Monotone Iterative Method and Adaptive Finite Volume Method for Parallel Numerical Simulation of Submicron MOSFET Devices

YIMING LI[1,2,*], CHENG-KAI CHEN[2], and PU CHEN[2]
[1]National Nano Device Laboratories, 1001 Ta Hsueh Rd., Hsinchu city, Hsinchu 300, TAIWAN
[2]National Chiao Tung University, 1001 Ta Hsueh Rd., Hsinchu city, Hsinchu 300, TAIWAN
[*]Corresponding address: P.O. Box 25-178, Hsinchu city, Hsinchu 300, TAIWAN

Abstract: - In this paper, we apply our proposed early parallel adaptive computing methodology for numerical solution of semiconductor device equations with triangular meshing technique. This novel simulation based on adaptive triangular mesh, finite volume, monotone iterative, and a posteriori error estimation methods, is developed and successfully implemented on a Linux-cluster with message passing interface (MPI) library. Parallel adaptive computing with triangular mesh has its flexibility to simulate multidimensional semiconductor devices with highly complicated geometry. Our approach fully exploits the inherent parallelism of the triangular mesh finite volume as well as monotone iterative methods for semiconductor drift diffusion equations on a Linux-cluster parallel computing system. Parallel simulation results demonstrate an excellent speedup with respect to the number of processors. Benchmarks and numerical results for a submicron N-MOSFET device are also presented to show the robustness and efficiency of the method.

Key-Words: - Monotone Iterative Method, Finite Volume Method, Adaptive Refinement, Triangular Mesh, Cluster Computing, Drift Diffusion Equations, MOSFET

1 Introduction

The main task of a semiconductor device simulator is to analyze the intrinsic and extrinsic electrical behavior of the most basic device's structures at a very fundamental physical level. In recent years, device simulation has been becoming a very important tool in the development of new devices and fabrication technologies (see [1] and references therein). According to the Maxwell's equations and carriers' conservation in transport, the drift diffusion (DD) equations that consist of Poisson equation and electron-hole current continuity equation have been successfully applied to describe the carriers' transport phenomena [2-8, 10, 11]. When the device scale is down to deep submicron or nanometer regions, scientific computing becomes currently one of the major approaches to solve the device equations efficiently in semiconductor device simulation. Parallelization of such numerical simulations with adaptive triangular mesh is still a very complex task.

In this paper, combining with triangular meshing technique we successfully extend our proposed parallel adaptive simulation methodology [4-8] to solve semiconductor device drift diffusion equations in C^{++} language. With the developed triangular mesh device simulator, one can analyze highly complicated and irregular multi-dimensional device with more precise estimation in its simulation domain. Our numerical results for a two-dimensional submicron N-MOSFET device are demonstrated to show the robustness and efficiency of the method.

Considering an adaptive triangular unstructured mesh and finite volume discretization scheme [9], a set of device DD equations, three partial differential equations (PDE), is discretized firstly and then directly solved by means of the monotone iterative method instead of the conventional Newton's iteration method. The monotone iterative method is a constructive alternative for numerical solutions of PDEs [4-8]. Compared with Newton's iterative method, major features of the present method for triangular mesh device simulation are as follows: (i) it converges globally with any arbitrary initial guess for submicron devices under various bias conditions, (ii) its implementation is much easier than Newton's iterative method, and (iii) it is inherently ready for parallelization.

To establish a physical based and efficient adaptive refinement scheme, we note a fact firstly that for most practical submicron devices (for example: MOSFET, HBT, SOI, and PN Diode [2]) the physical quantities, such as electrostatic potential and electron densities exhibit extreme variations within a quite small region, particularly in the inversion layer, depletion layer, and neighborhood of p-n junctions. The presence of layers implies that a local adaptive mesh refinement strategy for unstructured triangular mesh would capture the solution gradients in a very efficient manner. In this work, with this physical observation and a posteriori error estimation, an efficient adaptive triangular mesh refinement algorithm is now developed and successfully tested and implemented on our device simulator. The simulation starts from a simple initial triangular mesh, and automatically solve problem and refine mesh iteratively. The iteration will be terminated when a specified error criterion is reached. Numerical results for a typical N-MOSFET device are demonstrated to show the robustness and efficiency of the method. Our achieved parallel benchmarks, such as speedup, load balancing, and efficiency also show the good performance of the method through the work.

This paper is organized as follows. In Sec. 2, we introduce the semiconductor device drift diffusion model and associated physical models. In Sec. 3, we present the triangular mesh adaptive finite volume scheme and state monotone iterative method. Sec. 4, sketches the parallel computing of the triangular mesh deice simulator. In Sec. 5, simulation results for a submicron N-MOSFET device are presented to demonstrate the robustness and parallel efficiency of the method. Sec. 6 draws the conclusions.

2 Semiconductor Device Equations

The DD model is the first model for semiconductor device simulation [2, 3, 10, 11]. It was derived from Maxwell's equation as well as charges' conservation law and has been successfully applied to study device transport behavior, in the past decades [2-8, 10, 11]. It assumes local isothermal conditions and is still widely employed in semiconductor device design. A set of the DD equation is as follows:

$$\Delta\phi = \frac{q}{\varepsilon_S}(n - p + D), \tag{1}$$

$$\frac{1}{q}\nabla \cdot J_n = R(n,p), \tag{2}$$

$$\frac{1}{q}\nabla \cdot J_p = -R(n,p), \tag{3}$$

$$J_n = -q\mu_n n\nabla\phi + qD_n\nabla n, \tag{4}$$

$$J_p = -q\mu_p p\nabla\phi - qD_p\nabla p. \tag{5}$$

In above equations (1)-(5), the Eq. (1) derived from Maxwell's equation is so-called the Poisson equation. The Eqs. (2) and (3) derived from the charge conservation law are the electron and hole continuity equations. The Eqs. (4) and (5) are electrons and holes current equations, respectively. The unknown $\phi = \phi(x,y)$, in Eq. (1), to be solved is the electrostatic potential, n and p are electrons and holes concentrations. The function $D = -(N_D^+ - N_A^-)$, in Eq. (1), is the specified ionized net doping profile and is a spatially-dependent given function, and $R = R(n,p)$ is the recombination rate for electrons and holes. In this study, the R is assumed to be the Shockley-Read-Hall recombination process [2, 3]:

$$R(n,p) = \frac{(np - n_i^2)}{t_p(n + n_i) + t_n(p + n_i)}, \tag{6}$$

where t_n and t_p are the electron and hole lifetimes, respectively. The quantity $q = 1.60218 \times 10^{-19} C$ is the elementary charge, $\varepsilon_S = 11.9\varepsilon_0$ is silicon permittivity. The N_D^+, and N_A^- are ionized donor and acceptor impurities, and $\varepsilon_0 = 8.85418 \times 10^{-14} F/cm$ is the permittivity in vacuum. The D_n, D_p, μ_n, and μ_p are electron and hole diffusion coefficients and mobility functions, respectively. In general, the mobility functions in DD model strongly depend on electric field, doping concentration, and electric current density [2, 3, 10, 11].

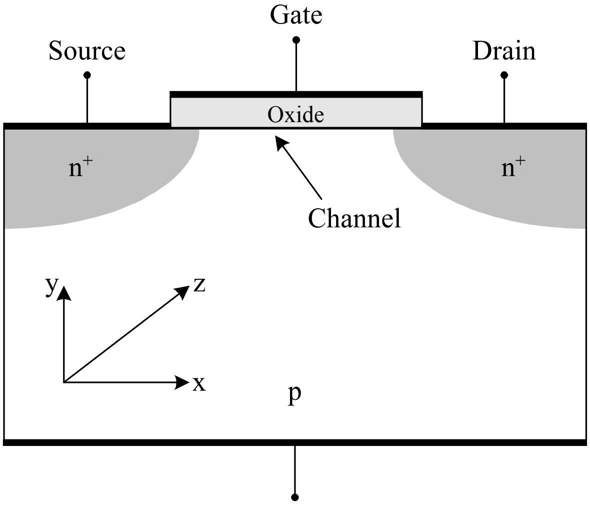

Fig. 1. A two-dimensional domain of a submicron N-MOSFET device.

As shown in Fig. 1, the DD equations (1)-(5) are subject to mixed type boundary conditions in a two-dimensional simulation domain. On the left and right sides, the homogeneous Neumann type boundary condition is considered. On the Source, Gate, Drain and Substrate contacts, the Dirichlet type boundary condition is applied.

In conventional device DD model simulation, various numerical methods have been developed for the approximation solution of the system (1)-(5) with the primal state variables (ϕ, n, p) or (ϕ, φ_n, φ_p) [3] and have their advantages. Based on the Boltzmann statistics [2, 3], we solve the transferred DD equations (1)-(5) in terms of (ϕ, u, v) [2, 3]. This formulation of the model is a set of self-adjoint PDEs, and is very favorable for the triangular finite volume method and monotone iterative method. The Eqs. (1)-(5), in this work, are solved with Gummel's decoupled algorithm, adaptive finite volume method, and monotone iterative method on a cluster parallel computing system. The Eqs. (1)-(5) together with applied finite volume and monotone iterative methods lead to a robust convergence property in computer simulation; hence, the parallization of this solution approach has also been developed and successfully implemented. In the next section, we describe the adaptive solution steps used in this simulation.

3 Adaptive Computational Methods

The Gummel's decoupling algorithm, adaptive finite volume method, and monotone iterative method will be presented in this section.

Applying the Gummel's decoupling method to decouple three DD equations, we can prove mathematically the nonlinear system rising from the finite volume discretization [9] for the individually decoupled PDE on a triangular mesh has at most one solution. In addition, it also can be shown that the solution sequences constructing from monotone iterative formula converge to the solution of the nonlinear system monotonically.

3.1 Gummel's Decoupling Scheme

One of efficient solution methods in semiconductor device simulation is often used Gummel's decoupled method to decoupled these tree coupled PDEs and then solve each PDE iteratively. In this work, we solve the decoupled PDE with our proposed early adaptive computing procedure [4-8]. The basic idea of well-known Gummel's decoupled [10] method is that the device equations are solved sequentially (see Fig. 2). In the DD model, Poisson's equation is solved for $\phi^{(g+1)}$ given the previous states $u^{(g)}$ and $v^{(g)}$. The electron current continuity equation is solved for $u^{(g+1)}$ given $\phi^{(g)}$ and $v^{(g)}$. The hole current continuity equation is solved for $v^{(g+1)}$ given $\phi^{(g)}$ and $u^{(g)}$. The superscript index g denotes the Gummel's iteration loops. Each decoupled PDE is solved with our adaptive computing algorithm. We describe this solution method in Sec. 3.2.

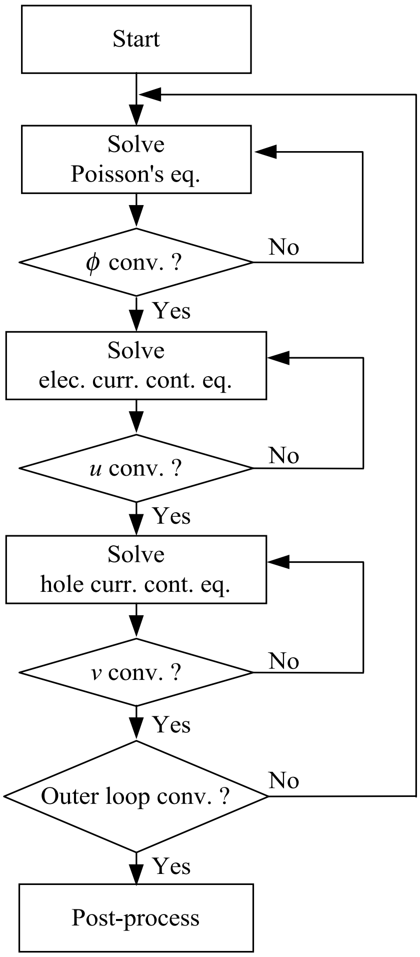

Fig. 2. A flowchart for Gummel's decoupling algorithm in semiconductor device simulation.

3.2 Adaptive Finite Volume Algorithm

The theoretical concept of the adaptive finite volume method relies on the estimations of the solution gradient and variation of carrier lateral current density along the device channel surface. A posteriori error estimation provides not only a global assessment of the quality of numerical solutions but also a set of local error indicators to incorporate with refinement strategies. This physical based error estimation and error indicators applied here is not restricted to any particular types of mesh structure. In this work, we use triangular mesh refinement to simulate more complicated device geometry. The data structure of the triangular mesh is designed with hierarchical, and is suitable for the implementation of the adaptive algorithm by using the object-oriented programming concepts.

As shown in Fig. 3, given a decoupled PDE in semiconductor device DD model, we first partition the solution domain into a set of finite volumes. The PDE is then approximated by finite volume method. For the electron and hole current continuity equations, we also apply the Scharfetter-Gummel exponential fitting [3] to locate the sharp variation of the solutions. With sparse matrix technique, we construct the global matrix and the corresponding vector. The monotone iterative [4-8] solver is directly applied to solve the system of nonlinear equations.

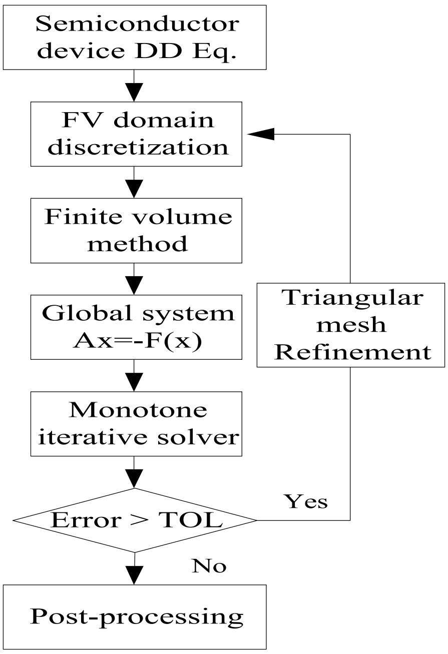

Fig. 3. Adaptive finite volume solution algorithm.

Once an approximate solution is computed, an a posteriori error analysis is performed to assess the quality of the approximate solution. The error analysis will produce error indicators and an error estimator. If the estimator is less than a preset error tolerance (TOL), the adaptive process will be terminated and the approximate solution can be post-processed for further physical analysis. Otherwise, a refinement scheme will be employed to refine each of the current elementst. A finer partition of the domain is thus created and a new solution procedure is repeated. Using the maximum gradient of electrostatic potential ϕ in equation (1) and/or the variation of current density J_n in equation (4) as error estimation the discretization mesh is adaptively generated. A refinement scheme is employed to refine each of the current elements depending on the magnitude of the error indicator for that element.

3.3 Monotone Iterative Method

The classical Gummel's decoupling and solution method for semiconductor device DD model has some steps: (I) Scharfetter-Gummel exponential fitting [11] for the electron and hole current continuity equations, (II) three inner loops of Newton's iteration for each unknown function, and (III) an outer loop for all unknown functions [3, 10].

Our method replaces Newton's iteration by the monotone iteration in which the corresponding discrete system is of the following form [4-8]:

$$(D+\lambda I)Z^{(m+1)} = (L+U)^{(m)} - F(Z^{(m)}) + \lambda I Z^{(m)}, \tag{7}$$

where Z is the unknown vector, F is the nonlinear vector form, and D, L, U, and I are diagonal, lower triangular, upper triangular, and identity matrices, respectively. The monotone iterative parameter λ is determined node-by-node depending on the device structure, doping concentration, bias condition, and nonlinear property of each decoupled equation.

The monotone iterative method applies here for semiconductor device simulation [4-8] is a global method in the sense that it does not involve any Jacobian matrix. However, the Newton's iterative method not only has Jacobian matrix but also inherently requires a sufficiently accurate initial guess to begin with the solutions. Note that the Eq. (7) is highly parallel; consequently, the monotone iterative method is very cost effective in terms of both computational time and storage memory.

4 Parallel Simulation Algorithm

After the calculation of error estimation and error indicators and checking adaptation stages, the next step is to determine whether workload balance still exists or not for all processors. When a refined tree structure is created, the number of processors for next computing will be dynamically assigned and allocated following the total number of nodes firstly. Then a geometric dynamic graph partitioning method in *x*- or *y*-direction, as shown in Fig. 4, is applied to partition the number of nodes to each processor.

A computational procedure for parallel domain decomposition is as follows: (Di) Initialize the MPI environment and configuration parameters. (Dii) Based on unstructured triangular meshing rule, a tree structure and mesh are created. (Diii) Count number of nodes and apply a *dynamic partition algorithm* to determinate number of processors in the simulation. All nodes are numbered, besides that the boundary and critical points are identified. (Div) All assigned jobs are solved with equation (7). The computed data communicates by the MPI protocol. (Dv) Do convergence test for all elements and run the adaptive refinement for those needed elements. (Dvi) Repeats steps (Diii)-(Dv) until the error of all elements is less than a specified error bound. (Dvii) Host processor collects data and stops the MPI environment.

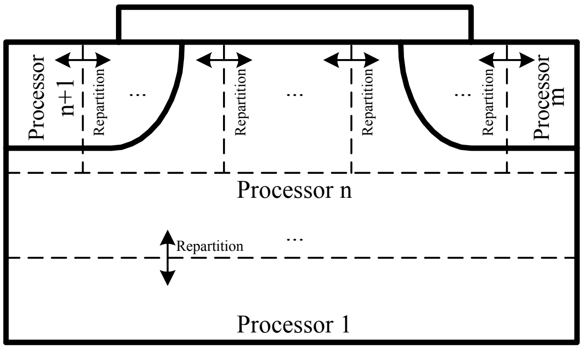

Fig. 4. An illustration of parallel dynamic partition of domain decomposition for a two-dimensional N-MOSFET.

The load balancing *dynamic partition algorithm* in step (Diii) is outlined as follows: (Pa) Count the number of total nodes. (Pb) Find out the optimal number of processors. (Pc) Calculate how many nodes should be assigned to each processor. (Pd) Along *x*- or *y*-direction in device domain, search (from left to right and bottom to top) and assign nodes to these processors sequentially. Repeats this step until all nodes have been assigned. (Pe) In the neighborhood of p-n junction, one may have to change search path for obtaining a better load-balancing performance.

5 Simulation Results and Discussions

We now present some typical simulation results; the first example is a 0.25μm N-MOSFET device with the gate oxide thickness 7.0 nm. The device has elliptical $5*10^{20}$ cm^{-3} Gaussian doping profiles in source and drain regions, 10^{16} cm^{-3} in the p-substrate region, and a shallow $5*10^{17}$ cm^{-3} implantation in channel surface. The junction depth is 0.13μm and the lateral diffusion under gate is 0.09μm. Figs. 5, 6, and 7 show the initial and finial refined mesh, potential and electron concentration, respectively. In this example, the initial mesh has 162 elements and after about 10 refinement levels the finial mesh has 2644 elements for a 5V_T (V_T = 0.0259V) mesh refinement criterion. The stopping error bound between any two successive iterations is less than $10^{-5}V_T$ for all unknowns.

Our next example is designed to demonstrate the robustness of the simulator for the same test device. As shown in the Fig. 8, the global convergence behavior in the Gummel's iteration loop is confirmed, and the maximum norm error for both of the ϕ and u here are less than $10^{-5}V_T$ after 20 iterations. Our monotone iteration loop for a specified Gummel's iteration loop also has similar excellent convergence property. In addition, as shown in Fig. 9, by setting a more strict refinement error 1V_T for all elements, we find the simulator has a very good efficiency in refinement levels and error control.

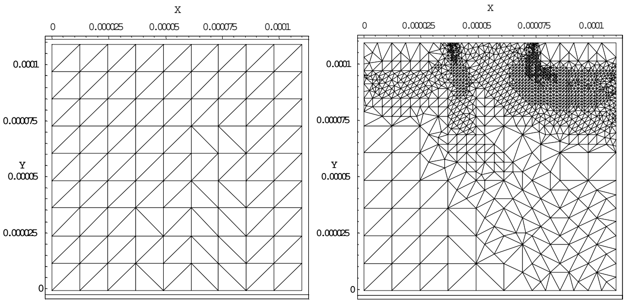

Fig. 5 The left figure is the initial mesh and the right one is the refined mesh. A submicron N-MOSFET simulation at $V_D = V_G = 1.0$ V.

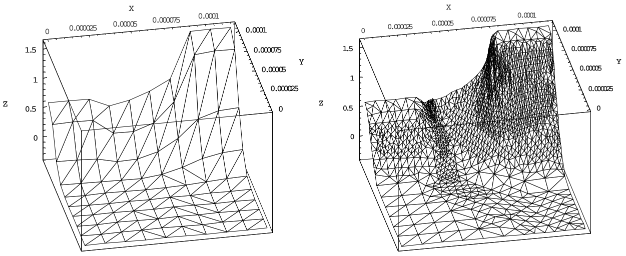

Fig. 6 The initial (left) and final (right) computed electrostatic potential, respectively.

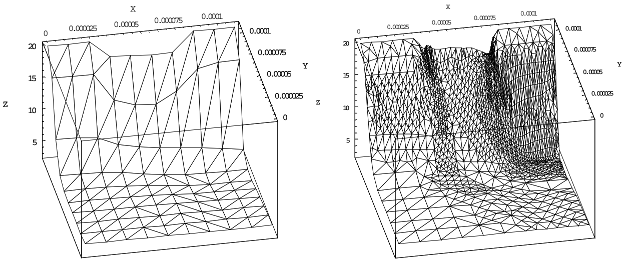

Fig. 7 The initial (left) and final (right) computed electron concentration, respectively.

Furthermore, performance of the parallel adaptive simulation approach for submicron N-MOSFET device simulation is also presented in this work. The device structure used for this test is the same with the first example and the refinement criterion is setting now to be 0.2V_T in this simulation. A good speedup in parallel time can be observed in Table 1. The parallel efficiency for about 1,600,000 refined nodes is over 70%. The superior scalability of the parallel processing is mainly due to the nature of the monotone iterative method. A dynamic load balancing ~ 8% for the same refined nodes is also obtained successfully in this work.

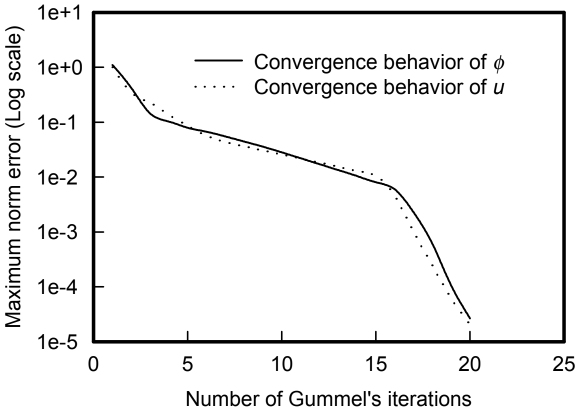

Fig. 8 Convergence behavior for unknowns ϕ and u in the Gummel's iteration loop.

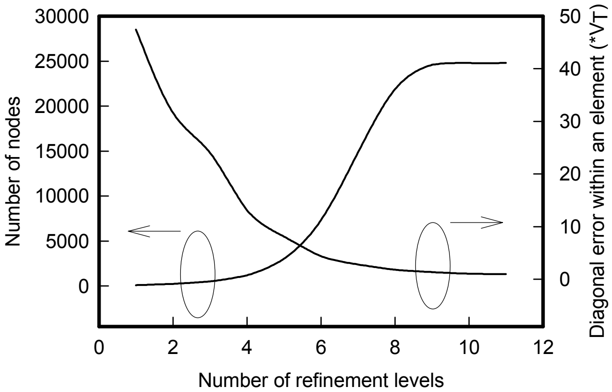

Fig. 9 Number of nodes and maximum diagonal error (within an element) versus number of refinement levels.

Table 1. Achieved parallel performance for the parallel adaptive triangular mesh finite volume simulation on a 8-processors Linux-cluster using MPI library.

Number of processors	1	2	4	8
Parallel time (Sec.)	239,241	129,667	70,067	41,417
Speedup	--	1.85	3.41	5.78
Efficiency	--	92.25%	85.36%	72.2%

6 Conclusions

In this paper, we have successfully generalized our proposed early parallel adaptive computing methodology with triangular meshing technique in submicron MOSFET device simulation. This parallel simulation mainly relies on adaptive triangular mesh, finite volume, and monotone iterative methods. Parallel adaptive computing with triangular mesh has its flexible to simulate more realistic semiconductor devices with highly complicated and irregular geometry in 2D or 3D. Numerical results and benchmarks for a submicron N-MOSFET device are also presented to show the robustness, efficiency, and parallel performance of the method.

References:

[1] R. W. Dutton, A. J. Strojwas, "Perspectives on Technology and Technology- Driven CAD.," *IEEE Trans. CAD*, Vol. 19, No. 2, 2000, pp. 1544-1560.

[2] S. M. Sze, Physics of Semiconductor Devices, 2nd Ed., Wiley-Interscience, New York, 1981.

[3] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*, Springer-Verlag, Wein-New York, 1984.

[4] Yiming Li, et al., "*Adaptive finite volume simulation of semiconductor devices on cluster architecture*," in "Recent Advances in Applied and Theoretical Mathematics" Edited by N. Mastorakis, WSES Press, Dec. 2000, pp. 107-113.

[5] Yiming Li, et al., "A New Parallel Adaptive Finite Volume Method for the Numerical Simulation of Semiconductor Devices," *CCP2000 Adv. Prog. Tech. Digest*, 2000 p. 138.

[6] Yiming Li, et al., "*Monotone Iterative Method for Parallel Numerical Solution of 3D Semiconductor Poisson Equation*," in "Advances In Scientific Computing, Computational Intelligence and Applications" Edited by N. Mastorakis, et al., WSES press, 2001, pp. 54-59.

[7] Yiming Li, et al., "A Novel Approach for the Two-Dimensional Simulation of Submicron MOSFET's Using Monotone Iterative Method," IEEE *Proc. Int. Symp. VLSI-TSA*, 1999, pp. 27-30.

[8] Yiming Li, et al., "A Domain Partition Approach to Parallel Adaptive Simulation of Dynamic Threshold Voltage MOSFET," *Abst. Book CCP 2001*, Aachen, Germany, 2001, p. O38.

[9] R. S. Varga, *Matrix Iterative Analysis*, Springer, New York, 2000.

[10] H. K. Gummel, "A self-Consistent Iterative Scheme for One-Dimensional Steady State Transistor Calculations," *IEEE Trans. Elec. Dev.*, Vol. ED-11, 1964, pp. 455-465.

[11] D. L. Scharfetter, H. K. Gummel, "Large-Signal Analysis of a Silicon Read Diode Oscillator," *IEEE Trans. Elec. Dev.*, Vol. ED-16, pp. 66-77, 1969.