A Breakthrough in Parallel Performance in MSC/NASTRAN V70.7

Louis Komzsik
Chief Numerical Analyst
The MacNeal-Schwendler Corporation
815 Colorado Blvd, Los Angeles, CA 90041-1777
E-mail: louis.komzsik@macsch.com

Stefan Mayer
Numerical Analyst
MacNeal-Schwendler GmbH
Innsbrucker Ring 15
D-81612 München
E-mail: stefan.mayer@macsch.com

Petra Poschmann
Numerical Analyst
The MacNeal-Schwendler Corporation
815 Colorado Blvd, Los Angeles, CA 90041-1777
E-mail: petra.poschmann@macsch.com

ABSTRACT

More than a decade ago MSC offered the first parallel production system of MSC/NASTRAN. During this decade MSC has intensified its effort on parallel MSC/NASTRAN and is now ready to deliver MSC/NASTRAN V70.7, which contains very important new parallel features. This paper describes these exciting features and provides preliminary performance results for V70.7. We believe that this system marks the best in parallel MSC/NASTRAN performance ever and presents a breakthrough in parallel computing in our market.

1. Introduction

More than a decade ago MSC offered the first parallel production system of MSC/NASTRAN, based on the shared memory paradigm. At that time and during the following years the parallelization efforts concentrated on parallelizing the computations in several expensive modules, for example the matrix decomposition. However, during recent years it has become clear that a more extensive parallelization of MSC/NASTRAN is necessary to satisfy the users' ever growing demand for higher performance. Moreover, it has turned out that not only computations, but also the I/O traffic must be parallelized in order to obtain a highly efficient parallel MSC/NASTRAN.

MSC started to work on new parallelization approaches earlier this decade, this time based on the distributed memory paradigm to be able to address parallel I/O issues as well. First successes were obtained in the European Europort project, which resulted in the distributed parallel production version V69.2 available on the IBM SP architecture only. Encouraged by the results of this project, the efforts on distributed parallel MSC/NASTRAN have been intensified during the past two years. This resulted in versions 70.5.3 and 70.5.4 available on the IBM AIX Parallel Environment.

MSC is now ready to deliver MSC/NASTRAN V70.7, which contains even more exciting new parallel features. This version, in addition to IBM, will be delivered on the HP V class and possibly on SUN and SGI computers. It will also be possible to execute this version on a homogeneous cluster of workstations. The results obtained with this system mark the best in parallel MSC/NASTRAN performance ever and present a breakthrough in distributed parallel computing in our market.

2. Distributed memory computers

Distributed memory computers are essentially tightly coupled networks of workstations. The connection between the processors (nodes) of a distributed memory computer is usually a very fast proprietary network or switch. The communication between these nodes is via standard interface libraries, for example MPI (Message Passing Interface), which is used by MSC in the distributed memory parallel implementation. Each node of a distributed memory computer has its own memory, hence the name. Even more important is the fact that the nodes have their own local I/O (disk) device enabling efficient parallel I/O operations.

A leading example of these kind of computers is located at IBM Poughkeepsie and has the following technical specifications:

- IBM RS/6000 SP with 397 type nodes and SSA disk
- 8 server nodes each with a Power2 160 MHz CPU
- 1 file server node with a Power2 160 MHz CPU
- 1 GB of local memory on each node
- 40 GB of local disk on each node
- RS/6000 SP Switch for data communications

MSC's in-house IBM RS/6000 SP has 8 39H type nodes with 66 MHz Power2 CPUs, 512 MB local memory and 8 GB wide SCSI-2 disks.

These two machines were used to obtain the results presented in Section 4. It is important to point out that special software techniques developed for these computers may be also executed on a regular network of workstations, naturally with less efficiency. Even shared memory parallel (smp) computers can emulate the MPI commmunication principle enabling the execution of the distributed memory techniques on these computers also. All of these reasons increase the value of distributed memory parallel (dmp) development work significantly.

3. Distributed solution techniques

The distributed solution techniques implemented in Version 70.7 are based on the principle of domain decomposition. We found two possible and convenient ways to decompose finite element problems. They are the frequency and the geometric domain decomposition. In the following the basic principles of both of these techniques are discussed. It is important to note, that the following distributed solution techniques provide the same results and output formats as the serial solutions, apart from some additional performance summaries related to the distributed processes. We also made the additional user interface needs very minimal as shown below.

3.1 Frequency domain decomposition based distributed solutions

The frequency domain decomposition technique applies only to certain solution sequences where the notion of frequency is utilized. Such are the normal modes and frequency response analyses. More specifically, the frequency domain decomposition based normal modes analysis is built on the frequency segment approach of the Lanczos method. The frequency domain decomposition based frequency response analysis utilizes the independence of the discrete frequencies given on the user's list.

3.1.1 Distributed normal modes analysis

The segmented version of Lanczos was first introduced in the early 1990s to alleviate the problems encountered on very wide frequency range runs mostly on Cray computers. In auto-industry tests it was found that the orthogonality is lost when very long Lanczos runs are executed while trying to span a wide frequency range of interest. We introduced the segment version to force intermittent shifts at semi-regular intervals resulting in more frequent restarts of the Lanczos process. While it has significantly improved the solution quality, i.e. we avoided aborts due to loss of orthogonality, the number of shifts was usually more than in the non-segmented run.

It is a natural extension to this logic to execute the segments in parallel, which is the basis of the distributed parallel implementation in Version 70.7. This process is executed in a master/slave paradigm. One (the master) of the processors will execute the coordination and collection of the separately calculated eigenvectors into an appropriately ordered set. This guarantees the continuation and proper exit of the master process. The slave processes will contain only the eigenvalues and eigenvectors they found upon exiting the READ (Real Eigenvaue Analysis DMAP) module.

The user interface for this parallel method is as follows. On the submittal line the user must add "dmp=n", where n is the number of CPUs. The dmp keyword may also be written as "dmparallel" and it stands for distributed memory parallel. In addition to that the user should include a NASTRAN entry containing “numseg=m", where m is the number of segments. The number of segments may be less than the number of CPUs and in lieu of setting numseg, the default is m=n from the dmp command. The user also has the choice of introducing intermediate frequencies directly on the EIGRL continuation card. This may be especially advantageous in case of repeated runs (the most practical situation), where the user may be able to enforce better load balancing than the automatic intermediate frequency generation method does.

3.1.2 Distributed frequency response analysis

The distributed frequency response analysis is based on assigning exclusive subsets of the user given frequency list to each processor. Each processor calculates the responses of the structure at the frequencies given in its subset. The calculation may be executed in a master/slave mode which was the only method available in Version 69.2 or now in a more efficient symmetric ("all master") operation mode automatically selected by the code. In the master/slave mode the master processor is distributing the structural matrices and the frequency list to the slave processors in the beginning of the frequency response module and upon completion of the response calculation collects the results. This operation mode is used in the distributed modal frequency response analysis solution.

In the symmetric mode all processors behave identically. They calculate their own structural matrices and respective frequency subsets and the responses. They each also complete the solution sequence, resulting in higher scalability due to the locally kept output results. This symmetric operation mode is used in the distributed direct frequency response calculations.

The user interface for this parallel method is as follows. On the submittal line the user must add "dmp=n", just as in the case of distributed normal modes analysis. There is no need to add anything to the NASTRAN card. The number of subsets will be n. The number of frequencies in each subset will be the user given number of frequencies divided by n, with some adjustment if n does not divide evenly. The operational modes may be selected by the “slave=yes” or “slave=no” keyword settings on the submittal line. The user may also put the local results into local databases for future collection or separate viewing by the postprocessor program, such as MSC/PATRAN. This is done by setting "mergeresults=no" on the submittal line resulting in skipping the collection of the local results. The default "mergeresults=yes" will create one XDB file containing all the results.

3.2 Geometric domain decomposition based distributed solutions

The principle of geometric domain decomposition is applicable on a much higher level than the frequency domain decomposition. In the frequency case the distributed solution is focused on a certain module (READ and FRRD1) and solution sequence (Sol 103 and 108, as well as 111). The geometric domain decomposition principle transcends many DMAP modules and also solution sequences.

In this first production version of this technology we focused on the linear static analysis (Sol 101), however, many aspects of the development work carry over to other solution sequences to be delivered in distributed form in the future. This technology may also be used in connection with the frequency domain decomposition in a hierarchic fashion. Finally, this technology in part relies on the very strong superelement technology of MSC/NASTRAN which gives its foundation.

The cornerstone of the geometric domain decomposition is an automatic domain decomposition tool. This tool works from geometry (connectivity) information and uses heuristic algorithms to create subdomains. The main criteria in creating subdomains are: minimizing the boundary between the domains, achieving load balance and minimizing the cost of the solution of the interior of the domains. In Version 70.7 we use the EXTREME tool for domain decomposition, however, other tools such as METIS may also be used in the near future.

After automatically creating the subdomains, MSC/NASTRAN's superelement process will be executed. The shortcoming of the boundary solution of the conventional superelement process, the explicit creation of the boundary Schur compliment matrix, however, is avoided. This requires very advanced and efficient distributed algorithms designed and implemented by MSC. These proprietary algorithms encapsulate most of the interprocessor communication via MPI utilities.

The user interface for this parallel method is also "dmp=n" on the submittal line just as in the case of distributed normal modes or the case of frequency response. This will result in creating n subdomains and assigning one to each processor. Each processor executes the steps of linear static analysis from element matrix generation/assembly through constraint elimination until solution and data recovery for their own subdomain. Of course the correct solution of the boundary of the local subdomains requires aformentioned communication.

The data recovery for each domain is done independently on each processor and does not require any communication. This results in linear or better speed-ups and disk-space savings during that part of the run. Following data recovery, the output will be automatically merged so that all results will be displayed in the F06 file of the master processor. Each processor will have a local XDB file for its domain, assuming the user sets param,post,0. By default the user will have the local XDB files merged into one. By setting mergeresults=no the local XDB files will be retained and the merging is skipped. MSC/PATRAN can be used to postprocess these distributed XDB files and visualize the complete model.

4. Preliminary Version 70.7 results

To demonstrate the performance improvement obtained by the distributed Lanczos technique, we choose a carbody model of approximately 200,000 degrees of freedom consisting of mainly shell (QUAD4) elements (Figure 1).

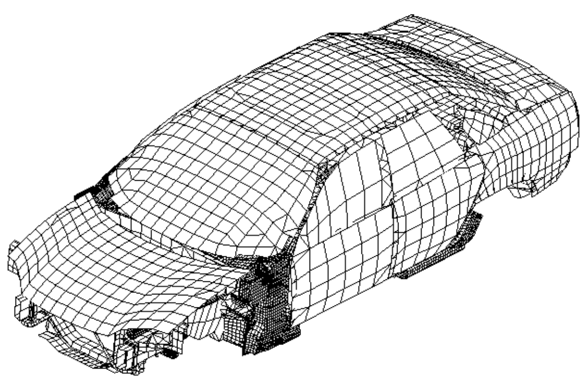

Figure 1. Carbody model

The normal modes analysis was executed up to 400 Hz, extracting 1876 eigenvectors, using 100Mw of memory. The disk usage (the main processor's scratch database high water mark) and the elapsed time of the total solution are shown in Table 1.

Number of CPUs	Disk Gbytes	Elapsed min:sec	Speedup
1	10.3	449:27	--
2	7.7	329:23	1.4
4	7.0	122:39	3.7
8	6.7	87:57	5.1

Table 1. Distributed normal modes performance results

The scalability and the speedups are quite remarkable. We obtained a 5 fold elapsed time speedup on 8 processors. It is even more important to point out the significantly lower disk requirements on the individual processors in the case of the parallel runs due to the smaller scratch space needs. The results were obtained with the machine at IBM Poughkeepsie described in Section 2.

The distributed parallel frequency response capability is demonstrated by the analysis of an exhaust manifold model (Figure 2).

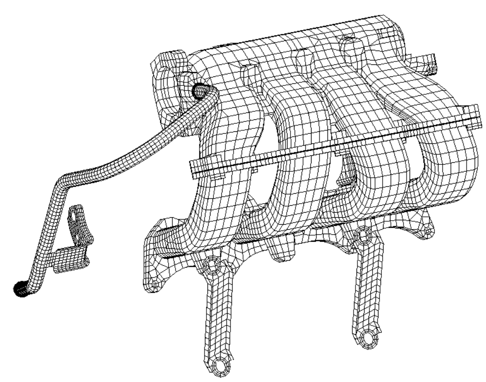

Figure 2. Exhaust manifold model

The model consists of app. 50,000 degrees of freedom and was built of mixed (shell and solid) elements. Direct frequency response analysis (Sol 108) with 100 frequency steps resulted in the performance measurements of Table 2.

Number of CPUs	Elapsed Seconds	Speedup
1	9521	--
2	5001	1.9
4	2672	3.6
8	1546	6.2

Table 2. Distributed frequency response performance results

This analysis represents an even better scalability. This is due to the fact that the complete frequency response solution was executed in parallel and none of the processors collected the response results of others, the results were kept local and the database separate (mergeresults=no). The results were obtained with MSC's in-house IBM RS/6000 SP, which is also described in Section 2.

The effect of the geometric domain decomposition based distributed execution of the linear statics analysis is shown with the following example. The model is an automobile crankshaft (Figure 3) with about 150,000 degrees of freedom and built exclusively from solid elements.

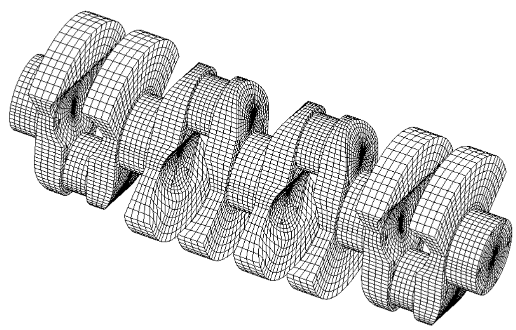

Figure 3. Crankshaft model

Table 3 compares the elapsed time (min:sec) and disk space requirements of Solution 101 with one and 8 processors, again on MSC's in-house IBM SP (see Section 2).

	1 Processor	8 Processor	Speedups/Saving
Disk	1.4GB	0.26GB	5.4
Elapsed	38:42	8:35	4.6

Table 3. Distributed linear statics performance results

The speedup of 4.6 on 8 CPUs is very good and the 5.4 fold saving in maximum local disk space is significant.

It is important to note that all 1 CPU runs shown in this section were executed with the fully tuned serial production implementation of MSC/NASTRAN and not from single processor runs of the parallel algorithms.

The relative merits of the 3 distributed parallel solutions will be shown by executing linear statics, normal modes and frequency response runs on the same model in the conference presentation.

5. Conclusions

We hope to have demonstrated the significant performance advantages of the new distributed solution techniques. These techniques present groundbreaking solutions in our segment of the CAE market. We believe that these capabilities will be very useful for a wide range of our user community.

We are continuing the distributed parallel development and are committed to further improvements. We are planning to expand the geometric domain decomposition based technology to other solution sequences, for example normal modes and transient analysis in the future.