Model Order Reduction of Large Scale ODE Systems: MOR for ANSYS versus ROM Workbench

Tamara Bechtold[(1)], Arie Verhoeven[(1),(2)], Jan ter Maten[(1)]
(1) Philips Research Laboratories, (2)Technical University of Eindhoven
E-mail:tamara.bechtold@philips.com

Abstract - In this paper we compare the numerical results of different model order reduction software tools, in order to test their scalability for the microelectronic-industry relevant problems. MOR for ANSYS is implemented in C++ and ROM Workbench is a MATLAB code. The chosen benchmarks are large scale linear ordinary differential equation systems, which arise from the finite element discretisation of electro-thermal MEMS models.

I. Introduction

The decreasing size of silicon chips and increasing integration density require permanently new and more powerful simulation tools and strategies in microelectronics and microsysytem technology. Model order reduction approach (MOR) [1] is successfully used to considerably reduce the computational time and resources. Mathematical development of MOR is an active area of research, which grows from the reduction of linear ordinary differential equation systems (ODEs) towards the reduction of parametrized and nonlinear differential algebraic equations (DAEs) and partial differential algebraic equations (PDAEs). The implementation aspects of model order reduction are advancing as well. Practical MOR has developed from academic prototyping environments to several strong tools that can be easily used as an extension of the commercial simulators like e. g. ANSYS.

In the today's age of fast computers it is possible to use quick prototyping tools like MATLAB or Mathematica for convenient implementation and testing of new MOR methods. However, the run time for the usually large-scale industry relevant problems enforces the use of programming languages, like C++. Such implementation offers better performances, but demands more time and programming skills from the developer.

The goal of this paper is to numerically compare two MOR tools, which belong to the described streams: MOR for ANSYS [2], which is a C++ code and ROM Workbench (RW) [3], which is written in MATLAB. Both tools are planned for use in the European project COMSON [4], which joins the efforts of the major European semiconductor companies and academic nodes to develop a software tool that could fulfill the demands of the modern microelectronic industry. Such comparison will give us a clear understanding up to which size and for what structure of the industrial problem, the MATLAB can be used and at which point one should switch to the compiled language implementation.

II. Benchmarks

We have chosen several electro-thermal MEMS models [5], which are described by the linear multiple-input multiple-output ODE systems of the form:

$$C \cdot \dot{x} + G \cdot x = B \cdot u(t)$$
$$y = L \cdot x \qquad (1)$$

whith initial condition $x(0) = x_0$. Here t is the time variable, $x(t) \in R^n$ is a state vector, $u(t) \in R^m$ the input excitation vector and $y(t) \in R^p$ the output measurement vector. $G, C \in R^{nxn}$ are linear (independant on the state vector) symmetric and sparse system matrices, $B \in R^{nxm}$ and $L \in R^{nxp}$ are input and output distribution arrays, respectively. n is the dimension of the system and m and p are the number of inputs and outputs. In Table I the dimension of each test model and the matrix structure of the matrix G are shown.

III. Methods

The goal of model order reduction is to produce the system of the same form as (1), but with much smaller dimension. MOR for ANSYS implements the Arnoldi algorithm [6] and ROM Workbench the Arnoldi-based PRIMA [7]. As MOR for ANSYS projects both system matrices, those two algorithms have the same computational requirements.

IV. Results

We have reduced the described case studies with both tools. In Fig. 1 the step responses of the full-scale and of the reduced order model in a single output node of the pyrotechnical microthruster (low order elements) are displayed. The difference between the reduced model computed with MOR for ANSYS and RW is of the order of rounding errors.

In the Table II we compare the reduction time (down to order 30) of MOR for ANSYS and RW for the described MEMS case studies. CPU time of RW is 10-20 times longer than the CPU of MOR for ANSYS and for the final test RW runs out of memory.

Note that in both implementations the system matrix G is factorized. In MOR for ANSYS this is done via TAUCS library which implements Cholesky decomposition with METIS reordering scheme. RW uses MATLAB function `lu` for the LU decomposition and `colamd` reordering. As Cholesky decomposition is at least two times faster than LU and METIS reordering is more effective than colamd, this may explain the big difference in the CPU time of both software tools.

V. Conclusion

We have compared two software tools, which are meant to be integrated into COMSON demonstrator platform. They belong to the two main implementation schemes, fast prototyping in the interpreter environment and the compiled language implementation in C++. Unfortunately, at present we can not say how much the CPU time difference is due to the interpretation overhead in MATLAB and how much it is due to different reordering and factorization schemes. But it is certain that the performance of RW can be improved by changing the reordering and factoring tools (TAUCS has a MATLAB interface).

VI. Acknowledgments

We would like to thank Dr. Evgenii B. Rudnyi for helping us with MOR for ANSYS, Prof. Dr. Gabriela Ciuprina for her help with ROM Workbench and to acknowledge the EU support through the COMSON project.

VII. References

[1] A. C. Antoulas, "Approximation of Large-Scale Dynamical Systems", Society for Industrial and Applied Mathematics, 2005, ISBN: 0898715296.

[2] www.imtek.de/simulation/mor4ansys.

[3] www.imek.be/codestar.

[4] www.comson.org.

[5] Oberwolfach Model Reduction Benchmark Collection, www.imtek.de/simulation.

[6] R. W. Freund, Krylov-subspace methods for reduced order modeling in circuit simulation, Journal of Computational and Applied Mathematics vol. 123, pp. 395–421, 2000.

[7] A. Odabasioglu, M. Celik, L. T. Pileggi, "PRIMA: Passive Reduced-order Interconnect MAcromodeling Algorithm", IEEE Trans. Comp. Aid. Desig. Int. Circ. Syst. 17(8), pp. 645-654, 1998.

Table I
Characteristics of MEMS Benchmarks

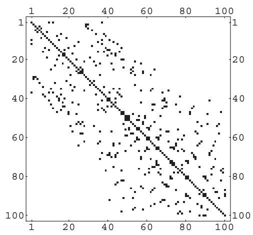
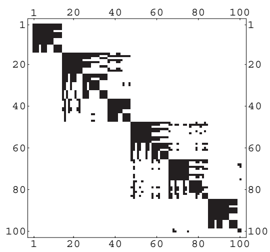
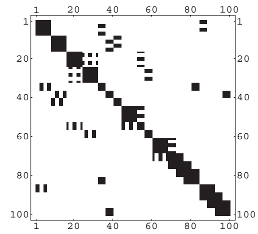
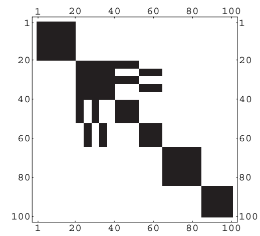
Model	System dimension	Structure of the system matrix G (beginning part)
Tunable optical filter (low order elements)	1 668	
Tunable optical fiter (low order elements)	106 437	
Pyrotechnical micro-thruster (low order elements)	26 360	
Pyrotechnical micro-thruster (high order elements)	79 171	

Table I
Characteristics of MEMS Benchmarks

Table II
Characteristics of MOR Methods and Timing

System dimension	nnz(G)	nnz(L)	Reduction time (s) - mor4ansys	Reduction time (s) - RW
1 668	6 209	$3.215*10^4$	0.07	0.11
106 437	1 406 808	$2.025*10^7$	24.1	414.14
26 360	265 113	$5.3*10^6$	10.5	238.12
79 171	2,215,638	$4.6*10^7$	128	Out of memory after 2.5h

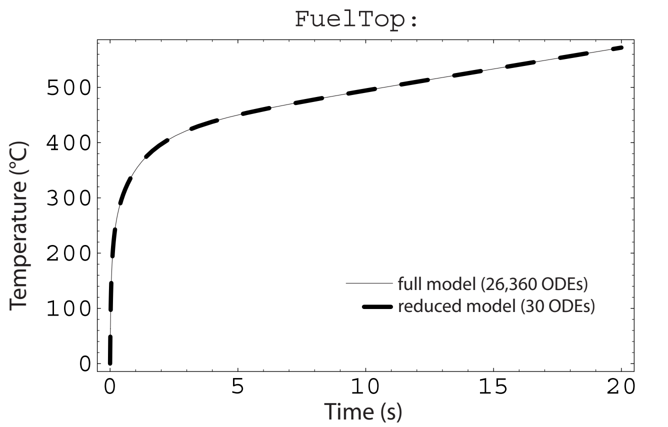


Fig. 1 Step response of the full scale and reduced models in a single output node.