

An Efficient Multigrid Poisson Solver for Device Simulations

Marco Saraniti, Achim Rein, Günther Zandler, Peter Vogl, and Paolo Lugli

Abstract—The aim of this paper is to show that the multigrid approach can provide an efficient two-dimensional Poisson solver used in the analysis of realistic semiconductor devices based on particle simulators. Our robust implementation of the multigrid method is faster by one or two orders of magnitudes than standard successive over-relaxation solvers and is capable, at the same time, of efficiently handling highly inhomogeneous grids and irregular boundary conditions relevant for realistic devices. All essential parts of the algorithm, such as coarsening, prolongation, restriction, and relaxation, have been adapted and optimized to deal with these complex geometries and large variations in the charge density. In particular, a new variant of the Gauß-Seidel-type relaxation scheme is introduced that is particularly suited for grids that lack globally dominant directions. As an example, the multigrid Poisson solver has been applied to two different electronic devices, a GaAs High Electron Mobility Transistor and a Si Metal Oxide Semiconductor Field Effect Transistor.

I. INTRODUCTION

THE MOST accurate methods to date for predicting the carrier dynamics in modern semiconductor devices are particle simulations such as the Monte Carlo [1] or the cellular automaton technique [2]. While these tools are elaborate and remarkably accurate, they have the drawback of being very slow. This aspect is inherent to a history-driven approach that follows the time evolution of a relevant sample of the charge carrier population. The common aspect of all particle simulations is the combined self-consistent solution of Boltzmann's transport equation (BTE) and Poisson's equation. The BTE is solved statistically via a stochastic procedure that tracks the phase-space distribution of the charge carriers in time. At discrete time steps, the resulting charge density is then used as input to the Poisson solver, which returns the electric potential that accelerates the carriers. At each step, the Poisson solver can access the previously computed potential as initial guess. Because of the highly nonlinear carrier dynamics in modern devices, it is necessary to refresh the electric potential very frequently, typically on a time-scale of femtoseconds [3].

Considerable effort has been devoted to speed up the simulation process, either by modifying the basic algorithms

or by a more efficient implementation of the physical models [4]. Overall, the computer time required by the traditional finite difference Poisson solvers might be comparable or even higher than the time needed to solve the BTE in a standard, nonparabolic, Monte Carlo approach when the simulated device exhibits complicated geometry and boundary conditions. In the cellular automaton method, these Poisson solvers even dominate the total numerical effort [5]. Therefore, the development of fast and highly efficient Poisson solvers is a key issue for particle simulators.

Several approaches are commonly used for solving Poisson's equation in the context of device simulations. In drift-diffusion methods [6], where only the lowest moments of the BTE are taken into account, finite-element algorithms are often employed [7]. In such context, multigrid techniques have been developed for the solution of all basic transport equations [8], and references therein. Approaches that include higher moments of the BTE or solve the full BTE, such as hydrodynamic or Monte Carlo methods, mostly use finite-difference schemes. Among the latter, the available schemes for Poisson's equation are: 1) mesh relaxation methods such as Gauß-Seidel, successive over-relaxation, or alternating-direction implicit methods; 2) matrix schemes such as the conjugate gradient technique; and 3) rapid elliptic solvers such as Fourier analysis and/or cyclic reduction [9]. To the best of our knowledge, multigrid approaches have not yet been used together with physical simulators for realistic semiconductor devices.

In this paper, we develop a highly efficient two-dimensional Poisson solver that allows us flexible handling of geometrically highly complex boundary conditions (such as irregular contact areas, ultrathin oxide layers, heterointerfaces, and surface charges) and of the large variation of the carrier density in different regions of a device. This is achieved by an optimal adaptation of the multigrid method [10] to device geometries and boundary conditions. For realistic device geometries, the present solver is computationally faster by one or two orders of magnitudes than standard iterative successive over-relaxation (SOR) solvers [11] and, at the same time, is capable of efficiently handling highly inhomogeneous grids and irregular boundary conditions.

This paper is organized as follows. In Section II we first briefly review the basic steps in the multigrid method applied to a finite difference representation of Poisson's equation. In Section III, we present the detailed ingredients of the optimized multigrid algorithm for device simulations. Detailed comparisons with other techniques and concrete applications

Manuscript received December 12, 1994; revised May 31, 1995 and November 20, 1995. This work was supported in part by SIEMENS, by the Deutsche Forschungsgemeinschaft (SFB 384), and by the Italian Research Council (CNR), the project "Sistemi Informatiche Calcolo Parallelo; sottoprogetto Calcolo Scientifico per Grandi Sistemi." This paper was recommended by Associate Editor S. Duvall.

M. Saraniti, A. Rein, G. Zandler, and P. Vogl are with the Physik Department and Walter Schottky Institut, Technische Universität, Munich, Germany.

P. Lugli is with the Dipartimento di Ingegneria Elettronica, Università di Roma Tor Vergata, I-00133 Italy.

Publisher Item Identifier S 0278-0070(96)01842-8.

are presented in Section IV. In particular, we study two typical field effect transistors, namely the silicon MOSFET and the gallium arsenide HEMT [12]. Final remarks and conclusions are given in Section V.

II. THE MULTIGRID METHOD

In this section, the basic concepts of the multigrid method are briefly explained, focusing on the aspects that are relevant for device simulations.

The multigrid technique is a well-established approach to solving differential equations both in a direct and iterative way [10]. We will concentrate on the two-dimensional Poisson's equation, $\nabla\phi = f(x, y)$. By employing finite differencing on a set of grid points that we denote by Ω_n , this elliptic differential equation transforms into an algebraic matrix equation of the form

$$Au = f \quad (1)$$

where the vector u denotes the solution, the matrix A represents the Laplace operator, and f is the forcing function.

It is probably worthwhile to mention that the use of particle simulations for the carrier dynamics requires the use of fixed rectangular grids to allow efficient particle tracking in real space [13]. Coupling the particle-tracking algorithms to adaptive or nonrectangular grids would imply prohibitive computational times for the simulation. This is not the case, for instance, when a drift-diffusion approach is followed [8].

In order to explain the multigrid method, it is useful to recall the basic steps in any standard stationary linear iterative method [11] first. In such a framework, a sequence of approximations $v^0, v^1, \dots, v^n, \dots$ of u is constructed that converge to u [11]. Let v^i be an approximation to u after the i th iteration. Since the exact solution u of (1) is unknown, one may define the residual

$$r^i = f - Av^i. \quad (2)$$

as a computable measure of the deviation of v^i from u . Next, we define the *algebraic error* e^i of the approximation v^i by

$$e^i = u - v^i. \quad (3)$$

Subtracting (2) from (1) and rearranging terms, it is easily seen that e^i obeys the so-called *residual equation*

$$Ae^i = r^i. \quad (4)$$

The approximations v^i can be obtained by relaxation methods such as the Jacobi and the Gauß-Seidel method [11] which are defined by splitting up the sparse matrix A into an easily invertible part and the remainder [9]. With these methods, a relaxation operator is applied to v^i to give a better approximation v^{i+1} by the reduction of the error e^i related to v^i . In this way the sequence of approximations $v^0, v^1, \dots, v^n, \dots$ is "relaxed" to the solution u .

The basic idea of the multigrid approach is to employ different length scales to efficiently reduce the error. Specifically, one solves (4) exactly on a grid Ω_{n-1} that is *coarser* than Ω_n . Since (4) is solved on a coarser grid, the resulting value of e^i is an approximation that is used to *correct* the previous

approximation v^i that has been determined on the original grid Ω_n

$$v^{i+1} = v^i - e^i. \quad (5)$$

The crucial advantage of this approach can be understood by considering the Fourier expansion of the error e^i . The long wavelength components of e^i get only slightly reduced on the fine grid since their spatial extent exceeds the range of the relaxation operator. The use of a coarser grid renders those components shorter wavelength and thus "visible" to the relaxation operator. This improves the convergence of the solver dramatically, as compared to a single-grid based relaxation scheme, such as the SOR.

The simplest version of the multigrid algorithm is the so-called *two-grid iteration* that employs only two grid levels. In the i th iteration, this procedure starts from the approximation v^i of u in (1) and consists of the following five steps:

- 1) smooth v^i on the grid Ω_n by applying some suitable relaxation scheme, called *pre-smoothing*, as detailed in Section III-C;
- 2) compute the residual according to (2) and transfer it to the coarser grid Ω_{n-1} . This step is called *restriction*, see Section III-B;
- 3) solve exactly (4) on the grid Ω_{n-1} ;
- 4) interpolate the resulting e^i to the finer grid Ω_n .—called *prolongation*—see Section III-B; subsequently, calculate v^{i+1} from (5);
- 5) smooth v^{i+1} on the grid Ω_n by applying some relaxation method, termed *post-smoothing*, as described in Section III-C.

Since (4), considered on Ω_{n-1} , has the same form as (1) on Ω_n , it is possible to extend the two-grid algorithm to a sequence of grids that are increasingly coarser. This is achieved by recursively applying the complete algorithm (steps 1–5) at step 3. The recursion stops when the coarsest grid Ω_0 is reached. At that grid level, (4) is solved exactly. Since this grid usually only contains very few points, this can easily be done. This multi-scale algorithm defines one complete *multigrid iteration* (labeled by the super index i). The procedure is then repeated until the required convergence threshold is reached.

The above discussion of the multigrid iteration refers to a cycle structure called V-cycle (see Fig. 1). More generally, one may interpret the multigrid iteration as the recursive application of γ two-grid cycles at any grid level, the V-cycle being characterized by $\gamma = 1$. The case $\gamma = 2$ is called W-cycle. It is possible to use any number γ of two-grid cycles at each level, obtaining better convergence at the cost of increased complexity of the algorithm. In the present application, we have found that the W-cycle is the best compromise between the number of iterations and computational effort per iteration.

In passing, we note that the multigrid method can be used either as an iterative process or as a direct solver (the so-called *full multigrid* or *nested iteration* method). Since, in the present applications, acceptable initial guesses are available [13], we found the iterative version to be more effective.

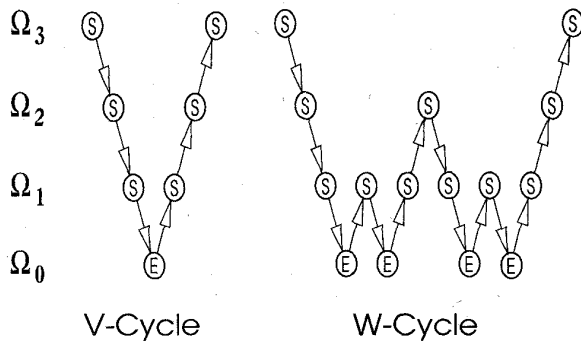


Fig. 1. Standard multigrid structure for a V and W-cycle. In this figure, the number of grid levels is four, the circles represent Smoothing or Exact solution operations, the arrows indicate prolongation (upward) and restriction (downward) operations. Ω_0 is the coarsest, and Ω_3 the finest grid.

III. OPTIMIZED MULTIGRID SOLVER FOR SEMICONDUCTOR DEVICES

In this section, we develop the ingredients of the present Poisson solver necessary to obtain a robust implementation of the multigrid scheme able to handle large spatial variations in the forcing function, inhomogeneously spaced rectangular grids and geometrically complex boundary conditions including surface or interface charges and different dielectrics, without constraints for the number of grid points.

The fundamental ingredients of the multigrid algorithm are the *prolongation* and *restriction* procedure, as well as the *relaxation* or *smoothing* scheme which is the core of the method. We have adapted these components to optimally deal with the situations encountered in realistic devices. Applications and results are presented in Section IV.

A. Coarsening

The first task in the multigrid approach is to define a sequence of grids $\Omega_n, \Omega_{n-1}, \dots, \Omega_0$. In this paper, we have chosen a *double coarsening* that is the coarser set Ω_{l-1} is obtained from Ω_l by doubling the grid spacing. With this scheme, all the points of Ω_{l-1} also belong to Ω_l which permits efficient transfer operations between grids and is recursively convenient [14].

The coarsening cycle is continued until the boundary conditions can no longer be adequately mapped onto the grid. Since the Dirichlet boundary conditions fix the potential, it is mandatory to have each Dirichlet area in the device represented by at least a single grid point on the coarsest grid. Areas with Neumann boundary conditions are less critical, on the other hand, and need not be present on all grid sets. When the coarsening process comes to an end along one direction, it is continued solely along the perpendicular one, i.e., the double coarsening is replaced by semicoarsening eventually.

In contrast to fast Fourier transform algorithms, the multigrid method does not require the total number of grid points to obey given conditions. However, the double-coarsening scheme does require some care when the total number of grid points is even in one or both directions. Consider, for

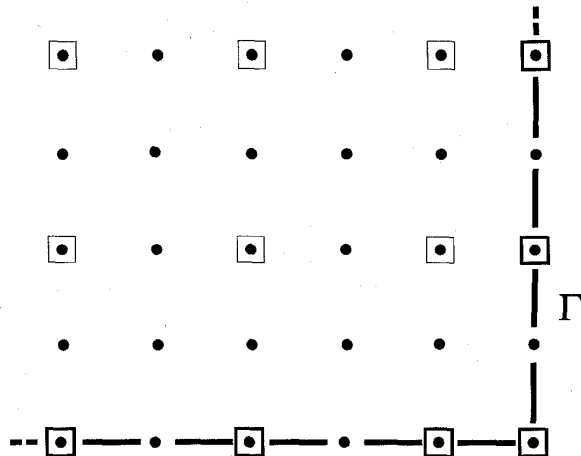


Fig. 2. The double coarsening scheme yields the grid Ω_l (circles) from Ω_{l-1} (framed circles). The line corresponds to the boundary Γ of the domain. The grid points in Ω_{l-1} are evenly spaced horizontally because there is an even number of grid points in Ω_l .

example, a one-dimensional horizontal grid with grid spacing Δ that contains an even number of grid points. In this situation, the coarsening procedure shifts the position of the right boundary to the left by an amount Δ . Particularly when the number of points in one dimension is $2^N + 2$ (N being a natural number), this “geometric mismatch” between all grids severely hampers prolongation and restriction, and causes an unacceptable slowing-down of the solver’s performance. We have solved this problem by adding one additional point at the right boundary of the coarser grid whenever necessary in order to keep the geometrical position of boundaries fixed on all grids, as shown in Fig. 2. As will be demonstrated in Section IV, the advantages of this scheme by far outweigh its cost, namely to introduce some irregular grid spacings (see Fig. 2). Clearly, if the finest grid happens to contain $2N + 1$ grid points, this problem never arises.

B. Prolongation and Restriction

Once the hierarchical grids have been generated, one needs to transfer the error from the coarser to the finer grids (*prolongation*) and in the opposite way (*restriction*) for the residual. We have employed the standard piecewise linear interpolation for the prolongation, passing on the values of those points that are common to both grids (framed circles in Fig. 3) and performing a linear or bilinear interpolation for the remaining points, as shown graphically in Fig. 3. This scheme is called *nine point prolongation* [10], [15] and can be represented by the transformation

$$\begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & 1 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}. \quad (6)$$

The situation is more subtle for the restriction transformations. In the present context, it turns out to be useful to use two different restriction operators, namely the so-called *full weighting* and the *half weighting* restriction. For a

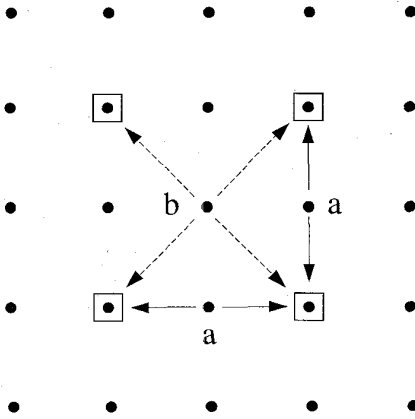


Fig. 3. Schematic representation of the coarse-to-fine prolongation scheme by piecewise linear interpolation. The values on the framed grid points of Ω_l are directly copied onto Ω_{l-1} , whereas the values on other points fall into two classes (a) and (b) and are obtained by linear and bilinear interpolation, respectively, as indicated by the arrows.

homogeneous grid with double coarsening, the full weighting restriction transformation reads

$$\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \quad (7)$$

whereas half weighting is represented by

$$\begin{bmatrix} 0 & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ 0 & \frac{1}{8} & 0 \end{bmatrix}. \quad (8)$$

The criteria that determine the most effective scheme in a given situation depend on the relaxation method, as will be discussed in detail in the next section. In principle, the adjoint of 6, i.e., full weighting, is always the optimal choice for the restriction operator [10]. However, when the smoother is a point-Gauß-Seidel-relaxation (see next section), half weighting is sufficiently accurate but permits a much faster and storage-saving implementation. Note that the weighting coefficients in 7 and 8 must be augmented by appropriate scale factors in the case of nonhomogeneous grids.

C. The Relaxation Scheme

The relaxation scheme forms the kernel of the method [16], [17]. Its task is to reduce the short wavelength Fourier components of the error on a given grid. The efficiency of the relaxation scheme is known to depend sensitively on details such as grid topology and boundary conditions. Therefore, there is not a single standard relaxation scheme that can be applied [16].

In this section we show that special combinations of two variants of Gauß-Seidel relaxation schemes provide a robust and efficient smoothing of the error for the two-dimensional Poisson's equation with highly complex boundary conditions.

As will be discussed below, even very complex devices can be mapped onto inhomogeneous two-dimensional grids with rectangular boundaries so that we used them throughout this

paper. Such grids consist of horizontal and vertical grid lines that we can refer to as X and Y directions, respectively.

We consider two Gauß-Seidel schemes [11], namely pointwise relaxation (PGS) and line relaxation (LGS). In both schemes, subsets of grid points—termed blocks following [11]—are formed and all block values of e^i are simultaneously updated such that (4) is obeyed. On the finest grid, v^i is updated and obeys (1). In PGS, a block contains only a single point whereas LGS operates on blocks containing contiguous points along one direction. When the line relaxation sweep is performed along only one direction, this procedure is termed simple LGS (respectively, X or Y -LGS), in contrast to alternating LGS (or XY -LGS) [17] where two consecutive line relaxation sweeps are performed along both X and Y directions.

The efficiency of these relaxation schemes can be measured by the so-called *smoothing factor* [18], [19]. For a better understanding, it is useful to briefly recall its definition. For simplicity, consider first a square grid with $N \times N$ grid points (N even) with periodic boundary conditions. Let the algebraic errors before and after a relaxation sweep by $e(x_1, x_2)$ and $\bar{e}(x_1, x_2)$, respectively, see (3). The Fourier components of e are given by

$$e(x_1, x_2) = \sum_{s,t=-N/2+1}^{N/2} c(\theta_s, \theta_t) \exp[i(\theta_s x_1 + \theta_t x_2)] \quad (9)$$

where $\theta_s = s\pi/N$ and $\theta_t = t\pi/N$. An analogous equation holds for \bar{e} . The convergence factor [20], or reduction factor [18] of the $\theta \equiv (\theta_s, \theta_t)$ component is defined by

$$\mu(\theta) = \left| \frac{\bar{c}(\theta)}{c(\theta)} \right|. \quad (10)$$

Finally, the *smoothing factor* is defined by [20]

$$\bar{\mu} = \max_{\rho\pi \leq |\theta| \leq \pi} \mu(\theta) \quad (11)$$

where ρ is the grid coarsening factor. A double coarsening scheme implies $\rho = 1/2$. Obviously, this definition places particular emphasis on the short wavelength Fourier components of the error [21], [20], because the remaining long wavelength components are reduced on the coarser grids.

On a square lattice, one has $\bar{\mu}_{\text{PGS}} = 0.5$, $\bar{\mu}_{\text{LGS}} = 0.44$ and $\bar{\mu}_{\text{XY-LGS}} = 0.14$ [18]. Thus, an application of XY -LGS reduces the short wavelength error components by almost one order of magnitude, whereas three cycles of PGS are needed to obtain an equivalent reduction of the error. On the other hand, the application of the XY -LGS smoother requires approximately twice as many operations as the PGS and more storage (typically three additional variables per grid point and direction). In addition, the simple and alternating LGS schemes require a setup procedure where a tridiagonal matrix must be factorized.

The situation changes completely for rectangular homogeneous grids with the ratio of the grid spacings along X and Y different from unity, $\eta = |\delta X / \delta Y| \neq 1$. In this case, the discrete Poisson's equation is a degenerate elliptic equation [22], [20], and the smoothing factor of the PGS

scheme dramatically increases whenever η deviates from unity. Then, error smoothing is obtained only along the dominant direction, i.e., the direction with the smaller grid spacing, and the performance of the Poisson solver goes down. In contrast, even a simple LGS always provides efficient smoothing when applied along the dominant direction. For this reason, we have employed this scheme in cases where a globally dominant direction exists (see Section III-D). The PGS is only used to refine the smoothing process along Dirichlet boundaries (contacts) (see Section III-D).

Since the SOR method is, in general, an efficient version of PGS, a grid spacing with $|\delta X/\delta Y| \neq 1$ tends to decrease the error reduction rate and to slow dramatically the solver, as shown in Section IV. This effect can only be avoided by using line instead point relaxation in the SOR algorithm (see the SLOR algorithm in [23]).

For a realistic device geometry, one has to deal with rectangular and inhomogeneously spaced grids. In this case, a globally dominant direction does not exist and the application of the XY-LGS smoother along both directions is computationally too costly. The simple LGS scheme, on the other hand, leads to satisfactory convergence only in a few favorable cases.

We have therefore developed and tested another concept that can be termed *grid alternating X and Y-LGS relaxation*. So far, we have focused on a given grid and discussed the relaxation on that grid. The key point in this new concept is to apply the X and Y-LGS schemes alternately on *successive* grids. If the X-LGS scheme is used on grid Ω_i , the Y-LGS relaxation is used on Ω_{i-1} (after the restriction of the residual) or on Ω_{i+1} (after prolongation of the error). We have found that this particular succession of alternating sweeps significantly improves the speed of the Poisson solver on inhomogeneous grids, as will be demonstrated in Section IV. Importantly, only the computational effort of a simple LGS is required with this new scheme.

The idea behind the grid alternating LGS scheme is that some components of the error, which are not efficiently reduced by line relaxation along a given direction, are *still visible* to the perpendicular line relaxation in an immediately adjacent grid. In other words, most of the error components, which are not reduced on a given grid by a line relaxation along one direction, can be propagated to the adjacent grid, where they can still be significantly reduced by LGS operating perpendicularly to that previous direction.

Thus, although for each multigrid cycle the effective error-reduction due to the grid alternating LGS is smaller than the XY-LGS one, the number of required operations is reduced. As a result, the total required time is sensibly less using grid alternating instead traditional XY-LGS, even if the number of iterations is increased. As final remark we want to stress that the efficiency of a relaxation scheme is strictly dependent on the grid structure, i.e., on the geometrical coupling between the grid points, and on the various boundary conditions required to simulate real devices. In some cases the combination of different schemes, like LGS plus PGS around the contact regions, or the grid alternating LGS itself, can dramatically reduce the execution time of the solver.

We have yet to specify the order in which blocks are processed. We have adopted the standard block labeling schemes in the smoothing procedure. For the LGS scheme, the alternating-line or zebra LGS numbering scheme is known to be most efficient. There, the odd lines are relaxed first, followed by the even ones. In the PGS scheme, we have used the checker-board or red/black labeling [10].

After the relaxation process has been carried out on a given grid, the residual vanishes on some blocks, namely on all black points in the case of PGS with red/black ordering, and on all even lines after an odd/even zebra line relaxation. This fact can be utilized to reduce both the storage allocation and the execution time since the zero values of the residual need not be restricted to the coarser grid. In general, nine grid points enter the restriction transformation, see (7). In case of red/black PGS, however, only five grids points have nonzero residuals. For the full-weighting restriction transformation, all five values are needed, whereas only the central grid point enters the half-weighting transformation. Thus, it is computationally efficient to select the half-weighting restriction scheme with PGS; the small loss in accuracy is compensated by a factor of five less operations. On the other hand, there is no advantage in using half-weighting in LGS. Since only the three points lying on the odd lines have nonzero residuals with an odd/even LGS, we can use the full-weighting transformation (7) and still need to access only three values instead of nine in the restriction process.

D. The Discretization Scheme

The grid spacing in a Poisson solver for particle-based simulators must be adapted to the length scale associated with the spatial changes of the potential. This length scale is given by the Debye length [9]. Since this scale is spatially varying, one is forced to use inhomogeneous grids. On the other hand, a large variation of the grid spacing can result in unacceptable losses of accuracy [24]. Therefore, we restrict the mesh expansion ratio [25] between neighboring grid points to a value of less than two. Fortunately, the increase of the number grid points is not a major issue in the multigrid method since the solution of the Poisson's equation on N points requires only $O(N)$ operations. It should be noticed that the requirement related to the Debye length produces grids with a high number of cells (up to 25 000, see Section IV), if the device geometry has to be faithfully represented.

When a device discretization is scaled to a coarser grid, the internal boundaries between different doping regions and materials will not, in general, maintain their relative position. The convergence of the multigrid algorithm can be severely hampered by such geometrical mismatches between the position of boundaries on different grids, particularly when Dirichlet boundary conditions apply. We have observed that this problem is less severe when the W-cycle instead of the simple V-cycle is used. Importantly, we have found that the errors due to this geometrical mismatch can be efficiently suppressed by introducing a further smoothing after the post-smoothing step. Specifically, we have applied a PGS smoothing just for the grid points along the contact areas of

the device. Since the number of these points is small compared to the total number of grid points, the computational cost of this additional step is also small.

E. Computational Details

Our implementation of the multigrid Poisson solver consists of about 2000 source lines of uncommented code in Fortran 77. All coefficients for smoothing, restriction, prolongation, including pointers that connect the different grid sets, as well as nearest neighbor tables, are determined in the setup part of the program. This procedure minimizes the time required for each multigrid iteration, although it increases the required memory to values which, in any case, are not critical in the total balance of the employed storage resources.

The total storage requirement (in bytes) of the present implementation of the Poisson solver is given by

$$B = N_T(17W_I + 26W_R) + 20NW_R \quad (12)$$

where N is the number of grid points in the finest (input) grid, N_T is the total number of grid points on all grid sets together, W_I and W_R are the length of integer and real words in bytes, respectively. The implementation of a solver with the same peculiarities, but based on the SOR method, would require

$$B = N(9W_I + 32W_R) \quad (13)$$

bytes (in this case $N_T \equiv N$). The difference in storage requirement is mainly due to the presence of a single grid in the SOR scheme, which reduces the total number of grid points to be stored and does not require the restriction-prolongation coefficients and the related pointers. An additional memory saving is due to the absence of the three (or six, in case of XY-LGS) real variables used to store the factors for line relaxation (see Section III-C). It should be pointed out that the requirements of the multigrid approach are due to the choice of the authors to reduce the execution time as much as possible, together with the irregular nature of the grids, in which *all* the involved coefficients depend on the position. A regular grid would require a remarkably smaller memory allocation.

In any case, it has to be stressed that the typical amount of memory required by the multigrid method is not dramatic even in the unfavorable cases presented here, being in the range of some Mbytes for a typical device geometry.

IV. RESULTS

In this section we demonstrate the efficiency of our multigrid Poisson solver. We have coupled this solver to a two-dimensional Monte Carlo and a cellular automaton particle device simulator and studied two different state-of-the-art semiconductor devices, namely a High Electron Mobility Transistor (HEMT) and a Metal Oxide Semiconductor Field Effect Transistor (MOSFET) [26]. The first type is the standard representative of microwave devices based on GaAs, and the second one is the most common Si device for digital applications. By comparing the performance of the multigrid solver with that of an optimized SOR solver, we show that for both applications the multigrid solver is faster by more than an order of magnitude.

A. HEMT

The basic element of a HEMT is the heterostructure between two different semiconductor materials that provides a confining potential and allows to spatially separate the free charge carriers from the doping atoms. High frequency performance can be obtained by using suitable materials (in our case InGaAs for the channel) together with some sophisticated device geometry such as pulsed doping, gate recess, and T-shape of the gate [27]. While the discussion of such aspects is beyond the scope of the present paper, it has to be stressed that these technological and geometrical features have to be accounted for in a realistic simulator.

In Fig. 4, the geometry of the simulated submicron T-gate Gallium Arsenide HEMT is shown. From top to bottom, there is a layer of highly doped GaAs (light gray), followed by two AlGaAs layers (darker gray) which surround the InGaAs channel (darkest gray). The bottom thick layer is the GaAs substrate with a thin AlGaAs layer embedded in its upper part. Dirichlet boundary conditions are imposed on the contacts (black areas), a surface charge with a sheet concentration of $4 \times 10^{12} \text{ cm}^{-2}$ is assumed at the free surfaces at the top of the device and a zero perpendicular field (Neumann condition) is set on the remaining boundaries.

The grid is homogeneously spaced along the horizontal X direction, while a nonuniform grid is required for the Y direction, to account for the different doping concentrations and material compositions. A portion of the actual grid is shown in Fig. 4(b). The number of points in the finest grid is 258×89 ; the resulting multigrid space includes approximately 31 000 grid points distributed over 8 grid levels as detailed in Table I. The high number of grid points, in particular along the x -axis, is due to the small Debye length in those regions with high doping profile. Due to the dopant concentration of $5 \times 10^{18} \text{ cm}^{-3}$ in the cap region, the corresponding Debye length amounts in fact to 2 nm. We found that the choice of a grid spacing of 5 nm along the source-drain direction, together with a frequent solution of the Poisson's equation (every femtosecond), is the maximum achievable spacing to avoid unphysical effects such as the "numerical heating" of the carrier gas [9].

This space has been generated from the finest grid Ω_7 and the coarsening procedure stopped after grid level Ω_3 along the Y direction, following the rules developed in Section III-A. We point out that the choice of Ω_7 is particularly unfavorable and provides a stringent test of the multigrid solver. Since $258 = 2^8 + 2$, the presently adopted double-coarsening procedure results in a very irregular spacing for the coarser grids (see Section III-A and Fig. 2). We have found that the choice of optimal grid spaces (e.g., with $257 = 2^8 + 1$ points on the finest grid) give a speedup of 25%.

The Monte Carlo and cellular automaton procedure used for the HEMT simulation will be described elsewhere [28]. Here we just present the output of the Poisson solver once a stationary solution is reached. The contour plot of the two-dimensional potential for the simulated HEMT is shown in Fig. 5.

In Fig. 6, we depict the computer time required by the multigrid and SOR Poisson solver in the HEMT simulation,

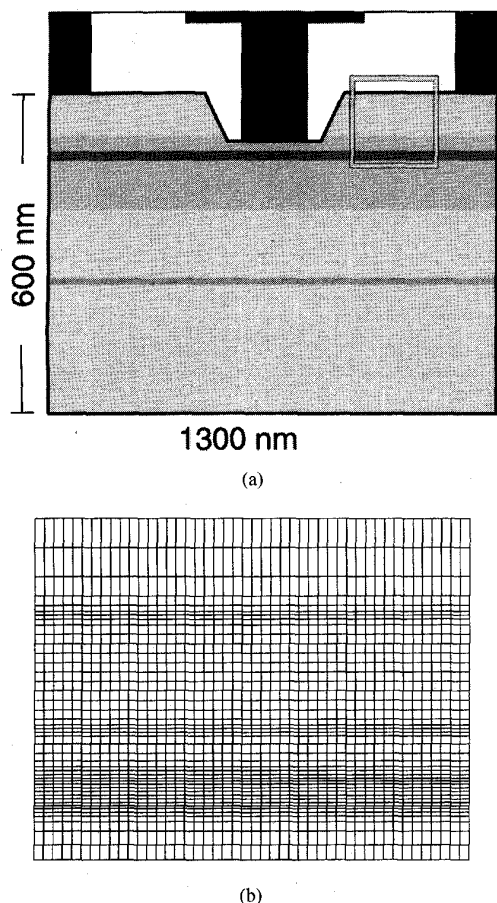


Fig. 4. (a) Geometrical structure of a T-gate GaAs HEMT. The black areas (on the top) represent, from left, the source, gate (“T” shaped) and drain contacts. The Poisson’s equation is solved in the rectangular domain which includes the two white regions (representing air) around the gate. Materials with different dielectric constant are drawn with different gray tones. (b) The portion of the grid, emphasized by the square frame between gate and drain, is shown.

TABLE I
GRID SPACE STRUCTURE FOR THE HEMT

grid	points	X-points	Y-points	coarsening
Ω_7	22962	258	89	X,Y
Ω_6	5850	130	45	X,Y
Ω_5	1518	66	23	X,Y
Ω_4	408	34	12	X,Y
Ω_3	126	18	7	X,Y
Ω_2	40	10	4	X
Ω_1	24	6	4	X
Ω_0	16	4	4	-

as a function of the convergence threshold. The comparison has been carried out by using the stationary charge distribution from the simulation and by initializing the two solvers with zero potential everywhere. The overrelaxation parameter in the SOR was optimized to achieve optimal convergence [29] and Chebyshev acceleration was employed. The solvers were executed on a workstation with a single, super-scalar, processor. Since both methods are iterative, convergence has been checked after each iteration. As a measure of convergence,

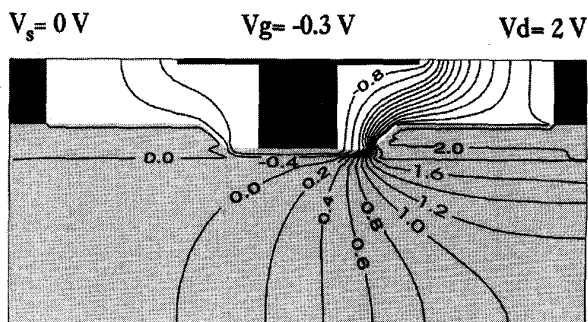


Fig. 5. Contour plot of the potential distribution in the HEMT. The external potentials on the source, gate, and drain contacts are shown.

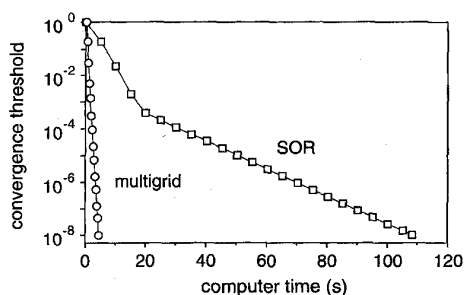


Fig. 6. Comparison between the convergence behavior of multigrid and successive over-relaxation Poisson solvers, as applied to the HEMT. The computer time, in seconds, is measured every iteration in the multigrid (circles) and every 100 iterations in the SOR (squares). The convergence threshold is the maximum allowed relative difference between two successive iterations.

the maximum relative difference of the potential after two consecutive iterations has been taken. A physically acceptable threshold value for the device simulations lies in the range $[10^{-6}, 10^{-8}]$. The relaxation scheme used in this test was simple LGS along the X -direction. Since the grid is fairly regular, there is no pronounced advantage in using the grid-alternating technique proposed in Section III-C.

As one can see, the slope of the multigrid curve is constant, i.e., the error reduction rate of the multigrid does not depend on the threshold. In contrast, the convergence of the SOR slows down after a relatively fast initial decay because the low-frequency components of the error that are inadequately reduced. It should be noted that even the *initial* error reduction is performed more efficiently by the multigrid, due to the small smoothing factor (see Section III-C) of LGS compared to the one of the point relaxation scheme used by the SOR. The HEMT grid topology, together with the irregular boundary conditions, is responsible for the huge number of iterations required by the SOR to converge (2200 iterations with 10^{-8} threshold).

As a consequence, the multigrid scheme is seen to be about 30 times faster than the SOR in this application. As noticed above, such speed-up is obtained for an initial potential which is zero everywhere on the device. When an initial guess is used—e.g., the potential computed in the previous iteration, with a charge distribution 1 femtosecond older than

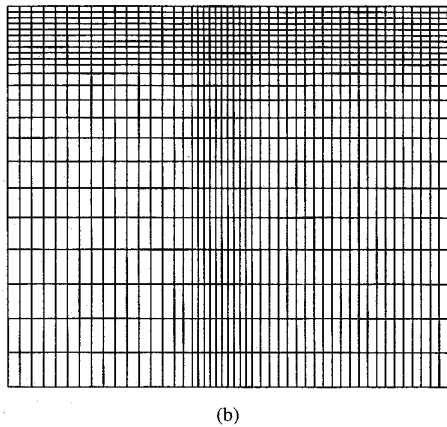
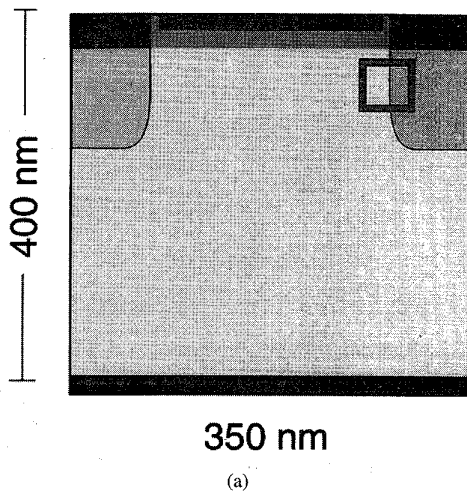


Fig. 7. (a) Geometrical structure of the simulated MOSFET. The black areas on the top depict, from left, the source, gate and drain contacts. A backgate contact is present at the bottom. The gray area below the gate represent a 5 nm layer of silicon oxide. (b) The portion of the grid, highlighted by the square frame on the upper right corner of the transistor, is shown.

the current one—the speed-up amounts to a factor 12 (18) when a convergence threshold of 10^{-6} (10^{-8}) is used.

Typically, the solution of Poisson's equation by the multigrid method requires about 1 sec with a 10^{-6} threshold, less than half of the time needed for a Monte Carlo cycle of 1 fs when 150 000 particles are used.

B. MOSFET

The MOSFET is the most advanced device in silicon-based technology [26]. The presence of highly doped regions under the source and drain contacts (see Fig. 7) requires a grid spacing which is irregular in both the X and Y direction. Furthermore, the channel of the MOSFET is confined at the Si/SiO₂ interface over a dimension of a few hundred angstrom, thus requiring a very fine grid in the top portion of the device. Following the same considerations adopted in generating the HEMT grid, we have chosen the finest grid for the complete MOSFET to have 221×129 points; the resulting grid space consisted of 7 grid levels, with the coarsest one

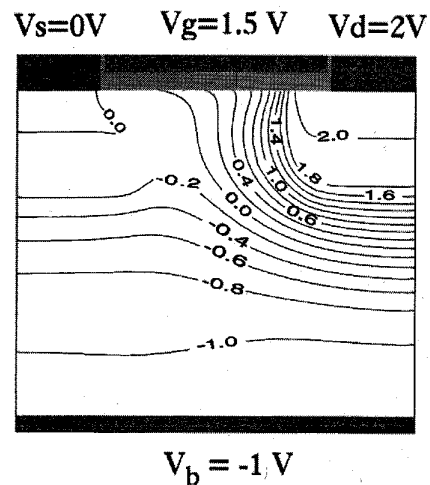


Fig. 8. Contour plot of the potential distribution in the MOSFET. The applied potentials on source, gate, drain, and backgate (including built-in) contacts are shown.

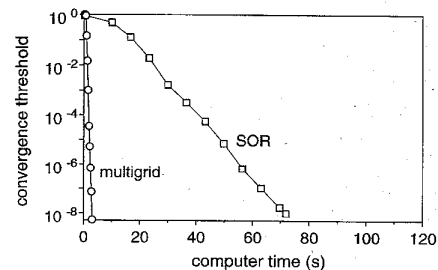


Fig. 9. Comparison between the convergence behavior of multigrid and successive overrelaxation Poisson solvers as applied to the MOSFET. The computer time, in seconds, is measured every iteration in the multigrid (circles) and every 100 iterations in the SOR (squares).

having 5×5 points. The presence of a backgate contact on the bottom side of the transistor should be noted.

Fig. 8 shows a contour plot of the computed potential. Because of the inhomogeneous grids, the method of grid alternating LGS—that we have introduced in Section III-C—is highly profitable. Compared to a standard XY-LGS relaxation, our new smoothing scheme requires the same number of iterations but only 50% of the computer time. In addition, the points around the Dirichlet's areas have been subsequently smoothed with a single iteration of PGS, as noted before.

In Fig. 9, the convergence rates of the multigrid and SOR Poisson solver are compared for the MOSFET geometry. In this case, the resulting speed-up of the multigrid scheme is again excellent. The scheme is faster than SOR by a factor of 20 in this case, in spite of the fact that the backgate contact has been introduced to prevent very long wavelength errors from occurring, and to make this example a favorable case for SOR. Indeed, Fig. 9 shows that both methods are able to reduce the error by approximately the same amount per iteration in this example (i.e., the slope of both the curves is constant), but the absolute value of the error reduction is much larger in the multigrid scheme.

V. CONCLUSION

We have presented an implementation of the multigrid algorithm for the solution of two-dimensional Poisson's equation in the context of particle semiconductor devices simulations. The results clearly demonstrate the efficiency of the multigrid scheme. The complex boundary conditions and inhomogeneous grids that are dictated by the physics of the devices require a careful tuning and adaptation of all parts of the multigrid method, such as coarsening, prolongation, restriction, relaxation. For the central component, the smoothing, we have developed a new variant of the Gauß-Seidel-type relaxation scheme that is particularly suited for device grids that lack globally dominant directions. It is hoped that the remarkable gain in performance due to the multigrid Poisson solver removes a major bottleneck in particle device simulators.

REFERENCES

- [1] C. Jacoboni and P. Lugli, *The Monte Carlo Method for Semiconductor Device Equations*. Wien-New York: Springer-Verlag, 1989.
- [2] K. Kometer, G. Zandler, and P. Vogl, "Lattice-gas cellular-automaton method for semiclassical transport in semiconductors," *Phys. Rev. B*, vol. 46, pp. 1382-1394, July 1992.
- [3] R. W. Hockney, "Measurements of collision and heating times in a two-dimensional thermal computer plasma," *J. Comput. Phys.*, vol. 8, pp. 19-44, 1971.
- [4] U. Ravaioli, "Vectorization of Monte Carlo algorithms for semiconductor simulation," in *Monte Carlo Device Simulation: Full Band and Beyond*, K. Hess, Ed. Dordrecht, The Netherlands: Kluwer Academic, 1991, ch. 9, pp. 267-284.
- [5] A. Rein, G. Zandler, M. Saraniti, P. Lugli, and P. Vogl, "Novel transport simulation of vertically-grown MOSFET's by cellular automaton method," in *Proc. 1994 IEEE Int. Electron Devices Meeting*, Dec. 1994, pp. 351-354.
- [6] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*. Wien-New York: Springer-Verlag, 1984.
- [7] C. M. Snowden, *Intro. Semiconductor Device Modelling*. Singapore: World Scientific, 1986.
- [8] J. Molenaar, "Multigrid methods for semiconductor device simulation," Center Math. Comput. Sci., Amsterdam, The Netherlands, Tech. Rep. CWI TRACT 100, Nov. 1993.
- [9] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*. Bristol, UK: Adam Hilger, 1988.
- [10] W. Hackbusch, *Multi-Grid Methods and Applications*. Berlin, Germany: Springer-Verlag, 1985.
- [11] D. M. Young, *Iterative Solution of Large Linear Systems*. Comput. Sci. Applied Math., New York: Academic, 1971.
- [12] S. Wang, *Fundamentals of Semiconductor Theory and Device Physics*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [13] S. E. Laux and M. V. Fischetti, "Numerical aspects and implementation of the DAMOCLES Monte Carlo device simulation program," in *Monte Carlo Device Simulation: Full Band and Beyond*, K. Hess, Ed. Dordrecht, The Netherlands: Kluwer Academic, 1991, ch. 1, pp. 1-26.
- [14] A. Brandt, "Multigrid techniques: 1984 guide with applications to fluid dynamics," Ges. für Mathematik u. Datenverarbeitung mbH, Bonn, Germany, Monograph 85, May 1984.
- [15] P. Wesseling, "Theoretical and practical aspects of a multigrid method," *SIAM J. Sci. Statist. Comput.*, vol. 3, pp. 387-407, 1982.
- [16] A. Brandt, "Guide to multigrid development," in *Multigrid Methods: Proc. Conf. Köln-Portz*, Nov. 23-27, 1981, W. Hackbusch and U. Trottenberg, Eds. no. 960 in Lecture Notes in Mathematics. Berlin, Germany: Springer-Verlag, no. 960 1982, pp. 220-312.
- [17] K. Stüben and U. Trottenberg, "Multigrid methods: Fundamental algorithms, model problem analysis and applications," in *Multigrid Methods: Proc. Conf. Köln-Portz*, Nov. 23-27, 1981, W. Hackbusch and U. Trottenberg, Eds. no. 960 in Lecture Notes in Mathematics. Berlin, Germany: Springer-Verlag, no. 960 in Lecture Notes in Mathematics, 1982, pp. 1-176.
- [18] R. Kettler, "Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods," in *Multigrid Methods: Proc. Conf. Köln-Portz*, Nov. 23-27, 1981, W. Hackbusch and U. Trottenberg, Eds., no. 960 in Lecture Notes in Mathematics. Berlin: Springer-Verlag, 1982, pp. 502-534.
- [19] C.-A. Thole and U. Trottenberg, "Basic smoothing procedures for the multigrid treatment of elliptic 3D operators," *Applied Math. Computation*, vol. 19, pp. 333-345, 1986.
- [20] A. Brandt, "Multi-level adaptive solutions to boundary-value problems," *Math. Computation*, vol. 31, pp. 333-390, Apr. 1977.
- [21] C. J. Kuo and B. C. Levy, "Two-color Fourier analysis of the multigrid method with red-black Gauss-Seidel smoothing," *Applied Math. Computation*, vol. 29, pp. 69-87, 1989.
- [22] A. Brandt, "Multigrid solvers on parallel computers," in *Elliptic Problem Solvers*, M. H. Schultz, Ed. New York: Academic, pp. 39-84, 1981.
- [23] R. S. Varga, *Matrix Iterative Analysis*. Series in Automatic Computation, Englewood Cliffs, N.J.: Prentice-Hall, 1962.
- [24] E. K. de Rivas, "On the use of nonuniform grids in finite-difference equations," *J. Computat. Phys.*, vol. 10, pp. 202-210, 1972.
- [25] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation—Foundations and Applications*. Amsterdam: North-Holland, 1985.
- [26] S. Sze, Ed., *High-Speed Semiconductor Devices*. New York: Wiley, 1991.
- [27] F. Ali and A. Gupta, *HEMTs & HBTs: Devices, Fabrication, and Circuits*. Boston-London: Artech House, 1991.
- [28] P. Lugli, M. Paciotti, R. Fauquembergue, J. Thobel, F. Dessenne, E. Calleja, E. Munoz, J. S. Rojas, R. Deutchmann, and G. Zandler, "HEMT modelling and simulation," in *Pseudomorphic HEMTs: Technology and Applications*, L. Ross and S. Swenson, Eds. The Netherlands: Kluwer, 1995.
- [29] P. J. Roache, *Computational Fluid Dynamics*. Albuquerque, NM: Hermosa, 1976.



Marco Saraniti received the Laurea degree in physics from the University of Modena, Italy, in April 1991.

He has collaborated with the Walter Schottky Institut (WSI) of the Technical University of Munich, Germany in 1992 to develop computational methods in device simulations. In 1993, he started the Ph.D. program at WSI on "Parallel implementation of the cellular automaton method for electron device simulation." His research work focuses on numerical aspects of the simulation of electron devices.

Achim Rein received the M.S. degree in physics from the Technical University of Berlin, Germany, in March 1992.

He is currently working towards the Ph.D. degree at the Walter Schottky Institut (WSI) of the Technical University of Munich, Germany, on cellular automata approaches for semiconductor transport. From 1990 to 1992 he studied the dynamical Hall effect as a mechanism for chaotic behavior.



Günther Zandler received the Ph.D. degree from the University of Innsbruck, Austria, in 1989.

In 1988, he became a research associate with the Institute for Experimental Physics at the University of Innsbruck. In 1989, he joined the Physics Department of the Technical University of Munich as a research associate for theoretical semiconductor physics. His research interests focuses on the simulation of electron devices, high field transport, and ultrafast phenomena in bulk and nano-structured semiconductors.



Peter Vogl received the Ph.D. degree from the University of Graz, Austria, in 1974.

In 1975, he became an Assistant Professor with the Institute of Theoretical Physics at the University of Graz, and received tenure in 1980. In 1990, he joined the Physics Department of the Technical University of Munich as an Associate Professor and, in 1993, as a full professor. He has held many visiting positions in the U.S., German universities, and research laboratories. He has published more than 100 papers in the area of theoretical solid-state physics. His research interests cover electronic structure of nano-structured semiconductors and polymers, semiclassical and quantum transport, and device simulations.

Paolo Lugli received the degree in physics from the University of Modena, Italy, in 1979. He received the M.S. degree in 1982, and the Ph.D. degree in 1985, both in electrical engineering from Colorado State University, Fort Collins.

In 1985, he joined the Physics Department of the University of Modena as a research associate. From 1988 to 1993 he was an Associate Professor of solid-state physics with the Engineering Faculty of the 2nd University of Rome Tor Vergata. In 1993, he was appointed as Full Professor of Optoelectronics at the same University. His current research interests involve the numerical simulation of semiconductor devices for electronics and optoelectronics applications, the Monte Carlo simulation of ultrafast phenomena in polar semiconductors, and the theoretical study of transport processes in nanostructures. He is the author of more than 100 of scientific publications, and coauthor of the book *The Monte Carlo Method for Semiconductor Device Simulations*.