

Performance Evaluation of Linear Solvers Employed for Semiconductor Device Simulation

S. Wagner*, T. Grasser*, and S. Selberherr[◦]

* Christian Doppler Laboratory for TCAD in Microelectronics
at the Institute for Microelectronics

[◦] Institute for Microelectronics, TU Vienna
Gußhausstraße 27–29, A-1040 Wien, Austria
Email: Wagner@iue.tuwien.ac.at

Abstract

We present the motivation and results of a benchmarking project for linear solvers employed for semiconductor device and circuit simulation. Based on examples coming from current research projects, the performance of a specific set of linear solvers is evaluated. In rare circumstances the results show that it is important to choose the appropriate type of solver for different kind of simulations.

1 Motivation and Introduction

A significant part of the computation time of a numerical semiconductor device simulation is spent on solving equation systems. Basically, such simulations require the solution of a nonlinear PDE system discretized on a grid. The nonlinear problem is solved by a damped Newton algorithm demanding the solution of a non-symmetric and sparse linear equation system at each step. Since the remarkably increased performance of today's average computers inspires to even more costly simulations (e.g. optimizations and mixed-mode device/circuit simulations), a further speed-up of the simulators is highly appreciated. In the course of this work we evaluated and compared the performance of several solvers. Rather than using a set of single matrices, this work is based on complete simulations with consistent settings, as typically encountered during daily work.

An in-house assembly and solver system has been developed, which is currently employed by the multi-dimensional device and circuit simulator MINIMOS-NT [1]. The assembly module [2] provides an API to the simulator models, several conditioning measures, sorting and scaling, which have been found to be essential for many solver types. Two iterative solvers, namely BI-CGSTAB and GMRES(M) [3] in combination with an ILU-preconditioner, as well as a Gaussian solver, all implemented in C++, are available. Specific properties for quality management are the main reasons for providing these solvers.

With regard to the growing demand for computational power, the on-going development of highly-optimized mathematical code must not be neglected. For that reason, the solver module has been equipped with an interface to external solvers. At the moment, two external modules can be employed: First, the Parallel Sparse Direct Linear Solver

#	Simulation	Type	Dim.	Ent. [%]	DC it	Remark
1	MOSFET	2D	2.704	3.24	17	test structure with $L_G = 1 \mu\text{m}$
2	Flash cell	2D	5.967	1.35	24	tunneling effects
3	Pin-diode	2D	6.335	1.40	13	optical generation model
4	Bjt transistor	2D	6.389	0.83	28	hydrodynamic equations
5	SA-LIGBT	2D	16.774	0.56	7/20	two $V_D = 0$ steps: 0 V, 5 V
6	SiGe HBT	2D	19.313	0.57	16	self-heating
7	Colpitts oscillator	C (1)	3.928	2.27	41	transient (400 steps)
8	Amplifier	C (1)	6.391	0.86	30	hydrodynamic equations
9	Ring oscillator	C (10)	25.246	0.36	29	transient (100 steps)
10	2-input nand gates	C (8)	146.614	0.06	8	transient (50 steps), init file
11	3-input nand gates	C (12)	219.920	0.04	9	transient (50 steps), init file
12	MagFET	3D	85.308	0.13	36	magnetic field
13	FinFET	3D	81.037	0.12	13	thin SOI finger
14	SOI	3D	87.777	0.13	10/13	two $V_D = 0$ steps: 0 V, 0.1 V
15	HBT	3D	119.098	0.09	67	first iteration scheme
			175.983	0.06	13	second scheme with selfheating
16	LD-MOSFET	3D	167.197	0.07	10	power device

Table 1: Six two-dimensional, five mixed-mode device/circuit (the number of devices is given in brackets), and five three-dimensional simulations were used for evaluating the solver performance. The dimension of the linear equation system, the number of non-zero entries, and the typical number of DC Newton iterations are given.

PARDISO [4, 5, 6], which provides a multi-threaded direct solver as well as a LU-CGS iterative solver implementation. Second, the Algebraic Multigrid Methods for Systems (SAMG) [7, 8], which not only provides multi-level algorithms, but almost the same iterative solvers as those of the in-house module. Both external packages are written in Fortran. Their only negligible overhead is a matrix storage format conversion.

2 Test Examples and Results

The 16 examples summarized in Table 1 were taken from current scientific projects at our institute consisting of field effect, bipolar, and silicon-on-insulator transistors. They were simulated on an IBM AIX 5.2 cluster (four nodes based on Power4+ architecture; 192 GB memory) and on a 2.4 GHz single Intel Pentium IV (1 GB memory) running under Suse Linux 8.2. For compiling and linking the native xlc/xlC/xlf compilers (32-bit; optimization level O5; linked against the ESSL library) and the Intel 7.1 compilers (IA32; optimization level O3) were used. The simulation time was measured with the *time* command, the fastest of three consecutive runs was taken.

The iterative methods still show a significant performance advantage over the direct solvers (e.g. example 12: ILU-BI-CGSTAB 288.78 s; PARDISO-LU one thread 2290.22 s; eight threads 641.92 s). However, the 1983 quotation “In 3D sparse direct solution methods can safely be labeled a disaster” [9] describes the experiences (in regard to both time and memory consumption) with the classically implemented, in-house LU factorization, but does not embrace the recent developments. Especially for mixed-mode device/circuit simulations the advanced direct methods show a significant performance advantage, even up to the highest dimensions (e.g. example 11: ILU-BI-CGSTAB 8865.75s; PARDISO-LU one thread 6486.90s; eight threads 5886.47s).

Two figures with specific results were chosen in order to sketch the complete evaluation scope. In Fig. 1 a comparison of different solvers for selected simulations on the Intel

computer is given. Due to the large simulation time differences, all times are scaled to the in-house ILU-BI-CGSTAB in the center of the graph. Interesting results are the superiority of the advanced implementations of LU factorization and iterative solvers for circuits and three-dimensional devices, respectively. The in-house GMRES(M) solver has advantages for circuits also, whereas the direct solver on the left hand side can in fact only be used for quality assessment of two-dimensional simulations.

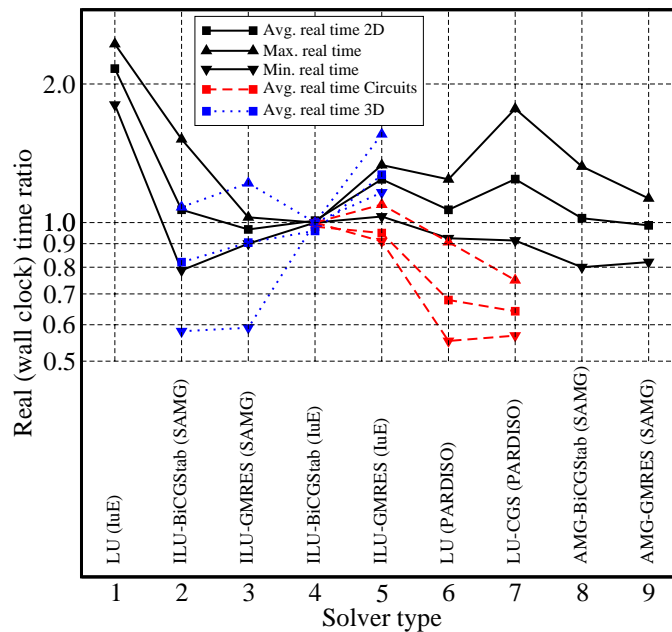


Fig. 1: Solving times (average, minimum, maximum) for selected simulations on the Intel computer. All times are scaled to the in-house ILU-BI-CGSTAB in the center.

To show the relative impact of multi-threading, the PARDISO-LU solving ratios (referring to the single-threaded version) against the number of processors/threads are shown in Fig. 2. For the three-dimensional examples, also the PARDISO-LU-CGS ratios are given. In addition to the real (wall clock) time required for solving the example, the cumulated user (CPU) times are shown, which are increasing due to the parallelization overhead. Whereas for two-dimensional device and circuit simulations too many processors can be even contra-productive, the marginal additional utility for three-dimensional simulations is drastically diminishing. Thus, for the average simulation four processors should be sufficient. Especially under scarce conditions, e.g. during optimizations, assigning two tasks per node of eight processors appears to minimize the real time effort.

3 Conclusion

We presented selected results from an exhaustive evaluation of linear solvers. The set of chosen solvers contains not only in-house codes, but accounts also for the on-going development of new solver modules and techniques. By benchmarking different kinds of complete simulation runs, respective conclusions can be drawn for the daily usage.

Acknowledgement

This work has been partly supported by *Infineon Technologies*, Villach, and *austriamicrosystems*, Graz, Austria.

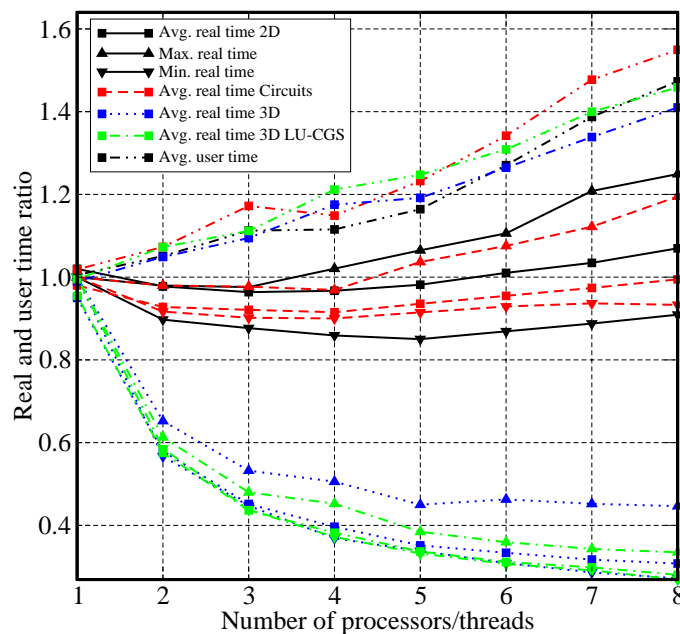


Fig. 2: The relative PARDISO real/wall clock times (average, minimum, maximum) and average user time versus the number of processors/threads on the IBM cluster.

- [1] Institut für Mikroelektronik, Technische Universität Wien, Austria, Minimos-NT 2.0 User's Guide, <http://www.iue.tuwien.ac.at/software/minimos-nt>, 2002.
- [2] S. Wagner, T. Grasser, C. Fischer, and S. Selberherr, in *Proc. 15th European Simulation Symposium ESS* (Delft, The Netherlands, 2003), pp. 55–64.
- [3] R. Barrett *et al.*, *Templates for the Solution of Linear Systems: Building Blocks of Iterative Methods* (SIAM, Philadelphia, PA, 1994).
- [4] G. Karypis and V. Kumar, *SIAM Journal on Scientific Computing* **20**, 359 (1998).
- [5] O. Schenk and K. Gärtner, *Future Generation Computer Systems* (2003), accepted, in press.
- [6] O. Schenk, K. Gärtner, and W. Fichtner, *BIT* **40**, 158 (2003).
- [7] T. Füllenbach (now: Clees), K. Stüben, and S. Mijalković, in *Proceedings 2000 International Conference on Simulation of Semiconductor Processes and Devices*, IEEE (2000), pp. 225–228.
- [8] T. Clees and K. Stüben, in *Proc. Challenges in Scientific Computing*, Springer, Berlin (2002), pp. 110–130.
- [9] R. Bank, D. Rose, and W. Fichtner, *ED-30*, 1031 (1983).