The Implementation and Speed-up of Coloured SOR Methods for Solving the 3D Poisson Equation on an Array of Transputers

Andrew R. Brown, Don Reid, Asen Asenov and John R. Barker

Nanoelectronics Research Centre,
Department of Electronics and Electrical Engineering,
University of Glasgow,
Glasgow, G12 8QQ, UK

Abstract
In this paper we present four different SOR variants for solving the 3D Poisson equation on an array of transputers. The performance of all variants has been tested and compared in terms of number of iterations and global computation time for different configurations of the transputer system. The parallel performance of the algorithms has also been evaluated and compared to a theoretical speed-up model.

I. Introduction

The realistic semiconductor device simulation (both classical, Monte Carlo or quantum mechanical) in many cases requires a 3D solution of the Poisson equation and leads to enormous problem sizes [1]. The single processor implementation of the corresponding 3D codes is limited by both the processor speed and the huge memory-access bottleneck. In the foreseeable future a significant low cost improvement in computer performance will only be available through Multiple Instruction Multiple Data (MIMD) systems (many of them transputer based), for which the necessary speed-up derives from the use of parallel processors sharing a large distributed memory. The point and block Successive Over Relaxation (SOR) methods are promising candidates for 3D parallel implementation on such computers. Although the recursive character of the original SOR method seems to be a serious impediment [2], for a large class of linear systems arising from finite difference, and in particular cases [3] from finite element approximations of the Poisson equation, the multicolour ordering of the grid points leads to easily parallelizable versions of the SOR method [4,5].

Here we present a systematic approach to the parallel implementation of scalable point and block black/red 3D SOR Poisson solvers on a 2D arrays of transputers for the purposes of semiconductor device simulation. Utilising the power and the flexibility of the Parsytec Supercluster Model 64 a wide range of experiments have been made both to compare the performance of the different SOR variants and to choose the optimum 2D transputer configuration for mapping 3D problems. The recently developed detailed performance theory [6] has been applied to underpin the experimental solver design.

II. Model Problem and Partition

The Poisson equation used in the semiconductor device simulations

$$\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}+\frac{\partial^2\psi}{\partial z^2}=-\frac{\rho}{\varepsilon} \qquad (1)$$

may be nonlinear or linear, depending on whether the charge density ρ is a function of the electrostatic potential Ψ or not. In the drift-diffusion and the hydrodynamic simulation approaches it is mainly used in a nonlinear form . In Monte Carlo and quantum mechanical simulation however it is usually enough to solve the linear Poisson equation.

In this study for simplicity and clarity we consider the linear 3D Poisson equation in a unit cube Ω with Dirichlet boundary conditions. A uniform grid is used to discretize the region Ω into Ω_h. The approximation of the second-order derivatives by the second-order accurate central differences on Ω_h leads to a set of *nxnxn* algebraic equations

$$\psi_{i-1,j,k} + \psi_{i+1,j,k} + \psi_{i,j-1,k} + \psi_{i,j+1,k} + \psi_{i,j,k-1} + \psi_{i,j,k+1} - 6\psi_{i,j,k} = b\frac{\rho_{i,j,k}}{\varepsilon} \tag{2}$$

for $i,j,k = 2,3, \ldots n-1$, where n is the number of points on a side of Ω_h, and b is a coefficient depending on n.. In most of the following experiments ψ was set to one, ρ was set to zero and the initial conditions were $\bar{\psi} = 0$.

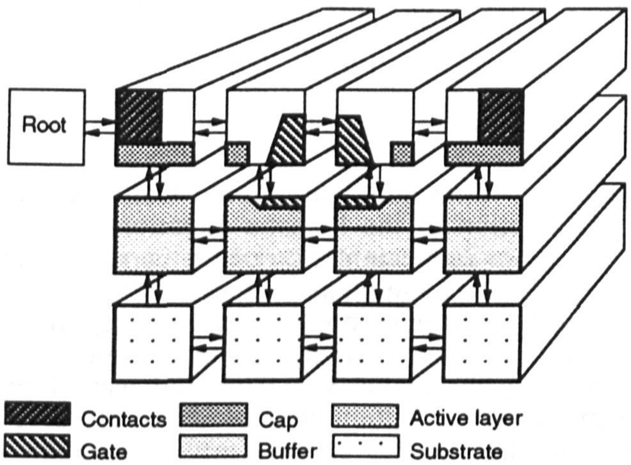

Fig. 1 Mapping of a 3D semiconductor device simulation domain on an 2D array of transputers

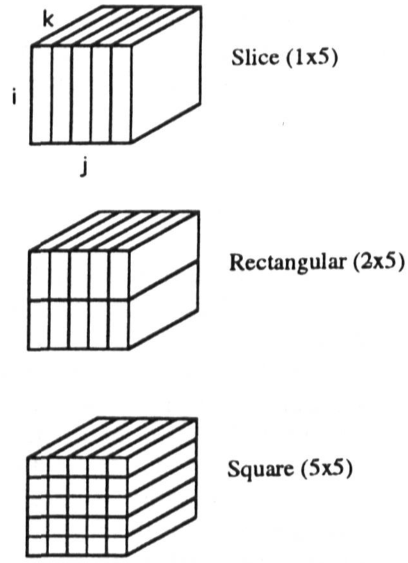

Fig. 2 Partition of a 2D discretization grid on a 2D array of transputers

Although it is intuitively clear that the best environment for solving topologically rectangular 3D problems is a 3D array of processors, we are restrict to a 2D array of *N*x*M* transputers. This reflects the connectivity of transputers, which have only four links, and offers a simple way of mapping a topologically rectangular 3D grid, where the overlap of subdomain boundaries is essential. The transputers have a natural ordering $p = 1,2,\ldots N$ and $q = 1,2,\ldots M$ Fig. 1. One additional 'root' transputer performs all management and synchronisation. The grid is partitioned into *N*x*M* subdomains along two of the spatial dimensions (i,j) and each of the subdomains involves all corresponding points in the third dimension k.. Examples of slice (*1*x*M*), rectangular (*N*x*M*) and square (*N*x*N*) partitions are given in Fig. 2. A universal communication harness GARH [6] supports all necessary global and local communications.

III. Back/Red SOR Variants

The implementation of the Point SOR method with Natural ordering in i,j directions on each processor may in many cases cause divergence for problems which behave well in a serial implementation [7]. This may be avoided if a black/red SOR variant is considered. In this approach each partition sub domain $\Omega_h^{p,q}$ is decomposed into two further subdomains - black

$\Omega_h^{Bp,q}$ and red $\Omega_h^{Rp,q}$. The nodes in the black and red subdomains are updated simultaneously and the overlapping boundaries are exchanged before each updating.

When a 2D array of processors is used, it is enough to apply the black/red ordering only in the *i,j* plane which leaves some degree of freedom in the *k* direction. To explore this freedom we consider four different parallel SOR schemes namely:

PSORBR$_{ij}$N$_k$ - Point SOR with Black/Red ordering in *i,j* direction and Natural ordering in *k* direction. (Fig. 3 a)

PSORBR$_{ijk}$ - Point SOR with Black/Red ordering in all three directions (Fig. 3 b).

PSORBR$_{ij}$A$_k$ - Point SOR with Black/Red ordering in *i,j* direction and Alternating directions in *k* direction.

BSORBR$_{ij}$T$_k$ - Block SOR with Black/Red ordering in *i,j* direction and tridiagonal equations solution in *k* direction.

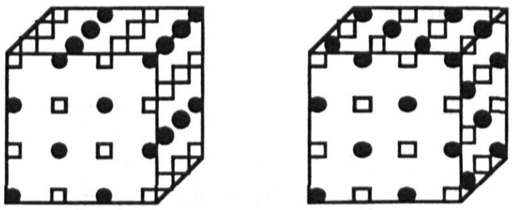

Fig. 3 Black/red ordering in the 3D case. (a) full black/red ordering (b) black/red ordering only in i,j plane

Table 1

Mesh size n	Natural ordering theory	PSORBR	PSORBRN	PSORBRA	PSORBRT
10	16	15	15	15	11
20	32	32	32	30	22
30	47	47	47	43	33
40	62	61	61	56	41
50	76	74	74	70	51

Table 2

Mesh size n	PSORBR with ω_e	PSORBRN with ω_b	PSORBRN with ω_e	PSORBRA with ω_e	BSORBRT with ω_b	BSORBRT with ω_e
10	0.087	0.093	0.087	0.087	0.106	0.077
20	0.635	0.635	0.635	0.595	0.677	0.552
30	2.184	2.184	2.184	1.998	2.346	1.985
40	5.450	5.539	5.450	5.003	5.888	4.828
50	11.278	11.582	11.278	10.668	12.163	10.169

The main disadvantage of the black/red ordering with all variants described above is that the natural ordering SOR theory [7] does not hold, which makes it difficult to estimate the optimum relaxation coefficient ω a priory. In order to compare more precisely our four different SOR variants, experimental optimum values for ω_e have been found by using a one dimensional minimum search based on the Golden Section method. In Table 1 the numbers of iterations providing an accuracy δ=0.001 are given for both ω_e and the theoretically predicted relaxation coefficients ω_b [7]. As can be expected the block variant BSORBRT leads to a significant reduction in the number of iterations but at the expenses of an increasing number of calculations per grid point. This point becomes clear from Table. 2 where the execution times in seconds are compared when all 64 transputers of a Parsytec Super Cluster were used, configured in an 8x8 array.

IV Speed-up analysis

To evaluate the parallel potential of the considered SOR variant and the optimal processor configuration we use the relative speed-up defined as the ratio between execution times on an array of processor and the execution time on a single processor. It is obvious that according to this definition the maximal available speed-up is equal to the number of processors on which the algorithm runs. The speed-up model developed in [6] for scalable 2D linear solvers implemented on an array of transputers was extended to the 3D case and takes into account the details of the solution domain partition, all global and local communication overheads and the computation time in the linear and nonlinear case. All parameters of the performance theory have been extracted from independent measurements.

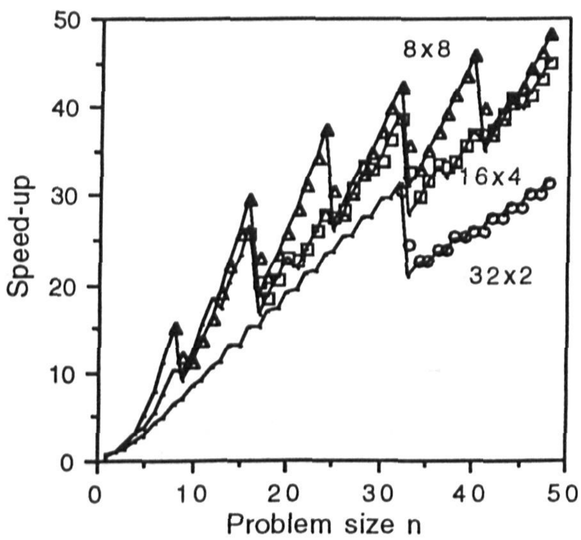

Fig. 4 Speed-up as a function of the problem size for different configurations of 64 transputers (PSORBRN variant)

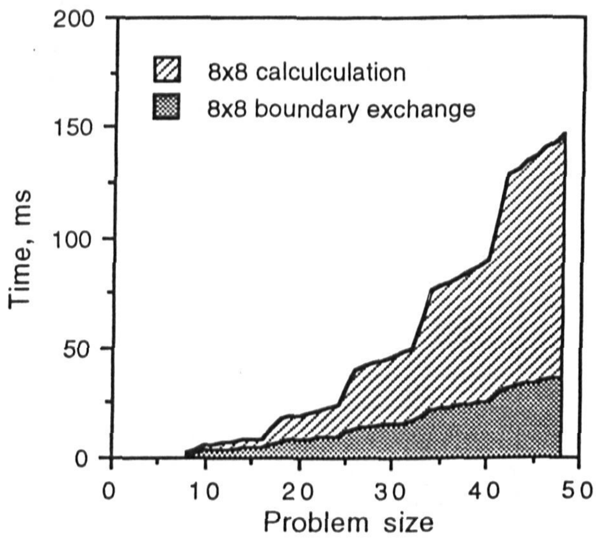

Fig. 5 Balance between calculation time and local communication time in the case of 8x8 transputers (PSORBRN)

The measured and calculated speed-up for three different configurations of 64 transputers is given in Fig. 4 for the PSORBRN variant as a function of the problem size. The picture shows very good agreement between the measured and predicted performance even in the finest details. The square partition which minimises the local communications shows the best performance for problem sizes n which are divisible by the transputer array size. Fig. 5 gives an idea for the balance between the local communication and calculation time for a single iteration in the 8x8 case. Finally Fig. 8 illustrates how the performance theory may be used to estimate the expected speed up for a particular problem size (n=50) mapped on a large number of transputers. It is clear that for large transputer systems the BSORBRT variant is superior to all other variants because of the higher calculations/ communications ratio.

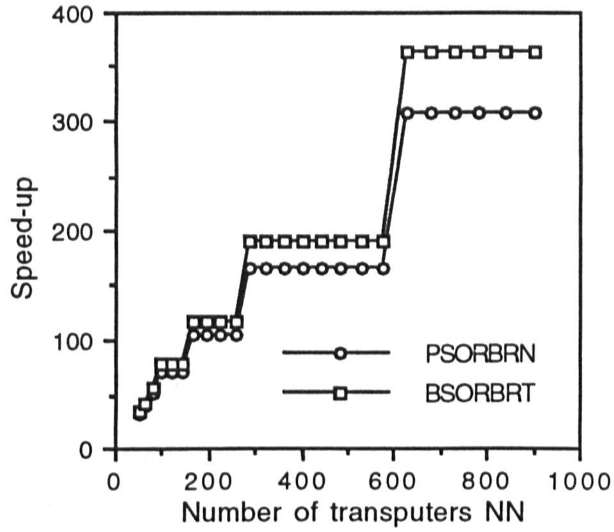

Fig. 6 Speed-up of 50x50x50 problem on a large array of transputers

V. Conclusions

Four variants of a black/red SOR method for solving the 3D Poisson equation on array of transputers have been implemented and tested. The behaviour of PSORBR and PSORBRN variants are very similar in terms of iterations required to achieve a certain accuracy. The third point variant PSORBRA slightly reduces the number of iterations. The block BSORBRT variant reduces the number of iteration by more than 25% but at the expense of a larger calculation time per iteration. Although for a medium size transputer system all four variants are very similar in terms of global computational time, our speed-up analysis shows that on a large array of transputers the advantages of BSORBRT will become more pronounced.

References

[1] H.C. Chan and T-J. Shien, "A Three-Dimensional Semiconductor Device Simulator for GaAs/AlGaAs Heterojunction Bipolar Transistor Analysis," IEEE Trans. Electron Dev., Vol. 38, No. 11, pp.2427 - 2432, 1991.

[2] L. Adams and J. Ortega, "A multicolour SOR method for parallel computation," in Proc. 1982 International Conference on Parallel Processing, Bellaire, MI pp.53-56, 1982.

[3] A. Asenov, D. Reid, J.R. Barker, N. Cameron and S.P. Beaumont, "Application of quadrilateral finite elements for simulation of recess T-gate MESFETs and HEMTs", in this Preceedings, 1993.

[4] D.J. Evans, "Parallel S.O.R. iterative methods", Parallel Computing, Vol. 1, No. 1, pp.3-18, 1984.

[5] U. Block, "Block colouring schemes for SOR method on local memory parallel computers", Parallel Computing, Vol. 14, No. 1, pp.61-75, 1990.

[6] A. Asenov, D. Reid and J.R. Barker, Speed-up of scalable iterative solvers implemented on an array of transputers, Parallel Computing, in press.

[7] D. Kamowitz, "SOR and MGR[v] experiments on the Crystal multicomputer", Parallel Computing, Vol. 4, No. 2, pp.117-142, 1987.

[8] R.S. Varga, "Matrix Iterative Analysis", Printice-Hall, Englewood Cliffs, NJ, 1962.