect, surprisingly enough. *TAXIS, more accurately pronounced "star-tee-axis," was introduced in version 8.0 and is used to facilitate the definition of primary variable values for rows, columns, and planes (and books and shelves, if you're nutty like that) in a table array (i.e,. zeroth rows, columns, planes, etc.). The syntax is *TAXIS,*parmindex*,*nAxis*,*val1*,*val2*,…,*val10* where *parmindex* is the starting parameter index for defining your primary variables and *nAxis* designates the axis being labeled (1 for row, 2 for column, 3 for plane, 4 for book, and 5 for shelf). Val1 to val10 are the primary variable values.

As an example, consider the following table of temperatures vs. x- and y-locations.

		y-locations				
		0	1.2	2.4	3.1	4
x-locs	0	600	480	390	300	240
	1.8	540	420	360	240	210
	2.5	450	360	270	210	150
	3	300	210	180	120	60

Table 1. Sample table array TEMPTAB: Temperatures vs. X and Y.

To define this table the old fashioned way, you would have to define each column heading as shown in Figure 1.

The Focus

```
*DIM,TEMPTAB,TABLE,4,5,,X,Y        Row headings are here, but where are
TEMPTAB(1,0)=0,1.8,2.5,3           the column headings?
TEMPTAB(0,1)=0,600,540,450,300     — We're going to look here!
TEMPTAB(0,2)=1.2,480,420,360,210   Not only there, but we're going to look here!
TEMPTAB(0,3)=2.4,390,360,270,180   And then we're going to look here!
TEMPTAB(0,4)=3.1,300,240,210,120
TEMPTAB(0,5)=4,240,210,150,60      And here!
                                   And here!

    And then we're going to Washington, D.C. to take back
    the White House! YEEAARGH!!!
```

Figure 1. Table array definition prior to ANSYS 8.0.

Figure 2 demonstrates table array definition using *TAXIS.

```
*DIM,TEMPTAB,TABLE,4,5,,X,Y        How about row and column headings on this one?
*TAXIS,TEMPTAB(1,0),1,0,1.8,2.5,3  — Row headings are here.
*TAXIS,TEMPTAB(0,1),2,0,1.2,2.4,3.1,4  — Column headings are here.
TEMPTAB(1,1)=600,540,450,300
TEMPTAB(1,2)=480,420,360,210
TEMPTAB(1,3)=390,360,270,180
TEMPTAB(1,4)=300,240,210,120       Groovy.
TEMPTAB(1,5)=240,210,150,60
```

Figure 2. Table array definition made easier with *TAXIS.

In conclusion…wait, I just wrote three paragraphs about a single command. What is there to conclude? Nothing, really. So I'll just say, *yaaayy* *TAXIS! *TAXIS rocks!

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
Jan '05	1/20	1/21	801	ANSYS Customization with APDL	Tempe, AZ
	1/24	1/25	201	Basic Structural Nonlinearities	Irvine, CA
Feb '05	2/7	2/9	101	Introduction to ANSYS, Part I	Irvine, CA
	2/10	2/11	202	Advanced Structural Nonlinearities	Tempe, AZ
	2/14	2/16	104	ANSYS Workbench Simulation - Intro	Tempe, AZ
	2/17	2/18	301	Heat Transfer	Tempe, AZ
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

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.

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.