

Practical Use of a Hierarchical Linear Solver Concept for 3D MOS Device Simulation

O. Heinrichsberger, M. Thurner[†], and S. Selberherr

Institute for Microelectronics, TU Vienna
Gußhausstraße 27-29, A-1040 Wien, AUSTRIA

[†]Campus Based Engineering Center Vienna, Digital Equipment Corporation
Favoritenstraße 7, A-1040 Wien, AUSTRIA

Abstract

Three-dimensional device simulations of sidewall trench-isolated MOSFETs involve linear systems of equations that are extremely ill-conditioned if the geometric channel length exceeds $2\mu m$. Since conventional preconditioned iterative methods tend to fail in such cases we have implemented a robust iterative linear solver using a hierarchy of incomplete factorizations of the coefficient matrix.

1. Introduction

It is well-known that preconditioned iterative methods for the solution of linear systems are a key part of every three-dimensional semiconductor device simulator. Convergence speed, efficiency of implementation and robustness for very ill-conditioned matrices are preconditions for any linear system solver in a production environment. Recently the BiCGStab method as iterative routine [6] and the incomplete factorization method with threshold pivoting (ILUT) as preconditioner [4] have evolved as the most promising methods for solving even the worst conditioned linear systems in device simulation [1].

The condition numbers of matrices in 3D device simulators range from moderate values (e.g. the majority continuity equation in MOSFETs) to extremely ill-conditioned ones (e.g. the minority continuity equation for long channel MOSFETs). To cope with this variety of condition numbers, a hierarchical strategy (not to be confused with nested iterative methods) for combining different preconditioners has been proposed [1]. In this work we report our experience with a hierarchical linear system solver used in the three-dimensional part of MINIMOS 5.

2. Incomplete LU Factorizations

Incomplete LU factorizations of the nonsymmetric coefficient matrix are known to be the only viable preconditioning methods to solve the discrete continuity equations reliably. Other methods such as polynomial or hierarchical bases preconditioners are only applicable to relatively well-conditioned problems. There are various ways to

factorize a sparse matrix incompletely: The standard method is to define some symbolic sparsity pattern for the incomplete factors and then computing matrix entries according to the pre-defined pattern. In the simplest case the original sparsity pattern of the matrix is used, denoted by $ILU(0)$. For block diagonal matrices the sparsity pattern may be augmented easily by additional diagonals along the original sparsity pattern. If one additional diagonal is allocated along each existing diagonal we call this factorization $ILU(1)$.

A more general class of ILU factorizations is obtained by keeping fill entries in the ILU factors only when they exceed some threshold, thus the sparsity pattern is defined during factorization. Clearly, for smaller values of the *drop threshold* parameter TOL a better ILU factorization with more fill-in can be created than for a larger TOL.

In this approach we precondition the BiCGStab method by two major *preconditioning levels* (Fig. 1):

- A vectorizable ILU preconditioner with positional dropping as the lowest hierarchy level,
- A parametric, non-vectorizable ILUT preconditioner in right position for higher levels.

In the first hierarchy we use either $ILU(0)$ implemented in conjunction with the Eisenstat matrix-vector multiplication method to save about 30% of arithmetic work [2] or $ILU(1)$. Both preconditioners are applied in split position.

The computation of the ILUT preconditioner for the second hierarchy level depends on two parameters: TOL which defines a drop tolerance weighted with the actual matrix rowsum for an individual matrix entry and FIL which determines the maximum number of matrix entries per row of an incomplete factor. TOL controls mainly the robustness of the preconditioner, FIL is necessary to limit memory usage, however, both parameters influence the robustness of the preconditioner.

To provide robustness for very ill-conditioned linear systems we use tolerances smaller than 10^{-5} and a fill-in of 20 beside the original sparsity pattern. This choice (in the two-dimensional parameter space) seems to yield an optimum in CPU time for our specific application (Fig. 2). An iteration limit of 400 for $ILU(0)$ and 100 for $ILU(1)$ is used as a switch between the hierarchies. For iteration termination we require the residual (left split preconditioned residual in case of ILU or true residual in case of ILUT) to be less than 10^{-8} to ensure convergence of the outer Gummel algorithm.

For ILUT factorization a row oriented compressed sparse matrix format is used. The factorization code is inherently sequential in nature and hence does not vectorize. Therefore the factorization must be implemented as efficient as possible. Efficiency can be exploited mainly by detecting unimportant matrix entries as early as possible and by an efficient sort algorithm to sort out the FIL largest elements in a row.

The package is written in FORTRAN 77 (iterative solver and preconditioner) and C (memory management and hierarchy sequencing) and has successfully been used on several platforms such as VAX/VMS, Alpha AXP/OpenVMS, DECstation/ULTRIX and HP9000/HP-UX.

The solver hierarchy is user configurable, if e.g. the user is equipped with a-priori knowledge on the conditioning of the problem the solver can easily be directed to automatically leap over less robust levels (in case of an ill-conditioned system). To our experience a two-level hierarchy (see Figure 1) is sufficient, hierarchies with more than one ILUT level have not proven advantageous in our simulations.

L_G (μm)	ILU(1)		ILUT(10^{-5})	
	ITMIN	CPU (s)	ITMIN	CPU (s)
1	190	3504	13	1770
2	350	5232	14	1882
3	749	6983	17	1880
4	805	8584	24	2181
5	1924	14725	26	2264
6	1046	8636	28	2184
7	2627	17226	29	1976
8	1804	13274	32	2251
9	2223	15946	35	2578

Table 1: Comparison of iteration counts and total simulation times (VAX 6420) for the first Gummel iteration using ILU(1) and ILUT(10^{-5}) preconditioned BiCGStab solvers for MOSFETs with various geometric channel lengths.

3. Solver performance

The sidewall trench isolation structure of the devices under consideration is given in Fig. 3 (width direction). The channel is limited in width direction by a shallow trench. Due to the small sidewall overlap a parasitic device is present with a smaller effective gate thickness thus lowering the threshold voltage [5]. The main doping profile as well as the field and sidewall doping profiles are constructed from two-dimensional slices. Device performance has been studied in dependence of gate length and width, sidewall overlap, field and sidewall doping.

Devices with different gate lengths (from $10\mu\text{m}$ down to $0.5\mu\text{m}$) have been simulated and their characteristics compared with measurements. We encountered serious convergence problems such as stagnation or iteration counts of several thousands for long channel devices ($L > 4\mu\text{m}$) at the beginning of Gummel's algorithm when using an ILUT(0) or ILUT(1) preconditioner. For CPU times and iteration counts of the minority continuity equation see Table 1. It is worth mentioning that this ill-conditioning is quite independent of biasing, specifically in the subthreshold region which was the domain of major interest during this investigation (see Fig. 4 for the short channel device showing the enhanced conductivity due to the large gate sidewall overlap).

In contrast to the ILU(0,1) preconditioner the CPU times when using the ILUT preconditioner are almost independent on the device channel lengths and much smaller. The iteration count in the ILUT level can be kept less than one hundred in all cases. Even for the largest problems the CPU time consumption is quite moderate provided that memory is large enough to keep the ILUT preconditioner in core. At least 32MB main memory is necessary for the preconditioner to solve problems involving grids with several 10^5 mesh points. The characteristic given in Fig. 4, containing 40 bias points has been simulated in 7 hours.

References

- [1] M. Driessen, H.A.v.d. Vorst, *Bi-CGSTAB in Semiconductor Modeling*, Proc. SIS-DEP (1991).
- [2] O. Heinrichsberger et al., *Fast Iterative Solution of Carrier Continuity Equations for Three-Dimensional Device Simulation*, SIAM J.Sci.Stat.Comp., Vol. 13, No. 1, pp. 289-306 (1992).
- [3] C. Pommerell, *Solution of Large Unsymmetric Systems of Linear Equations*, Doctoral Thesis, ETH Z^{urich} (1992).
- [4] Y. Saad, *Preconditioning Techniques for Nonsymmetric and Indefinite Linear Systems*, J.Comp. Appl.Math., Vol. 24, pp. 89-105 (1988).
- [5] S.M. Sze, *High-Speed Semiconductor Devices*, John Wiley & Sons, p. 177 (1990).
- [6] H.A.v.d. Vorst, *A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Systems*, SIAM J.Sci.Stat.Comp., Vol. 13, No. 2, pp. 631-644 (1992).

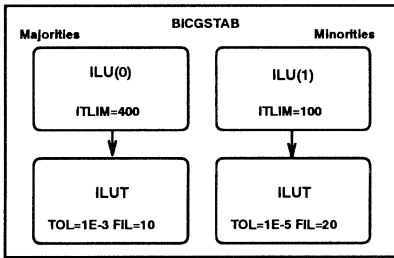


Fig. 1: Hierarchy Setup.

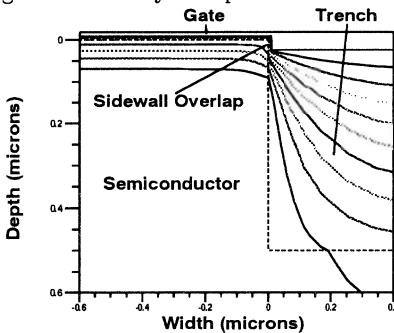


Fig. 3: Device geometry.

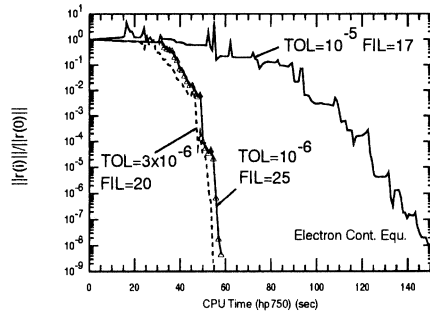


Fig. 2: Dependence on TOL.

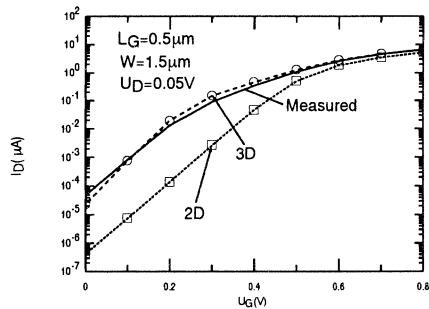


Fig. 4: Subthreshold characteristic.