The Focus

June 16, 2008 | A Publication for ANSYS Users | Issue 65

FITEMizing Entities for Fun and Profit

By Jeff Strain

If you've ever looked through an ANSYS log file or pulled up the session editor, you've probably noticed the commands FLST and FITEM preceding any command that invokes the picker. You probably also figured that it's just some funky internal ANSYS thing that you would ever use for APDL scripting. While it's true that we can usually create APDL macros without utilizing the FLST and FITEM commands, this is not always the case.

Let's say you're using a command where you want to be able to exceed the maximum number of entities the command syntax allows. Let's also say you need to designate these entities in a specific order. What would you do? A good example of this is the BSPLIN command, which only allows six keypoints to be specified. How would you program an APDL macro to create an airfoil through the thirteen keypoints in Figure 1?

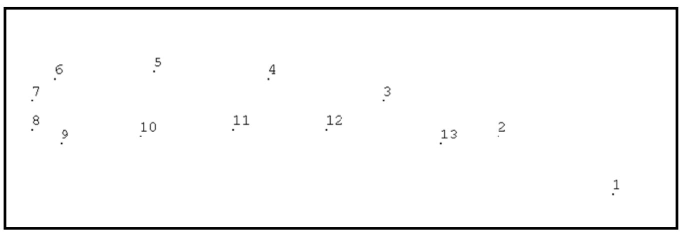


Figure 1: Airfoil Keypoints I Freehanded all by Myself

You could always try to force additional keypoints into the BSPLIN command by issuing BSPLIN,1,2,3,4,5,6,7,8,9,10,11,12,13, but there's a problem: After the sixth keypoint the arguments represent x, y, and z locations. Figure 2 is what you end up with.

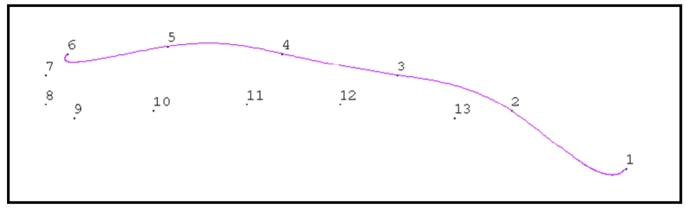


Figure 2: Yuck!

How about BSPLIN,ALL? Let's give that a go.

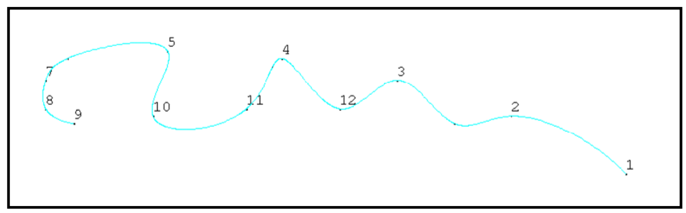


Figure 3: #@*!!#!

Now what? Remember the picking option? That lets you pick as many entities as you want doesn't it? How? The key is in the commands FLST and FITEM. FLST is used to define the settings associated with the command you'll be executing. The FLST command is followed by a series of FITEM commands that specify the entity numbers in order, to be used in the command you wish to execute. The FITEM list is called by placing the P51X label in the appropriate field of the command. The following paragraphs discuss FLST and FITEM in detail and how to use them for this example.

(Cont. on pg. 3)

Faster Solves with VT Accelerator

By: Eric Miller

When ANSYS, Inc. bought a little company called CADOE a while back they not only acquired a very talented group of developers and engineers, but they also got access to a whole family of technologies that can be used to create mathematical representations of data that interpolates and extrapolates very accurately. They call this collection of algorithms VariationalTechnology, or VT for short. Most people have run into VT when doing probabilistics as a much more efficient way to map a design space in DesignXplorer VT.

In version 11 ANSYS, Inc. expanded the application of VT towards speeding up the solution of non-linear structural and thermal problems along with harmonic dynamics problems. The tool is called VTAccelerator and it is available to anyone with a HPC license for ANSYS. It can significantly speed up non-linear solutions by taking past convergence data, mapping it, and predicting a convergence path that requires fewer equilibrium integrations. Sounds pretty simple but it actually has some harry, and proprietary, math and bookkeeping behind it.

(Cont. on pg. 2)

Table of Contents

VT Acel cont...)

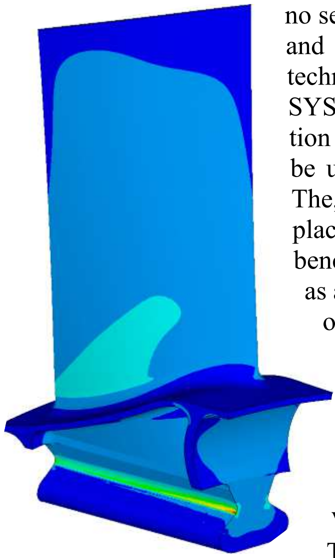


Figure 1: Fan Blade

The nicest part is that there is almost no setup required, just an HPC license and a model that benefits from the technology. When you start up ANSYS, you need to use the –VTA option to tell the licence server you will be using VT accelerator technology. The, for stress and thermal, simply place STAOPT,VT into your Workbench model as a command snippet or as an ANSYS command and you are off and running. For a transient analysis use TRAOPT,VT. Other than a faster convergence, the only change in your run will be the creation of some files used by the VT Accelerator algorithm.

What type of Speedup?

The ANSYS documentation says that you can expect a 2x to 5x speedup for standard runs and up to 10x on a restart run. Our testing points towards this as being the type of speedup you can expect on a substep that benefits strongly from VT Accelerator. If you have a lot of substeps that benefit, you will see these types of numbers. Also, if you have a model that struggles a bit to converge, things are even better. To be honest, we wanted to have a lot of benchmarking to share with everyone but the days just kept marching by and we just never had the time to run a bunch of cases. And for real work, we didn't have the time to run with and without to compare. So we ended up running two pretty typical, but tame models that showed good speedup, but not nearly as high as really troublesome models can generate. Maybe we will find time to do an update with more test cases in a future issue.

Equilibrium Iters per Substep

SubStep	No VTA	VTA	Diff
1	5	5	0
2	5	5	0
3	4	4	0
4	5	2	-3
5	4	1	-3
6	3	1	-2
7	3	1	-2
	29	19	-10

CPU Seconds Per Substep

SubStep	No VTA	VTA	% Diff
1	1991	1769	-11%
2	1657	1695	2%
3	1421	1521	7%
4	1684	1170	-31%
5	1391	649	-53%
6	1206	661	-45%
7	1150	659	-43%
	10500	8124	-23%

Table 1: Improvements in Speed

Figure 1 shows a sample Turbine Engine fan blade that was used for the first benchmark. A typical thing to do with this type of model is a large deflection static run with rotational (spin) loading. The blade is thin and tall enough to twist and bend that goes beyond small deflection assumptions. As Table 1 shows, turning on VT accelerator reduced run time by 23% and the number of iterations required from 29 to 19. The graph in Figure 2 shows how the convergence path changes.

The second benchmark model was a transient thermal analysis on one of the blower housings that PADT makes for hydrogen fuel

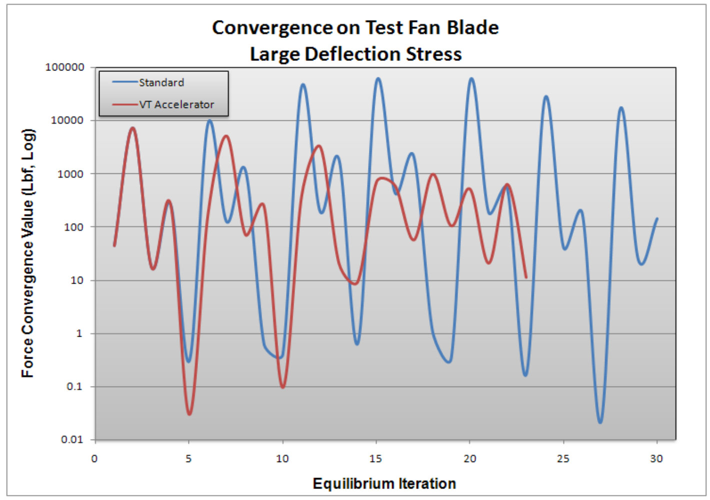


Figure 2: Convergence History

cell systems. Again, this problem is well behaved and converges quickly without VT. Even so, turning on VTA gave us a 12% reduction in CPU time, as is shown in Table 2. Well worth flipping it on if you have the licenses.

Faster Reruns

CPU Time Per Loadstep

LS	No VTA	VTA	% Diff
1	1342.4	1225.3	9%
2	2110.7	1847.4	12%
3	2071.7	1790.3	14%
4	1288.8	1116.9	13%
	6813.6	5979.9	12%

Table 2: Transient Thermal Speedup

One of the biggest benefits of VTA is that if you turn it on when you first run a problem, then make a change to your model that doesn't change the mesh (change material property, add/delete/change loads, change real constants or section properties, or morph the mesh) the solver uses the data from the previous run to speed up the convergence the second time. In order to take advantage of this you need the *.DB, *.ESAV and *.RSX files from the first run. Then, specify you are doing a restart with ANTYPE,,VTREST, specify your loads then solve as usual. This

(Cont. on pg. 3)

FITEM cont...)

FLST has the syntax
FLST,*NFIELD,NARG,TYPE,Otype,LENGTH*, where:

NFIELD: The field number in which the entity input is used, including the command itself. This field will have a label of P51X in the command.

NARG: The number of picked entity items. This should be equal to the number of following FITEM commands.

TYPE: The entity type: 1 for nodes, 2 for elements, 3 for keypoints, etc.

Otype: Designates whether the FITEM data is not ordered (NOOR) or in order (ORDER)

LENGTH: Is the length of number of items describing the list (defaults to NARG)

Defaults are generally used for the last two arguments.

NOTE: For the BSPLIN command, I discovered that this only works if you use P51X as the second argument, not the first. The example here reflects that.

In our little airfoil case, the FLST command will be:

FLST,3,13,3 !Second argument = third field. See above note.

We follow this with a number of FITEM commands using the syntax

FITEM,*NFIELD,ITEM,ITEMY,ITEMZ*, where:

NFIELD The same thing here as in the FLST command.

ITEM is the entity number to be picked. Also X location, if applicable.

ITEMY and *ITEMZ* are Y and Z locations, if applicable.

The series of FITEM commands for our case would be:

```
FITEM,3,1
FITEM,3,2
FITEM,3,3
FITEM,3,4
FITEM,3,5
FITEM,3,6
FITEM,3,7
FITEM,3,8
FITEM,3,9
FITEM,3,10
FITEM,3,11
FITEM,3,12
FITEM,3,13
```

And now, the pièce de résistance, the command we were trying to execute to begin with.

```
BSPLIN,,P51X
```

Or, to make it more brief and generalized:

```
kpmax=kpinqr(0,14)
FLST,3,kpmax,3
FITEM,3,(1:kpmax)
BSPLIN,,P51X
```

This will give you something that looks much prettier.

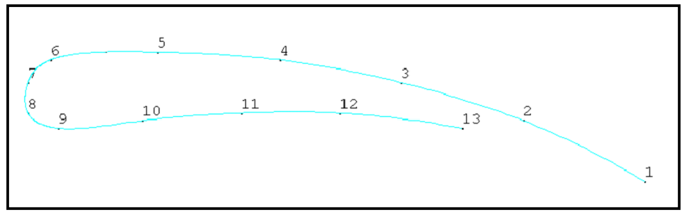


Figure 4: An Airfoil! Almost!

While you could respecify keypoint 1 at the end of the FITEM list, you would end up with a deformed blob at the trailing edge of the airfoil if you did. It's better to leave a gap and then close it with a final BSPLIN or L command, e.g. L,13,1.

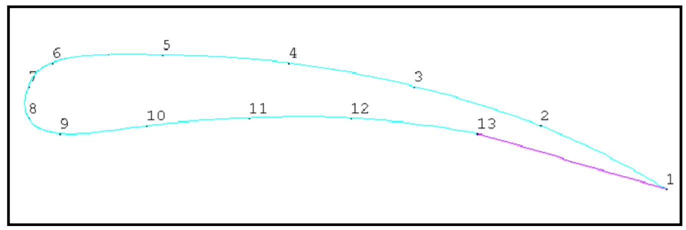


Figure 5: A Reasonable Facsimile of an Airfoil

Eh, it'll do as an example. Obviously if you were analyzing a real-life airfoil, you'd want to use more precise keypoint locations than I have.

FLST and FITEM are useful whenever you are scripting a command where the number of arguments is limited and the order of those arguments is critical. The most obvious example to me so far has been BSPLIN, but undoubtedly there are others out there.

VT Accel. cont...)

method gives good speedups even for contact and plasticity problems, which don't benefit for a first run with VTA on.

What problems does it work on?

The only thing to worry about is if your particular problem works with VT Accelerator. According to the manual, it is recommended for: "large deflection, hyperelasticity, viscoelasticity, and creep nonlinearities. The manual also points out that it has little or no effect on "Rate-independent plasticity and nonlinear contact analyses". But, as noted above, if you are going to do multiple runs which non-mesh changes, you can benefit from STA on the further runs.

Learning more

Information on VTA is a bit scattered in the manual. The best place to start is Chapter 4 of the Advanced Analysis Techniques Guide. This gives a good overview. It also discusses other uses for VTA including faster Harmonic Analysis and DesignXplorer. Also check out the STAOPT and TRNOPT commands.

HPC Update New Intel Architecture is Moist, Chewy

By Jason Krantz

At PADT, we value additional processors the same way kindergartners value fresh-baked chocolate-chip cookies. It turns out that Intel has been baking some particularly tasty cookies recently. Chips built on Intel's Nehalem architecture, the successor to the current Penryn architecture, are due out around November 2008. (Current Penryn chips include Xeon and Core 2 Duo/Quad processors).

Anandtech, a technical computing web site, recently got its hands on some Nehalem prototypes, and Anandtech's benchmarks make the upcoming Nehalem products look very interesting for both structural and CFD analysts. Here are a few highlights:

- More cores per socket (up to 8)
- An on-die memory controller; this eliminates the traditional front-side bus. AMD has benefited from a similar configuration since 2003. This improvement leads directly to the next one:
- Vastly improved memory bandwidth (~13 GB/s using triple-channel DDR3, roughly 70% more than current configurations).
- A larger, better on-chip cache, including an L3 cache (Penryn chips have no L3 cache).
- Better clock-for-clock performance; Anandtech's 2.66 GHz Nehalem chip was about 28% faster on video encoding benchmarks than a 2.66 GHz Penryn chip—faster even than Intel's Core 2 Extreme chip at 3.2GHz. Although video encoding involves a fair amount of matrix manipulation, it may or may not be a good stand in for engineering analyses. However, the numbers are promising.

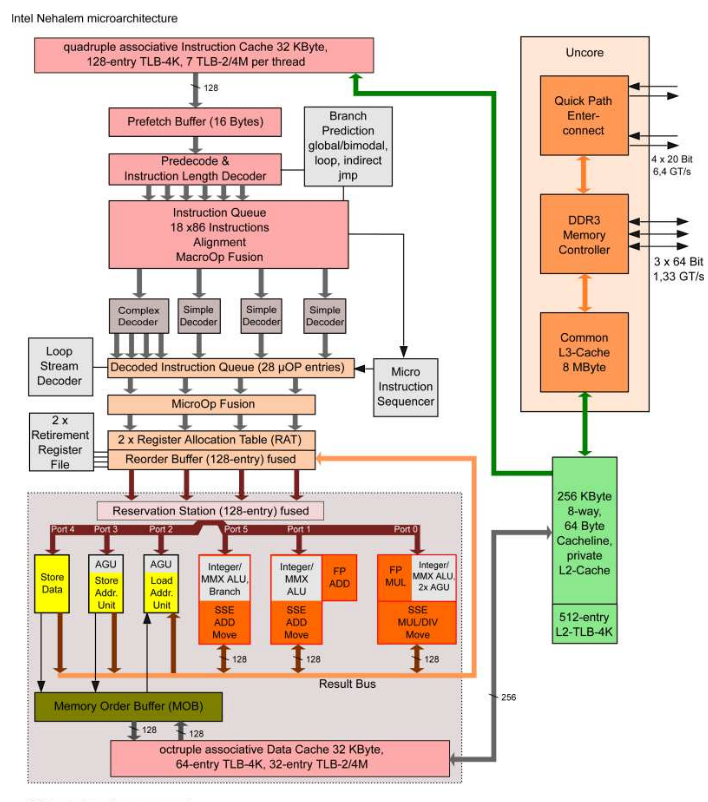
It seems that future Intel chips will be faster; this is barely news, right? Well, there's more to it than that. Since the Nehalem chips will have up to eight cores per processor, it may well be possible to get a true eight-core machine built with desktop-level (single-socket) components. These parts are generally much cheaper than server-level components such as Xeon processors and multi-socket motherboards.

Current DDR3 desktop memory prices, about $50/GB, are roughly comparable to current server memory prices (i.e., 800-MHz FB-DIMMs). However, DDR3 prices will likely fall significantly as DDR3 becomes the new standard, perhaps to the $33/GB price point of current DDR2 RAM. That means that in six to nine months, it might be possible to build an eight-core, 16-GB RAM machine for $1500-$2000.

Power consumption is only getting more important to corporations as energy prices rise. Anandtech's quad-core Nehalem sample consumed about 10% more power under full load than a current quad-core Penryn chip at the same clock speed. If Nehalem machines really do run 20-30% faster than current ones under real-world conditions, solving power per watt will actually increase significantly. It's worth remembering that one paysCPU watt-hours twice: once to run the computer and (roughly) once to cool the server room.

Naturally, there are caveats. Significantly, the Nehalem platform will mark the return of Intel's Hyper Threading concept, which was last seen on Pentium 4 processors. (Essentially, Hyper Threading doubles the number of processors the operating system sees, so if you're using one eight-core Nehalem chip, your OS will tell you that you have 16 processors). PADT has seen mixed results when using P4-era Hyper Threading; under many conditions, our models solve fastest when we've disabled Hyper Threading. Whether Intel will allow users to disable Hyper Threading is a bit unclear, but the re-implemented Hyper Threading option may operate more happily with CAE applications than the previous iteration did.

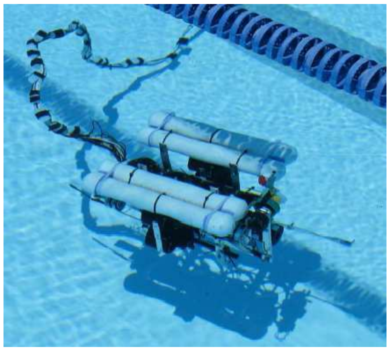
So analysts may want to delay purchases of hardware until late this year or early next. Of course, microprocessor companies are always whipping up faster, more tempting hardware—but this next batch of silicon-chip cookies may be particularly tasty.



The Architecture Schematic from Wikipedia

Editorial Comments

Science and Technology Education not Dead Yet



By: Eric Miller

We didn't have time to come up with a good Awesome APDL so we needed something to fill this space. Then I remembered some thoughts I had last weekend and figured "hey, blathering out your opinion fills spaces just as well as pictures from Doug's last vacations."

It is sort of popular these days to spout off about how horrible our education system is and how "kids today just can't cut it." or to quote statistics on how many engineers are being created in this country or that. And I've taken part in some of that grumbling. But last weekend I was asked to be a judge in an engineering competition for elementary through college students called the "National Underwater Robotics Competition" or NURC. In this competition teams build their own Remotely Operated Vehicles (ROV's) and use them to accomplish a "mission" that usually involves picking stuff up and hitting switches in a dark pool.

What amazed me is how innovative, creative and dedicated these kids were. The participants came from almost every ethnic and economic group you can find in this country, and they all showed the same "Yankee ingenuity" that grumpy old people like to say is dead and gone. I have to admit that I had tears welling up in a couple of oral presentations because I was so proud of these kids.

My favorite was a 7th grader who said "When we started, all we had was a box of parts and we didn't even know what the word solder meant. When we were done we knew how electricity works, how motors work and Jimmie, he is an expert at soldering now"

At the end of each presentation the judges try to ask if they students are interested in going into science, math or engineering. Almost every student raised their hand to say yes. One said, another eye welling moment, said a something similar to "before I did this project, I didn't think I could be anything like an engineer, but now I now I now I have what it takes and I can be the first person in may family to go to college"

Is there room for improvement in the US's education system? Yes. Do we need to find a way to encourage our best and brightest to go into technology careers? Yes. Is the future bleak and hopeless? No, not if the high school and college kids who showed up with their hand made ROV's are any indication.

Learn more at : www.h2orobots.org

News - Links - Info

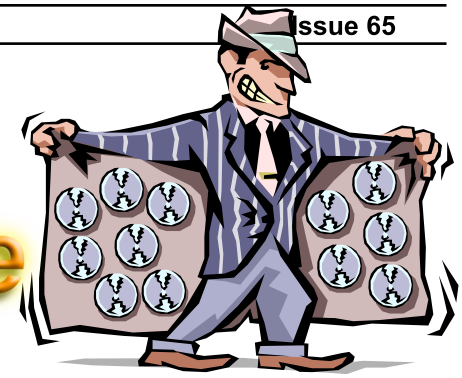
- Looking for a time killer while waiting for convergence? Check out eh Simulation Jigsaw Puzzles on the ANSYS site <link>
- New version of T-Grid is available <link>
- ANSYS, Inc. Is Hiring! Check out job opportunities on their careers page <link>
- Get ANSYS news directly by subscribing to their RSS feed: <link>
- PADT has been getting most of their new computers from the Dell refurb site. Big savings: <link>

Upcoming Training Classes

Month	Start	End	#	Title	Location
Jun '08	6/19	6/20	301	Heat Transfer	Tempe, AZ
	6/29	6/30	107	ANSYS Workbench DesignModeler	Tempe, AZ
Jul '08	7/8	7/9	104	ANSYS WB Simulation – Intro	Albq, NM
	7/10	7/11	205	ANSYS WB Simulation Dynamics	Albq, NM
	7/14	7/15	801	ANSYS Customization with APDL	Tempe, AZ
	7/21	7/22	203	Dynamics	Tempe, AZ
	7/24	7/25	102	Introduction to ANSYS, Part II	Tempe, AZ
Aug '08	8/4	8/5	104	ANSYS WB Simulation – Intro	Tempe, AZ
	8/6	8/8	152	ICEM CFD/AI*Environment	Tempe, AZ
	8/11	8/12	202	Advanced Structural Nonlinearities	Tempe, AZ
	8/14	8/15	501	ANSYS/LS-DYNA	Tempe, AZ
	8/18	8/19	604	Introduction to CFX	Tempe, AZ
	8/28	8/29	701	Design Optimization & Probabilistic Design	Tempe, AZ
Sep '08	9/4	9/5	301	Heat Transfer	Tempe, AZ
	9/8	9/10	101	Introduction to ANSYS, Part I	Tempe, AZ
	9/11	9/12	102	Introduction to ANSYS, Part II	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



Got CFD?

See why over 10 companies over the past 12 months have sent their CFD consulting work to PADT

- Experience
- Partnership
- Efficiency
- Speed

Contact Steve Hendry at steve.hendry@padtinc.com for more information

www.XANSYS.org

The largest independent Online Community for ANSYS users just got an upgrade and a face lift
Join over **3333** other users online for information and occasional attempts at humor