Date October 14, 2001 Memo Number STI:01/04
Subject **ANSYS Tips and Tricks: Sparse Direct Solver Improvements**
Keywords Solvers, Sparse direct solver, Linear static

## 1. Introduction:

There have been various sparse solver improvements in each release of ANSYS. This memo hopes to cover some general information on the solvers available in ANSYS as well as performance improvements between ANSYS 5.6 and 6.0 related to linear static analyses.

## 2. Background on the ANSYS Direct Solvers:

The sparse solver is a direct solver which uses a sparse storage scheme. In ANSYS, this solver is accessed via EQSLV,SPARSE. Although for linear static problems of 3D meshes, the PCG solver may provide better performance, the author will use the simple case of linear static problems to discuss the sparse solver.[1]

For linear static problems, the familiar equation is solved for:

$$[K]\{x\}=\{F\}$$

*Direct solvers* (or, more appropriately, *direct elimination solvers*) such as the frontal and sparse solvers factorize [K], then back-substitute to solve for {x}.

The frontal solver assembles the [K] matrix then basically triangularizes [K] via Gaussian elimination. This results in small memory requirements but *large disk space needs*, as the triangularized matrix (jobname.tri) can become excessively huge as the model size increases. The large disk space requirement is why the frontal solver is recommended for systems with 50-100k DOF or less.

Unlike the frontal solver, the sparse solver factorizes [K] using a sparse matrix storage scheme to minimize memory requirements. In other words, only non-zero coefficients of [K] are stored in memory. However, the resulting factorized matrix can still take up large memory and/or disk requirements (jobname.LN09). Hence, the sparse solver is recommended for systems less than 500k DOF.

A common situation which comes up for direct solvers is running the solution in-core or out-of-core. Running a solution *in-core* means that the problem fits in memory (physical RAM+virtual memory), and running problems in physical RAM will be much faster. Running an *out-of-core* solution means that some of the problem is in memory, but the rest is written to disk. This results in a slower solution because of the fact that there is much disk I/O taking place.

The frontal solver can control in-core vs. out-of-core solutions with the /CONFIG command (NBUF and SZBIO parameters). Although a detailed discussion of this is outside the scope of this memo, the basic idea is to create large enough buffers with /CONFIG to fit the jobname.tri and other scratch files in memory (note that there is a limit of NBUF*SZBIO of 512 MB).

On the other hand, the sparse solver will run a solution in-core if the scratch space is large enough (workspace [-m] minus database space [-db]). The sparse solver will print out information in the output window notifying the user of how much memory is required for in-core and out-of-core solutions:

```
SPARSE MATRIX DIRECT SOLVER.
 Number of equations = 449340,    Maximum wavefront =    296
 Memory available for solver =             1009.70 MB
 Memory required for in-core =             3158.87 MB
 Optimal memory required for out-of-core =  211.14 MB
 Minimum memory required for out-of-core =   41.04 MB
```

As shown above, in this situation, 3 GB is required for in-core whereas only 211 MB is required for out-of-core solution. It is best to supply as much memory to ANSYS as possible, even if the solution runs out-of-core. Also, if the problem cannot run in-core, ensure that at least the same amount of space is available on the hard drive (or use /RUNST to get *.LNxx file size estimates).

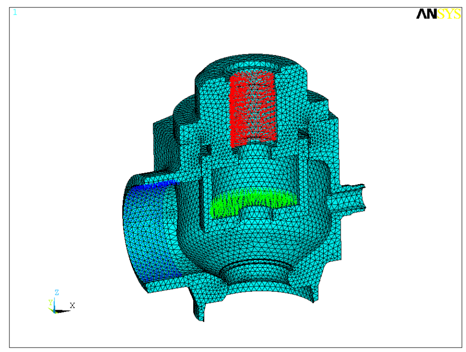
[1] For some information on the PCG solver, please refer to the Reference #1

## 3. ANSYS Sparse Direct Solver Enhancements:

The sparse direct solver is well-suited for nonlinear applications with ill-conditioned matrices. The sparse solver is also used with the Block Lanczos eigenvalue extraction method (linear buckling or modal analyses).

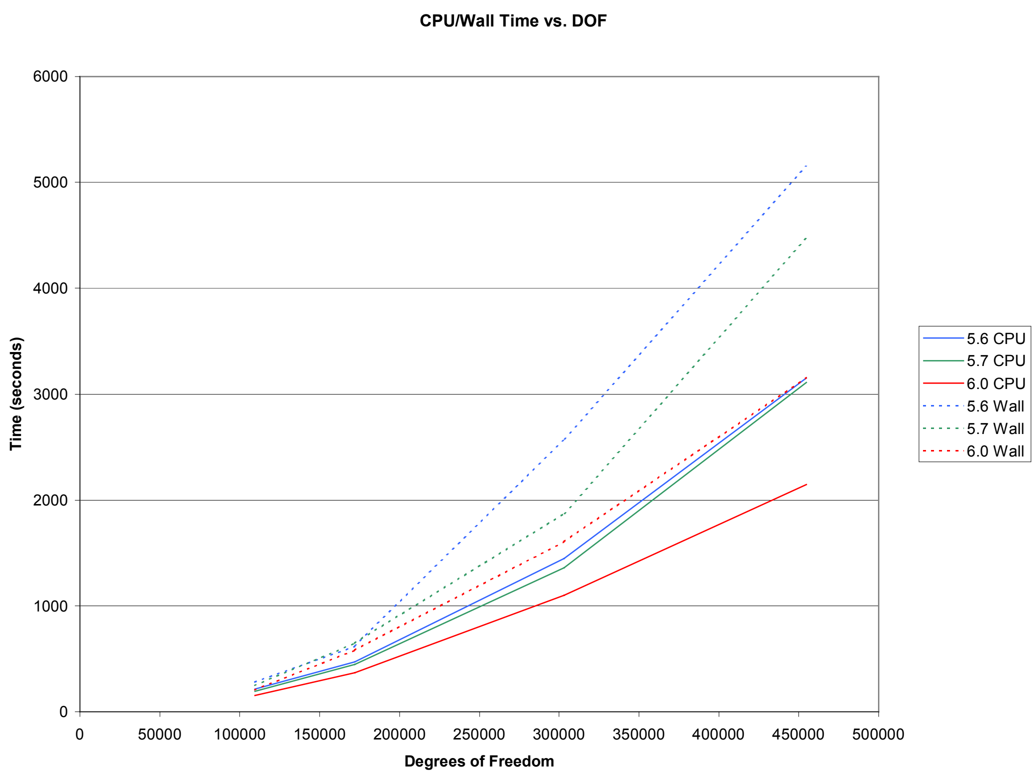
In the past few releases, various enhancements made to the sparse solver in harmonic, modal, and static/transient analyses have increased the performance of this solver. Specifically, for linear static analyses:[2]

- At 5.7, the *.LN22 file did not contain a copy of the workspace, so this reduced disk space requirements and wall times considerably.
- At 6.0, sparse assembly of matrices was added for the sparse direct solver (already existed for iterative solvers such as PCG, AMG, and JCG). This provided a speedup, especially for processing of constraint equations when assembling the [K] matrix prior to solution.



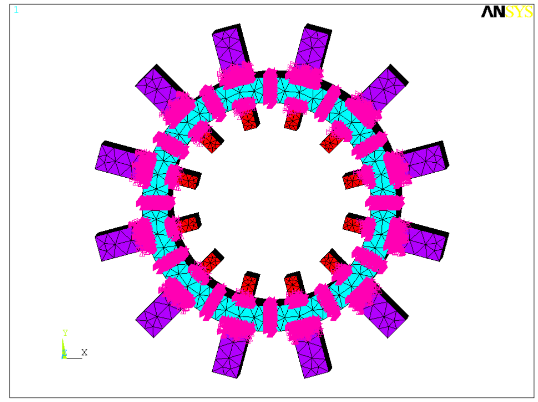

Some benefits of the sparse solver can be viewed with a simple linear static analysis of a model meshed with SOLID92 elements, as shown on the right. The same model was solved in ANSYS 5.6.2, 5.7.1, and 6.0 with the sparse solver with 4 different mesh densities. A workspace of 1200 MB and a database size of 100 MB were specified for the runs.

As shown on the graph below, the 5.6 CPU and wall times, designated by the blue lines, are longest (i.e., slowest solution). At 5.7, there is noticeable improvement in the wall time, as indicated by the green lines, because of the changes to the *.LN22 file (CPU time between 5.6 and 5.7 is still similar). There is further CPU and wall time decrease at 6.0 (red lines), due to improvements such as sparse assembly.




[2] We will focus on the linear static case, and, hopefully, a later memo may address improvements in other areas, such as eigenvalue extraction.

Another model with constraint equations, as shown on the right, was run. This consisted of 150,000 DOF with 9,000 constraint equations. The solution ran in-core with the following results (in seconds):




| Version | Total CPU | Total Wall |
|---|---|---|
| 5.6 | 510 | 593 |
| 5.7 | 473 | 559 |
| 6.0 | 281 | 371 |

Similar trends with the previous example can be seen here, with results at 6.0 being much more favorable than at 5.6 or 5.7.

The effect of sparse assembly and processing of constraint equations can be seen more readily in other analyses, namely modal cyclic symmetry problems with many constraint equations.

## 4. Conclusion:

Continual improvements in the sparse solver allow for faster solution times for any analysis using the sparse solver. While only the simple case of linear static analyses was shown in the present memo, many of the trends are also reflected in nonlinear, harmonic, and modal analyses, which the author hopes to address in a later memo.

The author wishes to stress that the user should not *always* use the sparse solver. Instead, *solver selection is dictated by the problem type*. The user should refer to Ch. 3.2 of the ANSYS 6.0 "Basic Analysis Procedures Guide" for more details on solver selection.

## 5. References:

1. S. Imaoka, "ANSYS Tips and Tricks: Memory Management and Faster Solution Times," http://ansys.net/ansys/tips/Week%2014%20-%20Memory%20Configuration.pdf
2. ANSYS 6.0 Theory Manual, Section 15.8 "Equation Solvers"

Sheldon Imaoka
ANSYS, Inc.

*This document is not being provided in my capacity as an ANSYS employee. I am solely responsible for the content.*

## Sheldon's ANSYS Tips and Tricks

Sheldon's ANSYS Tips and Tricks will be emailed to subscribers about once a week (or whenever I have time to write one of these up). You can subscribe by visiting the following URL:

http://ansys.net/ansys/ansys_tips.html

Archives will be posted on that page with password access (which will be mailed with each new issue). General ANSYS Tips and Tricks can also be found at the above URL.

## ANSYS Training

ANSYS, Inc. as well as the ANSYS Support Distributors (ASDs) provide training classes in many different areas of ANSYS, ranging from Introductory classes to CFD to Structural Nonlinearities to Electromagnetics. Information on training classes and schedules at ANSYS, Inc. can be found at:

http://www.ansys.com/services/training/index.htm

New classes which will be offered soon (or offered already) include High-Frequency Electromagnetics and Advanced Structural Nonlinearities. Please contact ANSYS, Inc. or your local ASD for their training schedules and offerings.

## XANSYS Mailing List

The XANSYS mailing list has more than 2300 subscribers (as of 9/25/01) with about 40 postings per day. This is a forum for exchanging ideas, providing/receiving assistance from other users, and general discussions related to ANSYS. (Note that it is recommended to contact your local ASD for issues related to technical support) You can sign up by visiting the following URL:

http://groups.yahoo.com/group/xansys

Otherwise, you can also subscribe/unsubscribe by sending an email to the following address:

```
Post message:  xansys@yahoogroups.com
Subscribe:     xansys-subscribe@yahoogroups.com
Unsubscribe:   xansys-unsubscribe@yahoogroups.com
List owner:    xansys-owner@yahoogroups.com
```

Because the amount of emails is very large, you can also subscribe in "digest mode" or access the postings via a web browser instead:

```
Digest mode:   xansys-digest@yahoogroups.com
Web-based:     xansys-nomail@yahoogroups.com
               http://groups.yahoo.com/group/xansys/messages
```