

A PARALLEL MULTIGRID SOLVER FOR SEMICONDUCTOR DEVICE EQUATIONS

X.Han, D.M.Barry, and M.J.Howes

Department of Electronic and Electrical Engineering, University of Leeds, LS2 9JT, UK

Email: een5xh@leeds.ac.uk, dmb and mjh@elec-eng.leeds.ac.uk

Abstract

This paper presents a parallel multigrid algorithm for solving semiconductor device equations in two-dimensions and its implementation on a MIMD parallel machine with distributed memory. The numerical experiments of a GaAs MESFET device simulation demonstrate the combined high efficiency of the domain decomposition method and the multigrid method. The parallel multigrid method using 16 processors is up to 60 times faster than a single-grid iterative solver using a processor.

I. INTRODUCTION

Semiconductor device simulations are an important tool for both physicists and device design engineers to analyse physical phenomena inside semiconductor devices and to predict the performance of new devices prior to fabrication. These physical models require the solutions of the Poisson equation for electrical potential and the current continuity equations for electron and hole concentrations. More complex models will also solve for the energy and temperature distributions. Numerical techniques must be used to solve these equations which require extensive computing power. The increasing complexity of both devices and physical models have challenged the available computing resources.

Many attempts have been made to design fast simulations. Multigrid (MG) methods offer a fast and robust iterative method for solving PDEs and have found applications in semiconductor device simulation [1-3]. The simulation of semiconductor devices possesses an inherent parallelism which requires many repeated operations on different grid points. It is clear that by exploiting the parallelism of the numerical algorithms leads to good speed ups of the simulation. The direct solution method has been used on MIMD parallel computers for device simulation [4], however this does not show good efficiency in the case of normal grid sizes. The Jacobi-SOR, Frankel iterative methods[5] and conjugate gradient iterative methods[6] have been implemented on SIMD Connection Machines. This approach is limited when modelling the irregular structures of modern devices. Transputer networks have also been used in semiconductor device simulation using a Monte Carlo method in [7] and finite difference methods in [8].

In this paper we present for the first time a parallel implementation of the multigrid iterative method for the solution of two-dimensional device equations on a medium-grain MIMD parallel machine with distributed memory and demonstrate its parallel efficiency for device simulation.

II. SEMICONDUCTOR DEVICE EQUATIONS

The semiconductor model used is based on the drift-diffusion approximation. It includes the Poisson equation, the current continuity equations for electrons and holes. The coupled system consists of an elliptic differential equation and two parabolic partial differential equations with dependent variables potential ψ , electron concentration n and hole concentration p . After discretisation using a finite difference scheme on a rectangular grid with N grid points, together with boundary conditions, we obtain a set of

$3 \times N$ equations with $3 \times N$ variables ψ , n , and p . We can write them in symbolic form as

$$F(\mathbf{u}) = f \quad (1)$$

where F denotes the nonlinear difference operators, f a constant and \mathbf{u} is a matrix including ψ , n , and p . The Gauss-Seidel (GS) technique coupled with successive over-relaxation (SOR), GS-SOR, is used for the Poisson equation along with successive under-relaxation (SUR), GS-SUR, for the continuity equations. The coupled Poisson and the continuity equations are solved using Gummel's approach[9].

III. MULTIGRID METHOD

It is obvious that PDEs need to be solved on a fine-enough grid to obtain accurate solutions. In semiconductor simulation a large truncation error may cause convergence problems. In some circumstances the larger the grid spaces the smaller the time step is required to maintain solution stability [10]. Classical iterative methods slow down with increasing grid point number. Multigrid iterative methods are highly efficient solvers for PDEs, in which the combination of fine grid relaxation and coarse grid correction produces a fast convergence rate. The multigrid full approximation scheme (MG-FAS) [11] is used in this work. To solve the system $F(\mathbf{u})=f$ the MG method uses a sequence of grids, $G_k(1 \leq k \leq K)$, where G_1 is the finest grid and G_K is the coarsest grid. There exists a system $F_k(\mathbf{u}_k)=f_k$ on each grid G_k . P_{k+1}^k is a prolongation operator from a coarse grid G_{k+1} to a fine grid G_k and R_{k-1}^k is a restriction operator from a fine grid G_{k-1} to a coarse grid G_k . MG-FAS cycles may be defined as follows.

- (a) Interpolate u_1 and f_1 to each of the grids and solve $F_k(\mathbf{u}_k)=F_k(R_{k-1}^k \mathbf{u}_{k-1}) + R_{k-1}^k(f_{k-1} - F_{k-1} \mathbf{u}_{k-1})$
- (b) IF G_k is the coarsest grid ($k=K$)
 solve $F_K(\mathbf{U}_K)=F_K$ exactly
 Prolongation correction $V_K = | \mathbf{u}_K^{new} - \mathbf{u}_K^{old} |$ to the next fine grid G_{K-1}
 ELSE
 Solve $F_k(\mathbf{u}_k)=f_k + P_{k+1}^k V_{k+1}$
 Prolongation correction $V_k = | \mathbf{u}_k^{new} - \mathbf{u}_k^{old} |$ to the next fine grid G_{k-1}
- (c) Do (b) until to the finest grid G_1

The Gauss-Seidel method is used as smoothing operator on all grids except on the coarsest grid where the GS-SOR/SUR method is employed to speed-up the solution. The interpolation operators are bi-linear prolongation and half-weighting restriction.

IV. PARALLELISATION AND IMPLEMENTATION

There are two methods to exploit parallelism in multigrid methods [12]. One is straightforward and based on the domain decomposition technique. In this method the multigrid is divided into several smaller sub-grids, where each sub-grid includes all levels from the finest to the coarsest. These sub-grids are then distributed onto several processors and all processors do the same multigrid operations on different sub-grids in parallel. This method is referred to as data-parallel multigrid (DPMG). The other method is to carry out the multigrid operations concurrently by many processors on different grid levels, in which the multigrid is not partitioned but each level of the multigrid is assigned to a processor and several MG operations on different grid levels are done at the same time. The second method may be called operation-parallel multigrid (OPMG). The DPMG method is used in this work.

Fig.1 shows a 3 level multigrid decomposition and sub-domain mapping onto 3 processors. All operations mentioned above, solving, smoothing, computing errors, and interpolations between different level, are local to each processor. At inter-sub-domain boundaries relaxations and interpolations in a sub-domain will

need the boundary data of the adjoining sub-domains. Column dummy points at each side of sub-domains along the boundaries are allocated and communications are required at all grid levels to update the values of the dummy points.

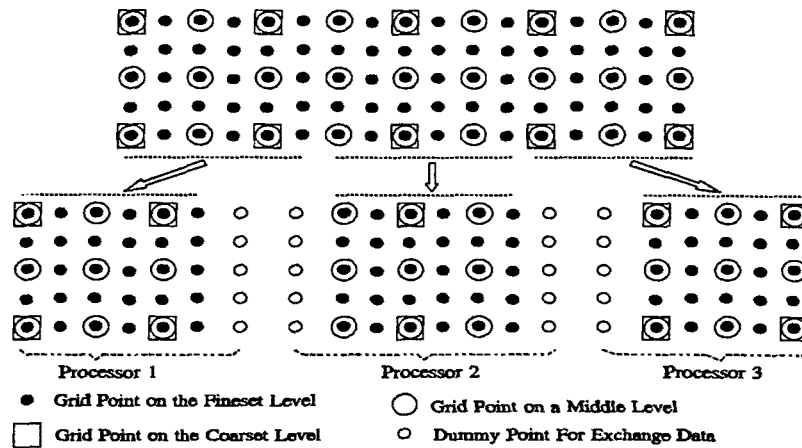


Fig.1 Partitioning and Distributing the Multigrid onto 3 Processors

Thus all levels of the multigrid are partitioned and sub-grids are mapped onto the processors in a way that the locality of the interprocessor communication profits from the locality of the discretisation grid to reduce the communication overheads. A transputer-based machine hosted by a workstation is used in this implementation. The machine includes 16 processors with 2 Mbytes of memory each. These are connected as a ring. The parallel version of multigrid operations includes the following steps,

- (a) *Solving equations (or smoothing errors) on a grid level and updating boundary points of the grid level after every iteration.*
- (b) *Calculating the errors on a grid level.*
- (c) *Prolongation from a coarse grid level to the next fine grid level.*
- (d) *Updating boundary points on the fine grid level.*
- (e) *Restriction from a fine grid level to the next coarse grid level.*
- (f) *Updating boundary points on the coarse grid level.*

It is obvious that the amount of communication is proportional to the number of grid points on the side boundaries of sub-grids for a ring of processors. The amount of the computation, including all multigrid operations, is proportional to the number of grid points in the sub-domains. The parallel overhead is the ratio of the amount of the communication and the amount of the computation. So the overhead becomes larger as the grid becomes coarser.

V. EXPERIMENTAL RESULTS

A GaAs MESFET device with 1 micron channel length has been used in the current work. For a GaAs MESFET device, which is entirely unipolar, only the Poisson equation and the continuity equation for the majority carrier of electrons need to be solved. The bias voltages applied to the contact source and the contact gate are 0V with a built-in voltage of - 0.8V on the gate. The external drain voltage is increased linearly from 0V to 5V over 1 ps and then fixed at 5V. Comparisons are made between the GS-SOR/SUR methods and the MG-FAS method both on a single processor and 16 processors. The simulation time required using the two methods are listed in Table 1. The speedup factors due to the multigrid method and due to the data-parallel algorithm are calculated from the execution time.

It can be seen from the table that the multigrid method is implemented on the transputer-based system without significantly impairing the efficiency of the multigrid. The parallel multigrid still produces a typical parallel speedup of domain decomposition techniques. Since only a few grid points are left on a very coarse grid, communication overhead becomes more significant; the speedup of the parallel multigrid method drops from 7.10 to 4.62. The speedup due only to the data-parallel method also decreases from 13.1 to 8.51 for the same reason. The speedup of both sequential and parallel multigrid is affected by the fact that the solution of a time-step can be started from the solution of the previous time-step as a very good initial guess and so the initial error is normally smaller compared with general initial problem. Even so the overall speedup from the combination of the multigrid method and the domain decomposition method is still over 60.

VI. CONCLUSIONS

It has been demonstrated that it is possible to obtain fast solutions for the coupled semiconductor equations using the multigrid method on a medium-grain parallel machine with distributed memory. Both the parallel efficiency of the underlying data-parallel algorithm and the convergence rate of multigrid method are maintained. By using the parallel multigrid method on a 16 processor machine the simulation time is reduced by more than 60 times compared with the single grid sequential solver with the Gauss-Seidel method coupled with successive over/under-relaxation scheme. The parallel efficiency is expected to improve with an increase in the problem size.

Table 1 Execution Time(in seconds) and Speedup(Sp)

Processor No.	GS-SOR/SUR	MG-FAS	MG-Sp
1	4087	575.6	7.10
16	312.1	67.60	4.62
Data-Para. Sp	13.10	8.51	60.46 [†]

† Overall Speedup

REFERENCES

- [1] A S Shieh, On the Solution of Coupled System of PDE by a Multigrid Method, *IEEE Trans. on Electron Devices*, Vol.ED-32, No.10, pp.2083-2086, 1985
- [2] P W Hemker, A Nonlinear Multigrid Methods for One-Dimensional Semiconductor-Device Simulation - results for the Diode, *J. of Comput. and Appl. Math.*, Vol.30, No.1, pp.117-126, 1990
- [3] P M de Zeeuw, Nonlinear Multigrid Applied to a One-Dimensional Stationary Semiconductor Model, *SIAM J. on Sci. and Stat. Comput.*, Vol.13, No.2, pp.512-530, 1992
- [4] K Motegi and S Watanabe, A Parallel Algorithm for Solving Two Dimensional Device Simulation by Direct Solution Method and Its Evaluation on the AP-1000, *IEICE Trans. Fundamentals*, Vol.E75-A, No.7, pp.920-922, 1992
- [5] J P Darling and I D Mayergoyz, Data Parallel Algorithms for the Numerical Modeling of Semiconductor Devices, *Proc. of the 5th SIAM Conf. on Parallel Processing for Sci. Comput.*, Houston, TX D910325-27, CH.4, pp.394-400, 1992
- [6] D M Webber, E Tomacruz, R Guerrieri, T Toyabe, and A Sangiovanni-Vincentelli, A Massively Parallel Algorithm for Three-Dimensional Device Simulation, *IEEE Trans. on CAD*, vol.10, No.9, pp.1201-1209, 1991
- [7] B Boittiaux, G Goncalves, and M P Haye, A Transputer Network for a Device Simulator, *Microprocessing and Microprogramming*, Vol.35, pp.127-132, 1992
- [8] C S Tsang-Ping, D M Barry, and C M Snowden, Parallel Implementation of A GaAs MESFET Electron-Thermal Simulation on a Transputer-Based System, *In this proceedings*.
- [9] H K Gummel, A Self-Consistent Iterative Scheme for One-Dimensional Steady State Transistor Calculations, *IEEE Trans. on Electron Devices*, Vol.ED-11, pp.455-465, 1969
- [10] M Reiser, Large-Scale Numerical Simulation in Semiconductor Device Modeling, *Comp. Math. Appl. Mech. Eng.*, Vol.1, pp.17-38, 1972
- [11] A Brandt and N Dinar, Multigrid Solution to Elliptic Flow Problems, *Numerical Methods in Partial Differential Equations*, Ed. S Parter, pp.53-147, 1977
- [12] R S Tuminaro, Multigrid Algorithms on Parallel Processing Systems, *PhD Thesis, Stanford University*, 1990