3-D Device Simulation Using Intelligent Solution Method Control

DANIEL C. KERR* and ISAAK D. MAYERGOYZ

Department of Electrical Engineering, University of Maryland, College Park, MD 20742

In this paper, a hybrid solution method is implemented for solving the semiconductor transport equations. The hybrid "local Newton" method consists of a combination of the fixed-point iteration (FPI) and Newton's methods. The FPI technique is nearly ideally suited to solving large, 3-D systems of semiconductor equations on machines of limited computer memory; however, it has certain limitations. This motivates the local Newton method, which coordinates the use of both the FPI and Newton's methods, for convergence faster than either method alone.

Keywords: Newton method, fixed-point iteration method, semiconductor device simulation, three-dimensional

1. INTRODUCTION

Numerical simulation of semiconductor devices requires the solution of a large, nonlinear, coupled system of discrete equations. By formulating and discretizing the model equations and boundary conditions, a system of discrete equations is formed. The number of discrete equations is equal to the product of the number of mesh points and the number of model equations. The solution method should be fast, accurate, and robust. Furthermore, the solution method must be suitable for engineering workstations with limited computer memory.

To introduce the solution methods, the equation and domain coupling is described. The system of discrete equations is coupled together both in equation space and in real space. In equation space, the discretized variables at a node are coupled to the other vari-

ables at the point. In real space, the discretized variables of one equation are coupled to the other variables of the same equation at neighboring points. All the existing solution methods can be classified according to the way they treat this coupling, as shown in Figs. 1 and 2. The Newton method is a fully-coupled method and includes all the coupling in equation and real space. The Gummel method is decoupled in equation space but coupled in real space. The FPI method is decoupled both in equation space and in real space. The point-Newton method treats the equation coupling point-wise in real space. Finally, an interesting method explored here is called **selective coupling**, and combines the advantageous features of several different methods. Each of these methods can be described by using the diagram shown in Fig. 1. The Newton method gathers together the variables from all the equations at all mesh points into one large

* Present address. National Semiconductor Research Laboratory, 2900 Semiconductor Drive, Mail Stop E-100, Santa Clara, CA 95052-8090. Tel: (408) 721-1700. Email: dankerr@ampere.nsc.com

FIGURE 1 Portion of 2-D mesh showing discretized variables of Poisson (solid circles) and current-continuity (hollow circles) equations. In equation space, the discretized variables are coupled to the other variables at each node. In real space, the variables of one equation are coupled to the other variables of the same equation at neighboring points

matrix equation and solves it iteratively. The Gummel method is decoupled in equation, so it gathers up the variables at all mesh points for one equation into several matrix equations and solves them iteratively. The FPI does not use matrices at all. Instead, a single equation is solved and iterations are performed over all mesh points and all equations.

To complete the chart, the point-Newton method collects the variables for all the equations at only one mesh point into a small matrix and iterates over all mesh points. Finally, there is selective coupling, or the local Newton method. What this method does is first analyze the coupling of the points and classify them according to their degree of equation or space coupling. Then, the appropriate solution method is applied. For example, the channel region of a MOS-FET is strongly coupled both in equation and in space. Clearly, these points should be solved using Newton's method. However, points in the substrate are weakly coupled and can be efficiently solved using a decoupled method.

Each of these methods has its strengths and weaknesses for different simulation problems. For the full coupling or Newton's method, the advantage is its superlinear (quadratic) rate of convergence. This rate of convergence is very fast, and no other method can beat it. However, the method is not globally convergent, that is, the Newton method requires a good initial guess in order for successful convergence to occur. To partially overcome this problem, typical

implementations of Newton's method are damped, which hurts the rate of convergence. Newton's method is limited to small meshes by memory requirements.

The advantages of the equation decoupling (or Gummel) method are that the computer-memory requirements are reduced, and the solution time per iteration is reduced. On the other hand, this method requires acceleration techniques in order to be competitive with Newton's method. This method will slow or fail under certain conditions, for example high bias and high current conditions [1].

The FPI method, since it avoids the use of matrices in favor of explicit formulas, requires very little computer memory. This allows the use of very large 3-D meshes on an engineering workstation. The solution method is globally convergent for each equation, that is, the final convergence is not dependent on the initial guess. This means that a simulation of a high bias case can be performed without ramping up the bias, something many Newton solvers require. The algorithm is inherently parallel, which means it can be implemented on a parallel supercomputer with very high degrees of parallel efficiency. Finally, the algorithm is very simple, so it can be sped up with many convergence acceleration tricks. On the other hand, this method suffers the problems of Gummel's method, and in addition, the convergence may slow or fail when the discrete equations are tightly coupled in real space.

In summary, the Newton's method requires very few iterations, for example as few as 10-20 iterations, but each iteration takes a great deal of work. The amount of work per iteration and memory requirements grow strongly with the number of mesh points. Each iteration of the FPI method is very fast, but many iterations are required. Which method will be faster depends on the device, the mesh, and the biasing conditions. In some cases, Newton's method will be faster, while in other cases the FPI will be faster. However, for large, 3-D meshes, the FPI technique can be implemented and run on engineering workstations, while Newton's method cannot. The small memory requirements make the FPI technique ideal for 3-D simulations, and so this method is of primary interest here.

2. THE FIXED-POINT ITERATION TECHNIQUE

The FPI technique is applied to the solution of the semiconductor equations which are discretized by using the box-integration technique on unstructured meshes. Previously, the FPI algorithm was proposed for the solution of the nonlinear Poisson equation [2] and then demonstrated for 2-D MOSFETs [3, 4] using finite-difference meshes. First, the semiconductor equations are decoupled by using Gummel's iterative method [5], wherein each equation will be solved for its associated state variable while the other two variables are considered known and taken from the previous iteration. Then, for each equation in the drift-diffusion (DD) model the discretized flux J_{ij} or D_{ij} can be written as some difference in nodal values of the state variables,

$$C_{ij} = \alpha_{ij}(\eta_j - \eta_i), \quad (1)$$

where η is one of the state variables. Using this form of C_{ij} the discretized equation at node i can be written as [3]

$$\sum_{j \in N_i} \alpha_{ij} A_{ij} \eta_j - \alpha_i A_{ij} \eta_i = f_i + G_i(\eta_i), \quad (2)$$

where the nonlinear terms G_i are monotonically increasing, and the coefficients have the following properties,

$$\alpha_{ij} \geq 0, \alpha_i > 0, \text{ and } \alpha_i \geq \sum_{j \in N_i} \alpha_{ij}. \quad (3)$$

By using these properties, the fixed-point iterations, described below, can be shown to be globally convergent, *i.e.*, these iterations will converge for any initial guess [3]. This is an important property for practical, routinely convergent computations.

The FPI algorithm consists of the set of nested loops shown in Fig. 2; these loops can be called the Gummel loop, the mesh loop, and the Newton loop. Outmost, the Gummel loop is responsible for decoupling the equations and linearizing the current-continuity equations. This loop stops when the maximum change in state variables during each Gummel cycle is less than the convergence criteria. During each step of the Gummel loop, the mesh loop sweeps

FIGURE 2 Flowchart of the FPI algorithm, which consists of three nested loops. The Gummel and mesh loops are shown

over all the mesh points, and stops when the maximum change in the state variable for the equation satisfies its convergence criterion, which is typically more stringent than the criteria in the Gummel-loop level. Innermost, during each step of the mesh loop, eq. (2) is solved, and if the equation is nonlinear, the Newton method is used, requiring its own loop.

Since this iterative method avoids the use of matrices in favor of explicit formulas, the method only requires a small amount of computer memory. In addition, each mesh point in the mesh loop could be mapped to its own processor in a massively parallel computer for the rapid solution of large-scale problems [6, 7]. Finally, owing to the simplicity of the algorithm, the method is easy to code, which is an important consideration when manipulating complex data structures. The performance of the basic FPI is enhanced by applying several convergence-acceleration techniques [8].

The rate of convergence will be slow whenever the equation or space coupling becomes very strong. In

pathological cases, the convergence can be slowed to imperceptible levels. For the equation coupling, this limitation is the same as other Gummel methods [1]. The convergence of the FPI technique is slow for meshes with very highly nonuniform mesh spacing. The coupling coefficients at a node with erratic mesh spacing will be unbalanced, and the solution cannot propagate through this connection. If the erratic point is isolated, the solution can propagate around the connection. However, in cases where a line of connections is erratically spaced, as in a pair of close lines in a tensor-product mesh, the convergence rate can be drastically reduced.

This limitation is a serious problem. Not all meshes can be used successfully. This limitation can be worked around by designing meshes which avoid abrupt changes of mesh density. In practice, this has the effect of forcing meshes with more mesh points. In order to overcome this limitation, the local Newton technique is introduced.

3. LOCAL NEWTON TECHNIQUE

The local Newton technique solves the discretized equations by the coordinated application of several solution techniques, especially the Newton method and the FPI method. The limitations of each method can be overcome by intelligently applying the best solution method to a given subset of equations. The mesh points are decomposed into groups based on their coupling in equation space or real space. The best solution method is applied to each group separately.

The rate of convergence is controlled by a small set of mesh points. In the fixed-point iterations, those points which are tightly coupled converge slowly. By treating those points with a solution method which includes the coupling of the equations, the rate of convergence can be improved.

The new solution algorithm is illustrated in Fig. 3. This method first analyzes the coupling of the discretized variables and classifies them according to their degree of equation or real-space coupling. After an initial guess is calculated, the degree of equation-

FIGURE 3 Flowchart of the local Newton technique. The discretized variables are partitioned into the appropriate solver based on the degree of equation and space coupling

space and real-space coupling is calculated, and the mesh points are partitioned into groups. The best available solver is applied to each group, and the iteration is over the groups. This approach avoids the weaknesses of each solver by matching its strengths to a narrowly-tailored problem. In this section, first Newton's method is summarized and the method of solving the resulting linear equations is described. Finally, the method of partitioning the discretized equations is introduced.

3.1 Newton's Method

Newton's method is a general numerical technique for solving nonlinear equations. The main idea of Newton's method, or the Newton-Raphson method for finite-dimensional spaces, is successive linearization. At a given initial guess, a linear approximation of the nonlinear equations is constructed and solved. Iterations proceed until the size of the updates of the variables falls below a predetermined threshold. In general, to solve some nonlinear system of equations $\vec{\mathcal{T}}(\vec{x}) = 0$, the following equations are solved:

$$J(\vec{x}^k) \delta \vec{x}^k = -\vec{F}(\vec{x}^k), \quad \vec{x}^{k+1} = \vec{x}^k + \delta \vec{x}^k \quad (4)$$

where the Jacobian $\mathcal{J}$ is defined by $\mathcal{J}(\vec{x}) = \left\{ \frac{\partial F_i}{\partial x_j} \right\}$

In this case, the equations $\vec{\mathcal{T}}(\vec{x})$ are the semiconductor equations which are currently being solved. The Jacobian matrix is assembled point by point for each function in $\mathcal{T}$. The resulting linear equations are solved by using standard sparse matrix methods. For efficiency, the size of the local Newton groups are restricted to less than several hundred nodes. With matrices of this small size, sparse Gaussian elimination can be used.

3.2 Partitioning of Discretized Equations

Several methods are used to partition the nodes into groups. The degree of equation coupling is measured by deviation from charge neutrality or inversion. For example, in a MOSFET, this assigns the channel

region to equation-space coupled solvers. The substrate points are quasi-neutral, so they are assigned to equation-space decoupled solvers. The degree of space coupling is measured by mesh nonuniformity and by strong current flows. To some extent, the coupling between mesh points can be predicted *a priori* on the basis of their physical location within the device. For example, in a MOSFET, the mesh points in the channel region will be strongly coupled due to the strong current flow. Thus, the channel region will be solved using Newton's method, the depletion regions surrounding the source and drain will be solved using Gummel's method, and the rest by the FPI method. This efficient and intelligent partitioning saves CPU time compared to any single method alone.

This selective coupling approach has been integrated into the general 3-D device simulation program SIMASTER. When used for simulations for which the FPI method converges well, the local Newton method yields speed improvements of between 30-50 %. However, the local Newton method also can be used for highly nonuniform meshes on which the fixed-point iterations fail to converge.

	COUPLED	(<i>Real Space</i>)	DECOUPLED
COUPLED	Newton		Point-Newton
↕ Equation Space		Selective coupling	
DECOUPLED	Gummel		Fixed-point

TABLE I The classification of the numerical solution techniques according to equation-space and real-space coupling

4. SUMMARY

In this paper, the numerical methods of solving the semiconductor transport models were introduced. The methods considered were the FPI method and a hybrid "local Newton" method consisting of a combination of the fixed-point iteration and Newton's method. The FPI technique is nearly ideally suited to solving the semiconductor equations on machines of

limited computer memory. The local Newton method coordinates the use of both the FPI and Newton's method, for convergence faster than either method alone.

References

- [1] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*. New York: Springer-Verlag, 1984.
- [2] I.D. Mayergoyz, "Solution of the nonlinear Poisson equation of semiconductor device theory," *J. Appl. Phys.*, vol. 59, no.1, pp. 195-199, 1986.
- [3] C. E. Korman and I. D. Mayergoyz, "A globally convergent algorithm for the solution of the steady-state semiconductor device equations," *J. Appl. Phys.*, vol. 68, no. 3, pp. 1324-1334, 1990.
- [4] M. Kurata and S. Nakamura, "An explicit method of numerical integration for the complete set of semiconductor device equations," *IEEE Trans. Computer-Aided Design*, vol.11, no. 8, pp. 1013-1022, 1992.
- [5] K. K. Gummel, "A self-consistent iterative scheme for one-dimensional steady state transistor calculation," *IEEE Trans. Electron Devices*, vol. 11, no. 10, pp. 455-465, 1964.
- [6] J. P. Darling and I. D. Mayergoyz, "Parallel algorithm for the solution of nonlinear Poisson equation of semiconductor device theory and its implementation on the MPP," *J. Parallel and Distributed Computing*, vol. 8, pp. 161-168, 1990.
- [7] G.-C. Tai, C. E. Korman, and I. D. Mayergoyz, "A parallel-in-time method for the transient simulation of SOI devices with drain current overshoots," *IEEE Trans. Computer-Aided Design*, vol. 13, no. 8, pp. 1035-1044, 1994.
- [8] D. C. Kerr, *Three-Dimensional Drift-Diffusion and Hydrodynamic Simulation of Semiconductor Devices*. PhD thesis, The University of Maryland, College Park, December 1995.