A New Parallel Lanczos Eigensolution Technique in MSC.Nastran Version 2001

Louis Komzsik
Chief Numerical Analyst
MSC.Software Corporation

ABSTRACT

The Lanczos eigensolution takes the dominant amount of time in most normal modes and modal frequency response analyses. This paper introduces a way to reduce the elapsed time of large eigenvalue jobs by using a newly developed parallel option within the Lanczos algorithm. We will compare the performance of the frequency domain decomposition based distributed memory parallel method introduced in V70.7 and the new, geometry domain decomposition based distributed parallel method introduced in Version 2001. The new method may be utilized on clusters of workstations or even PCs, and naturally on distributed or shared memory parallel computers. It is applicable to very large NVH models of the automobile industry.

INTRODUCTION

The Lanczos method has been available in MSC.Nastran since 1985 (Version 65). Almost a decade ago MSC offered the first parallel execution of the Lanczos method, based on the shared memory paradigm. At that time we focused on parallelizing the computations in several expensive steps, for example the matrix decomposition. However, during recent years it has become clear that a more extensive parallelization of the Lanczos method is needed to satisfy the users' ever growing demand for higher performance. We have also found that not only computations, but also the I/O operations need to be parallelized in order to obtain a higher efficiency in parallel eigenvalue analysis.

MSC started to work on new parallelization approaches late in the last decade; this time based on the distributed memory paradigm to be able to address parallel I/O issues as well. First successes were obtained on linear static analyses (see Ref. 1). Encouraged by these results, the efforts on distributed parallel MSC.Nastran have been intensified during the past two years. This has resulted in the frequency domain parallel Lanczos introduced in Version 70.7 (see Ref. 2). Our newest offering, based on the geometry domain decomposition principle is available in the new Version 2001.

1. Parallel Lanczos eigensolution techniques in MSC.Nastran

In order to understand the new geometry domain parallel method, first the already available frequency domain parallel method is briefly reviewed.

1.1 Frequency domain decomposition in Version 70.7

The frequency domain decomposition technique is advantageous in cases where the problem size is moderately large, but the frequency spectrum of interest is very wide. The frequency domain decomposition style Lanczos method is based on a frequency segment approach in which automatically created segments of the user defined frequency spectrum (F1, F2 on the EIGRL entry) are assigned to different nodes of the parallel environment. The user also has the choice of specifying the segments directly. This may be advantageous in case of repeated runs (most practical situations), where the user may be able to enforce better load balancing based on the knowledge of the modal distribution.

The essence of this method is that all processors solve the same global problem in different frequency intervals (the segments). This requires the collection of the global mode shapes of these intervals on a master processor to facilitate follow-on modal analysis techniques or appropriate post-processing.

1.2 Geometric domain decomposition in Version 2001

The geometric domain decomposition based Lanczos is applicable to very large problemsizes. More importantly, it is now not confined to the READ module as the geometry domain decomposition is executed in the SEQP (once structural sequencer) module prior to element generation and assembly. This fact results in additional performance improvements outside the

READ module (namely in EMG, EMA and others upstream of READ). This technology may also be used in connection with the frequency domain decomposition in a hierarchic fashion in the future.

It is important to emphasize that this version still solves the **global eigenvalue problem** posed by the user and solves it **exactly**. This is **not** a **component modal reduction** method. The only difference from the serial Lanczos method that in this case the global matrices are partitioned and each processor sees only its local portion.

After receiving only the submatrices corresponding to the automatically created subdomains, the Lanczos process will be executed simultaneously on each of the subdomains located on the different processors. The boundary segments of course require interprocessor communication, a known bottleneck of the method. On the other hand the communication on the boundary is minimized via our very advanced proprietary method and its efficient distributed implementation.

It is important to point out that the geometry domain parallel method has the additional advantage of significantly reduced resource (memory and disk) requirements since the problem to be solved is smaller and smaller as the number of domains is increasing. Of course the size of the boundary is increasing with the number of subdivisions, reducing the efficiency gain. Nevertheless, the geometry domain method is very competitive in cases when the frequency domain parallel method (each processor working on the same full geometry) cannot complete the analysis due the lack of resources.

Finally, it is important to note that since this version solves the global eigenvalue problem posed by the user in subdomains, the resulting eigenvector table (PHIA output of READ) contains only the (geometrically) local segments of all the eigenvectors. This result is properly interpreted by the DISOFP (distributed Output File Processor) module enabling complete data recovery. Naturally all the eigenvalues are available on all processors in the LAMA table and LAMAM matrix; both are outputs of the READ module.

1.3 User interface

The user interface in V2001 for both distributed methods needs on the submittal line: dmp= n, where n is the number of CPUs.

For the frequency method, in addition the user needs to set the parameter NUMSEG on the NASTRAN card, or on the EIGRL card. Currently the number of segments (NUMSEG) must agree with the number of CPUs used (dmp). The user also has the choice of introducing intermediate frequencies directly on the EIGRL continuation card. The intermediate frequency locations and other segment specification aspects of the EIGRL card are described in Ref. 3.

For the new geometry method, no other information is needed from the user. The number of geometry domains is automatically set to dmp (given on the submittal line as mentioned above). This limitation may be released later, enabling the use of a hierarchical approach, where both frequency and geometry domain paradigms are used.

2. Performance analysis

2.1 Benchmark problem

The MSC.Nastran model used in this exercise was an automobile (body in white) model consisting of 232,649 grids and 232,404 (mainly quadrilateral) elements resulting in 1,377,677 global degrees of freedom. SOL 103 was used up to 200 Hz with both frequency and geometry domain parallel ways of execution. It is interesting to observe the result of the automatic subdivision process, obtained by our proprietary multilevel vertex subdivision method. The most relevant information is summarized in Table 1.

Number of	Interior Grids		Boundary Grids		Degrees of Freedom	
Subdivisions	Largest	Smallest	Largest	Smallest	Interior	Boundary
1	232,649	-	-	-	1,377,677	-
2	124,420	107,721	508	508	640,081	5,030
4	64,708	53,582	619	455	315,167	6,234
8	35,868	24,153	697	253	163,943	9,972
16	16,856	11,266	522	85	71,142	13,410

Table 1. Details of domain decomposition

On the previous to last column one can easily see that interior and therefore the local problemsize is significantly decreasing. This will contribute mainly to the performance improvement and the memory requirement reduction. The last column demonstrates the increase of the boundary size, which unfortunately is the bottleneck of the method and leads to increased shift and communication costs.

2.2 Computational environment

The performance runs were executed on an IBM SP machine which is a cluster of 8 workstation nodes containing 4 POWER3-II (375 MHz) processors. The nodes are connected with IBM's SP switch, which enables point to point communication between the nodes and has a bandwidth of 150 Mbytes per second. Each node is similar to the IBM 44P Model 270 workstation, however, with 8 Mbyte caches as opposed to the 4 Mbytes of the 44P. Each node has 8 GBytes of memory and 16 disks of 9 GBytes each, however, only two I/O paths. Therefore the elapsed time may suffer due to I/O conflicts. In light of this and since the nodes have a shared memory (SMP) architecture, the best overall performance may be obtained by using only 1 or 2 processors per node. This is the way the following performance results were obtained.

2.3 Performance results

The results shown in Table 2 are related to the READ module only. In the case of the new geometry domain decomposition method additional speedup is also obtained outside of the READ module, as mentioned earlier. The incorporation of this by using the total solution times would render the comparison to the frequency technique unfair. This could be especially true

when a large amount of data recovery is executed. The frequency domain execution times are from Version 70.7 and the geometry domain times are from Version 2001.

	Number of processors				
	1	2	4	8	16
Frequency domain					
CPU time sec	15,096.	9,372.	7,231.	5,691.	4,195.
I/O GB	1,043.	631.	465.	312.	245.
Elapsed min:sec	293:36	187:21	132:40	98:05	84:57
Speedup	-	1.56	2.22	2.99	3.47
Geometry domain					
CPU time sec	15,073.	10,581.	6,209	5,542.	4,427.
I/O GB	1,004.	576.	304.	206.	177.
Elapsed min:sec	309:21	186:39	109:19	95:27	74:10
Speedup	-	1.66	2.83	3.25	4.17
Memory Used MW	39.9	18.3	9.1	4.4	2.5

Table 2. READ module performance

These results show the better scalability and overall advantage of the newly introduced geometry domain method over the V70.7 introduced frequency method. This demonstrates our continuing commitment to improving the performance from version to version. The memory usage results in the geometry domain method demonstrate the additional advantage of the method in requiring smaller amount of memory as the number of subdivision increases This is contrary to the frequency method which required 40 Million words to run in every version.

For those readers interested in the detailed cost analysis of the Lanczos method, Table 3 presents a summary of the F04 file's User Information Message 5403 contents of the various runs (Reps stands for repetitions):

Number of	FBS		MPY		SHIFT		RUN	
CPUs	CPUs	Reps	CPUs	Reps	CPUs	Reps	CPUs	Reps
1	6,242.	177	145.	408	1,950.	7	6,327	5
2	4,151.	234	144.	538	1,676.	7	4,445.	5
4	2,307.	206	84.	473	1,380.	6	2,372.	4
8	1,653.	218	127.	496	1,906.	7	1,830.	5
16	1,392.	228	122.	527	1,259.	6	995.	4
Speedup	4.48		1.19		1.55		6.36	

Table 3. Cost elements of Lanczos method

The FBS operation and the Lanczos run scale reasonably well, however, the MPY and the SHIFT do not scale well. Since the matrix multiply contains lots of communications that is understandable. In fact the MPY cost may be even more in case of using coupled mass

approach, our example used the lumped mass approach in which case the mass matrix is diagonal.

The shifting cost (partial decomposition of the interior and complete decomposition of the boundary) has not improved at all above 4 CPUs. This is due to the increased cost related to the boundary and as mentioned above is one of the necessary bottlenecks of the approach. In specific, the decomposition time of the interiors of the 16-domain run was in the order of 10 seconds, while the boundary decomposition is in the order of hundred seconds. Latter fact is caused by the higher density of the boundary matrix and the communication required during its creation.

CONCLUSIONS

The final 16 CPU execution time with the new method (74 minutes 10 seconds) is easily fulfilling the auto industry's usual daily design cycle requirement. We believe that this is the state of the art solution in the area of large NVH analyses, however, we continue our quest for improving the performance of large eigenvalue analysis jobs.

In specific, a hierarchic execution of both of the above methods in the same run is subject of our current research. That would enable the feasible application of larger number of processors (in the range of 16-64) to be applied to a single analysis job in the future.

REFERENCES

1. Viersbach, U. et al.; "Distributed memory parallel MSC.Nastran on an IBM workstation cluster at Ford Cologne," Proc. of 1st MSC Worldwide Automotive Users' Conference, Munich, Germany, 1999

2. Komzsik, L. et al.; "A breakthrough in parallel performance in MSC.Nastran V70.7," Proc. of 1st MSC Worldwide Automotive Users' Conference, Munich, Germany, 1999

3. Komzsik, L.; *MSC.Nastran Numerical Methods Users Guide, 7th edition*, to be published, MSC.Software, 2000

ACKNOWLEDGEMENTS

Thanks are due to the IBM Corporation for providing the hardware used in the development and testing of the new method. In specific, Mr. Doug Petesch's help in executing the performance analysis and providing the benchmark results as well as his recommendations in the tuning of the system are also well appreciated. Finally, special acknowledgement is due to my co-workers Tom Kowalski and Ken Burrell of MSC for their contributions in the development of the geometry domain distributed Lanczos and the multilevel vertex domain decomposition technique, respectively.