

Analysis of Parallel Numerical Libraries to Solve the 3D Electron Continuity Equation

Natalia Seoane and A.J. García-Loureiro

University of Santiago de Compostela, Dept. Electronics and Computer Science,
Campus Sur, 15782 Santiago de Compostela, Spain
Phone: +34 981 563100 Ext. 13568, Fax: +34 981 528012
`natalia@dec.usc.es`, `antonio@dec.usc.es`

Abstract. In this paper we show an analysis of different parallel numerical libraries for solving the linear systems associated to the electron continuity equation from the 3D simulation of the semiconductor devices. We use domain decomposition techniques, such as Additive Schwarz, Multicolor SOR or Schur Complement methods, in order to find the best method of resolution considering the minimization of the execution time. The results were obtained in a Beowulf system with Myrinet 2000 network with MPI standard for message-passing communication.

1 Introduction

In this paper an analysis is presented of different parallel numerical libraries employed in resolution of sparse linear systems in 3D semiconductor device simulation. The main goal of this analysis is the selection of parameters to minimize execution time. The sparse matrices employed in our analysis are those resulting from the discretization of electron continuity equation for HEMT devices. In our case, these matrices have 29012 rows and 398102 nonzero elements. A similar study could be performed for any other type of semiconductor devices. Our results have been obtained in a Beowulf system with Myrinet 2000 network, using MPI for message-passing communication.

2 Numerical Libraries

The working sets are numerical libraries based on direct methods, SuperLU, and iterative methods, PSPARSLIB, PETSc and Aztec, analyzing in each case the parameters with the largest impact on execution time, such as solvers based on Krylov methods, parallel preconditioners, fill-in processes in Incomplete LU Factorizations, the Krylov subspace size and the number of processors employed.

2.1 SuperLU

SuperLU[1] is a general purpose library for the direct solution of large, sparse linear systems. The library performs a LU decomposition with partial pivoting

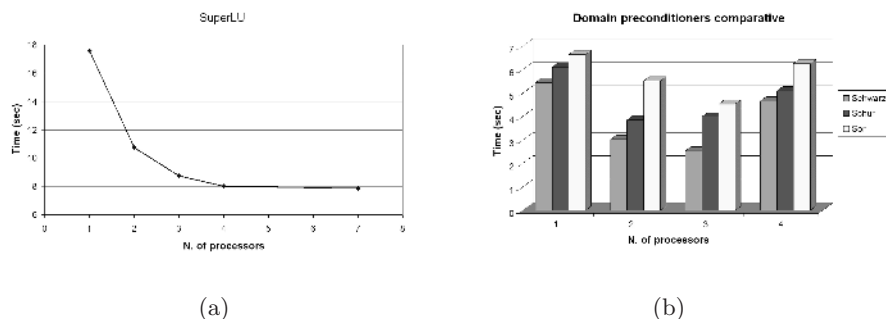


Fig. 1. (a) Factorization time for SuperLU library, (b) Comparison between the Additive Schwarz, Multicolor SOR and Schur Complement methods for PSPARSLIB

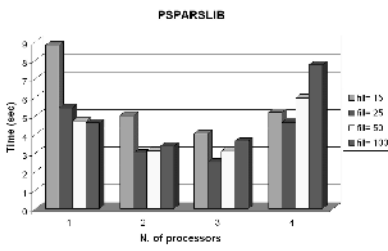
and triangular system solves through forward and back substitution. We can choose between sequential or parallel versions.

Using the distributed memory version, we analyze this library, solving in parallel a complete LU Factorization. Solving time decreases with the number of processors employed, which is becoming less marked (Figure 1(a)). For this type of matrix the use of this library is not very profitable in comparison with any of the analyzed libraries based on iterative methods.

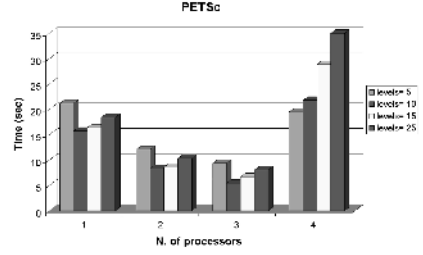
2.2 PSPARSLIB

The solvers included in PSPARSLIB[2] are CG, GMRES, FGMRES, DQGMRES, BCGSTAB and TFQMR. It uses domain decomposition preconditioners[3], such as Additive Schwarz, Multicolor SOR and Schur complement methods [4].

Making a solver comparison, we find that GMRES obtain the lowest solving times. From the available preconditioners, Additive Schwarz is the most appropriate, as can be deduced in Fig. 1(b), where a comparison among the preconditioners named above is shown. For the computation of the internal points we use an incomplete LU Factorization depending on two parameters: the fill value ($2 \cdot \text{fill}$ will be the maximum number of fill-in elements) and a drop tolerance (fixed to $10e^{-4}$). The lowest solving times were obtained for a value of $\text{fill} = 25$. An example of this can be observed in Fig. 2(a), where the influence of fill value in solving time is represented for GMRES. The influence of Krylov subspace size is very small, since variations of 5% in the solving time are found by changing two magnitude orders in the Krylov subspace size value. With respect to the number of processors, the tendency shows minimum execution times with 3 processors. There are several reasons because using 4 processors is not a good choice. Firstly, communication times between processors become increasingly high when they are compared to the computational time. This is due to the size of local matrices which is getting too small. Moreover, reducing the size of matrices involves less

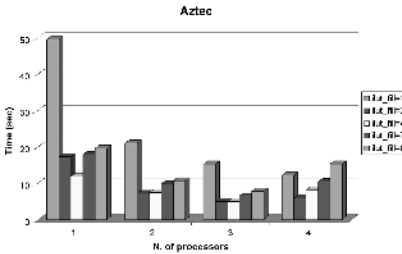


(a)

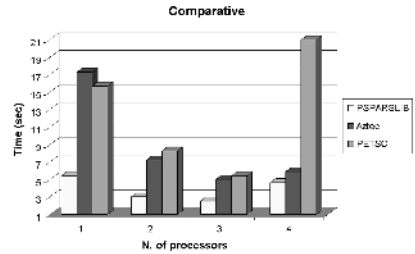


(b)

Fig. 2. Dependence of solving time with the fill value for GMRES solver with Additive Schwarz preconditioning for: (a) the PPARSLIB library and (b) the PETSc library



(a)



(b)

Fig. 3. (a) Dependence of solving time with the `ilut_fill` parameter for the Aztec library, (b) Comparison between PPARSLIB, PETSc and Aztec for the best solving times

number of internal nodes and an increase of interface nodes. This is translated into a higher cost in both communication and computation times.

2.3 PETSc

PETSc[5] includes several solvers such as CG, GMRES, TCQMR, BCGS and TFQMR. Some preconditioners included in PETSc are Jacobi, Block Jacobi, SOR and Additive Schwarz. From all of the solvers, we are only going to study those common to the ones existing in PPARSLIB in order to be able to make a comparative study. For the same reason, we choose the Additive Schwarz preconditioner, solving the internal points through an incomplete LU Factorization depending on a certain level of fill (*level* is the column number round the diagonal in which the fill-in is allowed).

The lowest solving times are found for the GMRES solver, when the levels of fill are between 10 and 12. With respect to the influence of the number of

processors employed, we find a decrease of the solving time when we increase the number of processors, although this tendency is not true for more than 3 processors, as can be observed in Fig. 2(b) relative to the GMRES solver.

2.4 Aztec

Aztec[6] includes a number of Krylov iterative methods such as CG, GMRES, TFQMR and BCGSTAB. These Krylov methods are used in conjunction with various preconditioners such as polynomial or domain decomposition methods.

In this library the same solvers of PETSc are studied, using as a preconditioner the domain decomposition method Additive Schwarz, the inner solve being an incomplete LU factorization depending on the parameter `ilut_fill` (which indicates that the final factorization can contain as most `ilut_fill` times the number of no zero elements from the original matrix) and a drop tolerance (fixed to zero).

The lowest execution times are found for BCGSTAB solver, the differences between BCGSTAB solver and the other solvers studied are considerable. The minimum execution times are achieved using 3 processors, with `ilut_fill` values between 2 and 4. An example of this can be observed in Fig. 3(a) in which we represent for the BCGSTAB solver the influence of `ilut_fill` in solving time.

Acknowledgments. This work was partly supported by the Spanish Government (CICYT) under the project TIC 2001-3694-C02-01

References

1. Demmel J.W., Gilbert J., Li X.S.: SuperLU Users Guide (1999)
2. Saad Y., Lo G.C., Kuznetsov S.: PPARSLIB Users Manual: A portable library of parallel sparse iterative solvers. Univ. of Minnesota, Dept. Computer Science (1997)
3. Saad Y.: Iterative Methods for Sparse Linear Systems. PWS Publishing Co.(1996)
4. Saad Y., Gen-Ching L.: Iterative Solution of General Sparse Linear Systems on Clusters of Workstations. Univ. of Minnesota, Dept. of Computer Science (1996)
5. Balay S., Gropp W.D., McInnes L.C., Smith B.F.: PETSc 2.0 Users Manual, ANL-95/11 - Revision 2.0.24. Argonne National Laboratory (1999)
6. Hutchinson S. A., Shadid J., Tuminaro R.S.: Aztec User's Guide, SAND95-1559. Sandia National Laboratories (1995)