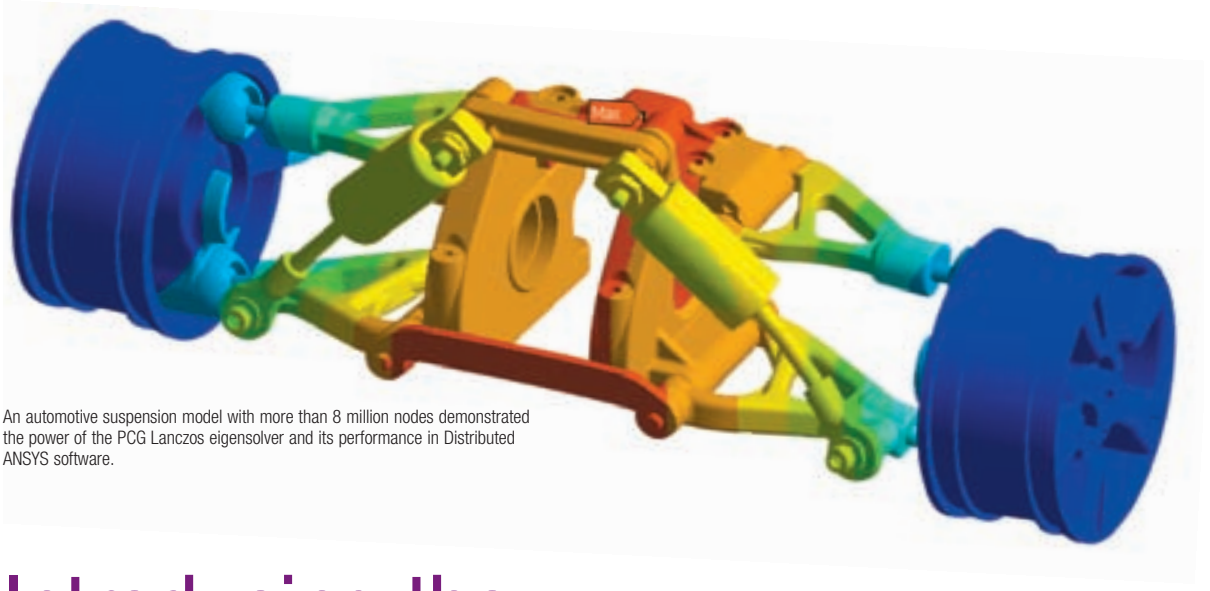


An automotive suspension model with more than 8 million nodes demonstrated the power of the PCG Lanczos eigensolver and its performance in Distributed ANSYS software.

Introducing the PCG Lanczos Eigensolver

A new eigensolver in ANSYS 11.0 determines natural frequencies and mode shapes using less computational power, often in shorter total elapsed times than other tools on the market.

By Jeff Beisheim, Senior Development Engineer, ANSYS, Inc.

Eigensolvers determine natural frequencies and mode shapes of structures, typically some of the most computationally demanding tasks in structural analysis. ANSYS 11.0 software improves upon its extensive selection of these tools with a new eigensolver that combines the speed and memory savings of the ANSYS PCG iterative solver with the robustness of the Lanczos algorithm. This powerful combination allows users to solve for the natural frequencies and mode shapes of their model using fewer computational resources, often in shorter total elapsed times than other eigensolvers available on the CAE market.

Starting Up PCG Lanczos

The PCG Lanczos eigensolver can be selected using the LANPCG label with the MODOPT command (or in the Analysis Options dialog box). ANSYS Workbench users can choose this eigensolver simply by selecting the Iterative option in the Analysis Settings details pane.

The PCG Lanczos eigensolver works with the PCG options command (PCGOPT) as well as with the memory saving feature (MSAVE). Both shared- and distributed-memory parallel performance can benefit from this eigensolver. Note that in version 11.0, Distributed ANSYS software supports modal analyses, and the PCG Lanczos eigensolver is the only eigensolver available that runs in a distributed fashion in this software.

Controlling the Parameters

The PCG Lanczos eigensolver can be controlled using several options on the PCGOPT command. The first of these options is the Level of Difficulty value (Lev_Diff). In most cases, choosing the default value of AUTO for Lev_Diff is sufficient to obtain an efficient solution time. However, in some cases you may find that manually adjusting the Lev_Diff value further reduces the total solution time. Setting higher Lev_Diff values (e.g., 3 or 4) can help for problems that cause the PCG solver to have some difficulty in convergence. This typically occurs when elements are poorly shaped or are very elongated. A new Lev_Diff value of 5 is available in version 11.0 and warrants further explanation.

A Lev_Diff value of 5 causes a fundamental change to the equation solver being used by the PCG Lanczos eigensolver. It replaces the PCG iterative solver with a direct solver similar to the Sparse direct solver making the PCG Lanczos eigensolver behave more like the Block Lanczos eigensolver. Due to the amount of computer resources needed by the direct solver, choosing a Lev_Diff value of 5 essentially eliminates the reduction in computer resources obtained by using the PCG Lanczos eigensolver compared to using the Block Lanczos eigensolver. Thus, this option is generally only recommended for problems that have less than 1 million degrees of freedom (DOF), although its efficiency is highly dependent on several factors such as the number of modes requested and I/O performance, for example. A larger number of modes may increase the size of problem for which a value of 5 should be used while slower I/O performance may decrease the size of problem for which a value of 5 should be used.

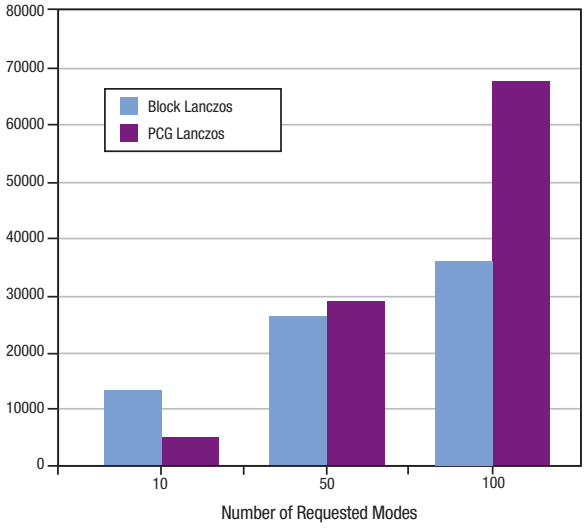
While the PCG Lanczos eigensolver has many innovative algorithms to guarantee that no modes are missed, another new option on the PCGOPT command allows users to force a Sturm sequence check that helps guarantee that this is the case. It is important that users keep in mind that a Sturm sequence check works by using a direct solver to perform a numeric factorization after the Lanczos algorithm finishes. This factorization involves a large number of computations. It should also be noted that if using a Lev_Diff value of 1 through 4, executing a Sturm sequence check will eliminate the savings in computer resources obtained by using the PCG Lanczos eigensolver compared to using the Block Lanczos eigensolver.

Guidelines on Using the Tool

Typically, the following conditions must be met for the PCG Lanczos eigensolver to be most efficient:

- The model would be a good candidate for using the PCG solver in a similar static or full transient analysis.
- The number of requested modes is less than 100.
- The beginning frequency input on the MODOPT command is zero (or near zero).

Like all iterative solvers, the PCG Lanczos eigensolver is most efficient when the solution converges quickly. If a model would not converge quickly in a similar static or full transient analysis, the PCG Lanczos eigensolver would not be expected to converge quickly either and would therefore be less efficient. Other factors such as the size of the problem and the hardware being used can affect the decision of which eigensolver to use. For example, on machines with slow I/O performance or limited hard drive space, the PCG Lanczos eigensolver may be the better choice.



A comparison of the Block Lanczos and PCG Lanczos solvers for a problem with 1 million degrees of freedom. The simulations were run on a Dell® 670 workstation running Windows® x64 with a 3.6 GHz single-core Xeon® EM64T chip and 4 GB RAM.

Comparing Eigensolvers

The accompanying chart shows a sample comparison of solution times between the Block Lanczos and PCG Lanczos eigensolvers for a 1-million DOF model. When requesting 10 modes for this model, the PCG Lanczos eigensolver is faster. At higher numbers of modes, however, the Block Lanczos eigensolver becomes the faster option.

In other runs, as problem size increased the PCG Lanczos eigensolver was faster than the Block Lanczos eigensolver at a higher number of requested modes. For example, when solving this model with 500,000 DOF, the PCG Lanczos eigensolver was faster when computing less than 20 modes. At 1 million DOF, the PCG Lanczos eigensolver was faster when computing less than 40 modes.

Parallel Performance in Distributed ANSYS

A suspension model with more than 8 million nodes was used to demonstrate the power of the PCG Lanczos technology along with the performance of this eigensolver in Distributed ANSYS. This model was solved on an SGI® Altix® Linux server with Itanium®-II processors. Requesting 10 modes with a model of this size would likely take several days using the Block Lanczos eigensolver.

Using one processor, the PCG Lanczos eigensolver was able to find the requested modes in 37 hours. Using 10 processors dropped the elapsed time to six hours, demonstrating how the combination of the powerful PCG Lanczos algorithm along with the parallel performance of Distributed ANSYS makes the PCG Lanczos eigensolver one of the fastest eigensolvers available in the CAE market today. ■