

# Model Reduction for High Dimensional Micro-FE Models

---

**Rudnyi E. B., Korvink J. G.**

University of Freiburg, IMTEK-Department of Microsystems Engineering,  
{rudnyi,korvink}@imtek.uni-freiburg.de

**van Rietbergen B.**

Eindhoven University of Technology, Biomedical Engineering, Bone &  
Orthopaedic Biomechanics, B.v.Rietbergen@tue.nl

---

## Abstract

*Three-dimensional serial reconstruction techniques allow us to construct very detailed micro-finite element (micro-FE) model of bones that can represent the porous bone micro-architecture. Solving these models, however, implies solving very large sets of equations (order of millions degrees of freedom), which inhibits dynamic analyses. Previously it was shown for bone models up to 1 million degrees of freedom that formal model order reduction allows us to perform harmonic response simulation for such problems at a computational cost comparable with that of a static solution. However, the main requirement then is that a direct solver is used and there is enough memory to keep the factor.*

*With common serial computers with 4GB of RAM such analyses are limited to bone models with about 1 million degrees of freedom. In order to extend this work to larger models, a parallelizable sparse direct solver needs to be implemented on larger parallel systems, which was the goal of this work.*

*Four bone models have been generated in the range from 1 to 12 million degrees of freedom. They have been used to benchmark the multifrontal massively parallel sparse direct solver MUMPS on SGI Origin 3800 (teras at SARA). The study showed that MUMPS is a good choice to implement model reduction of high dimensional micro-finite element bone models provided that there is enough memory on a high performance computing system.*

## Introduction

Three-dimensional serial reconstruction techniques allow constructing very detailed micro-finite element (micro-FE) model of bones that can represent the porous bone micro-architecture [1]. Such models can be used, for example to study differences in bone tissue loading between healthy and osteoporotic human bones during quasi static loading [2]. The main disadvantage of this approach is its huge computational requirements because of the high dimensionality of the models. This led to the fact that such computational analyses are limited to a static solution. There is evidence, however, that in particular high-frequency oscillating mechanical loads (up to approximately 50 Hz) might play an important role in the bone adaptive response [3]. Hence, there is a need to extent these static analyses to dynamic ones.

Model order reduction is a relatively new area of mathematics that aims at finding a low-dimensional approximation for a high-dimensional system of ordinary differential equations. It has been used successfully over the last few years in different engineering disciplines like electrical engineering, structural mechanics, heat transfer, acoustics and electromagnetics [4].

Implicit moment matching based on the Arnoldi algorithm [5] is a very competitive approach to find the low dimensional subspace that accurately approximates the behavior of a high-dimensional state vector. At IMTEK, software MOR for ANSYS [6] has been used developed in order to perform model reduction directly to finite element models made in ANSYS. Its computational kernel has been employed for the two bone models of dimensions 130 000 and 900 000 accordingly and it has been demonstrated that model reduction can speed up harmonic simulation by orders of magnitude on a computer with 4 Gb RAM [7].

The bottleneck of model reduction based on the Arnoldi algorithm is a solution of a system of linear equations that should take place for each Arnoldi vector [6]. As a result, the use of a direct solver is preferred. However, 4 Gb of RAM happens to be enough to keep a factor of a bone model up to 900 000 only [7]. The treatment of models with higher dimensions requires parallel computing.

The goal of the current work is to investigate what dimension of a bone model one can afford with the multifrontal massively parallel sparse direct solver MUMPS [8][9]. We first introduce the bone models employed as benchmarks. After that, we present the results obtained on SGI Origin 3800 (teras at SARA, <http://www.sara.nl/userinfo/teras/description/index.html>) and make conclusions.

## Benchmarks of Micro-FE Bone Models

The bone models are presented in Table 1. The matrices are symmetric and positive definite.

The first two models, BS01 and BS10, are from [7]. They have been used for debugging the code. The new bone models have been obtained from another bone sample that was scanned with higher resolution. Models from B010 to B120 represent increasing subsets from the same bone sample.

	BS01	BS10	B010	B025	B050	B120
number of elements	20098	192539	278259	606253	1378782	3387547
number of nodes	42508	305066	329001	719987	1644848	3989996
number of DoFs	127224	914898	986703	2159661	4934244	11969688
nnz in half K	3421188	28191660	36326514	79292769	180663963	441785526

*Table 1 Bone micro-finite element models*

The matrices as well as the driver to call MUMPS are available at Oberwolfach Model Reduction Benchmark Collection [10] (<http://www.imtek.uni-freiburg.de/simulation/benchmark/>).

## Performance of MUMPS on teras

Numerical experiments have been performed on SGI Origin 3800 (teras at SARA). MUMPS version 4.6.1 has been compiled with the MIPSpro compilers suite version 7.3.1.3m with full optimization (-O3) as a 64-bit application. The host processor has been used exclusively for matrix assembly (PAR=0) and did not take part during factorization. As a result, we use for plotting the number of processors reduced by one.

One processor on the SGI Origin 3800 is recommended to use about 1 Gb of memory and hence the minimal number of processors depends on the problem size. Memory requirement depends on the number of nonzeros in the factor. Fig 1 displays the ratio of nonzeros in the factor to nonzeros in the matrix as a function of the matrix dimension after METIS [11] and PORD [12] reordering. It shows that the first two matrices treated in [7] are much easier to deal with as the number of nonzeros in the factor is about only 10 times more than in the matrix. The connectivity in the new four matrices is much higher, the number of nonzeros

being about 40 times more than in the matrix. Note that the number of nonzeros in the matrix is proportional to the matrix dimension (see Table 1).

METIS reordering [11] gives the number of nonzeros in the factor about 14% less than PORD reordering [12] but unfortunately METIS is not yet fully 64-bit compliant [13] and could not be employed for matrices with higher dimensions.

Fig 2 to 4 show the dependence of the wall time, the total memory (INFOG(19)) and memory on the most memory consuming processor (INFOG(18)) for matrix B010 as a function of the number of processors (not counting the host processor). The behavior is similar for other matrices.

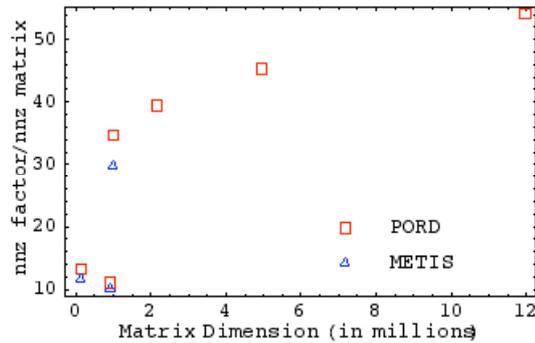


Fig 1 The ratio of non-zero elements in the factor to non-zero elements in the matrix.

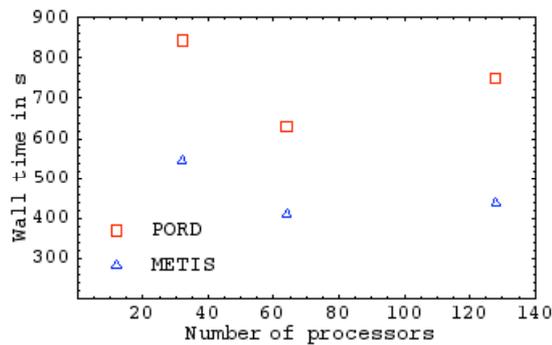


Fig 2 The wall time in s to factorize matrix B010 as a function of the number of processors.

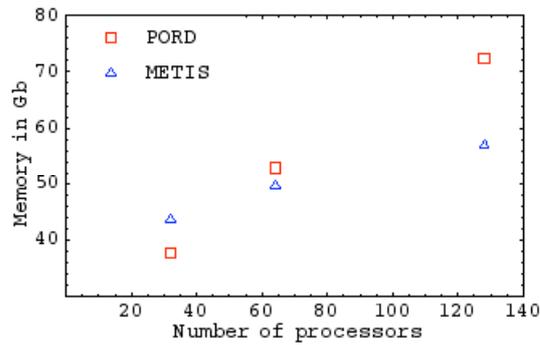


Fig 3 Total memory in Gb (INFOG(19)) to factor matrix B010 as a function of the number of processors.

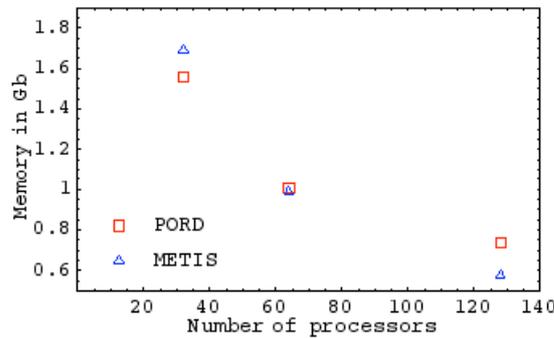


Fig 4 The memory in Gb on the most memory consuming processor (INFOG(18)) to factor matrix B010 as a function of the number of processors.

Our observations can be summarized as follows:

- The total memory required MUMPS to factorize a matrix is not constant and gradually grows when the number of processors is increased.
- There is an optimal number of processors to employ that depends on the matrix dimension.
- METIS reordering allows MUMPS for more efficient distributed computing than PORD.
- Provided there is enough memory to keep the factor, MUMPS allows us for efficient implementation of model reduction: to factorize a matrix once and then use back substitution during the Arnoldi process.

Fig 5 shows memory required by MUMPS per number ((INFOG(19) divided by the sum of nonzeros in the factor and the matrix) as a function of the number of processors for the four new matrices in the case of PORD reordering.

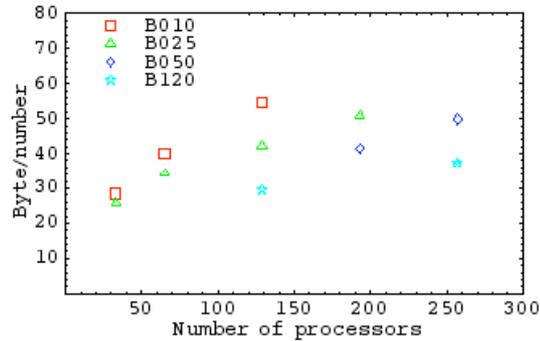


Fig 5 Memory required MUMPS per number as a function of the number of processors.

The matrix B120 requires about 1 Tb of RAM and if we want to treat bone models similar to the new benchmarks up to dimensions of 100 millions we may need up to 10 Tb of RAM.

## Conclusion

The study showed that MUMPS is a good choice to implement model reduction of high dimensional micro-finite element bone models provided that there is enough memory on a high performance computing system. In the future, the memory requirements can be reduced when METIS becomes 64-bit compliant.

Trilinos [14] interfaces MUMPS and, in our view, this is the best option to integrate MUMPS [8][9] with MOR for ANSYS [6].

## Acknowledgment

The work has been performed under the Project HPC-EUROPA (RII3-CT-2003-506079), with the support of the European Community - Research Infrastructure Action under the FP6 “Structuring the European Research Area” Programme.

---

## References

- [1] B. van Rietbergen, H. Weinans, R. Huiskes, A. Odgaard, A new method to determine trabecular bone elastic properties and loading using micromechanical finite-elements models. *J. Biomechanics*, v. 28, N 1, p. 69-81, 1995.
- [2] B. van Rietbergen, R. Huiskes, F. Eckstein, P. Rueeggsegger, Trabecular Bone Tissue Strains in the Healthy and Osteoporotic Human Femur, *J. Bone Mineral Research*, v. 18, N 10, p. 1781 - 1787, 2003.
- [3] L. E. Lanyon, C. T. Rubin, Static versus dynamic loads as an influence on bone remodelling, *Journal of Biomechanics*, 17: 897-906, 1984.
- [4] E. B. Rudnyi, J. G. Korvink, Review: Automatic Model Reduction for Transient Simulation of MEMS-based Devices. *Sensors Update* v. 11, p. 3-33, 2002.
- [5] R. W. Freund, Krylov-subspace methods for reduced-order modeling in circuit simulation, *Journal of Computational and Applied Mathematics*, v. 123, pp. 395-421, 2000.
- [6] E. B. Rudnyi, J. G. Korvink. Model Order Reduction for Large Scale Engineering Models Developed in ANSYS. *Lecture Notes in Computer Science*, v. 3732, pp. 349-356, 2005. <http://www.imtek.uni-freiburg.de/simulation/mor4ansys/>
- [7] E. B. Rudnyi, B. van Rietbergen, J. G. Korvink. Efficient Harmonic Simulation of a Trabecular Bone Finite Element Model by means of Model Reduction. 12th Workshop "The Finite Element Method in Biomedical Engineering, Biomechanics and Related Fields", University of Ulm, 20-21 July 2005. Preprint at <http://www.imtek.uni-freiburg.de/simulation/mor4ansys/pdf/rudnyi05ulm.pdf>.
- [8] P. R. Amestoy, I. S. Duff, J. Koster and J.-Y. L'Excellent, A fully asynchronous multifrontal solver using distributed dynamic scheduling, *SIAM Journal of Matrix Analysis and Applications*, v. 23, No 1, pp 15-41 (2001).
- [9] P. R. Amestoy and A. Guermouche and J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, v. 32 (2), pp 136-156 (2006). <http://graal.ens-lyon.fr/MUMPS/>
- [10] J. G. Korvink, E. B. Rudnyi. Oberwolfach Benchmark Collection. In: Benner, P., Mehrmann, V., Sorensen, D. (eds) *Dimension Reduction of Large-Scale Systems*, *Lecture Notes in Computational Science and Engineering (LNCSE)*. Springer-Verlag, Berlin/Heidelberg, Germany, v. 45, p. 311-315, 2005. <http://www.imtek.uni-freiburg.de/simulation/benchmark/>
- [11] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM Journal on Scientific Computing*, Vol. 20, N 1, 1998, pages 359-392. <http://glaros.dtc.umn.edu/gkhome/views/metis/metis/>
- [12] J. Schulze, Towards a tighter coupling of bottom-up and top-down sparse matrix ordering methods, *BIT*, Vol. 41, No. 4, pp. 800-841, 2001.
- [13] G. Karypis, Private communication, 2006
- [14] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, K. S. Stanley, An overview of the Trilinos project, *ACM Trans. Math. Softw.*, v. 31, N 3, p. 397-423, 2005. <http://software.sandia.gov/trilinos/>