IMPACT OF COMPUTING IN SCIENCE AND ENGINEERING 2, 219–243 (1990)

A Multigrid Approach for the Solution of the 2D Semiconductor Equations

J. MOLENAAR AND P. W. HEMKER

Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Received February 22, 1990

J. Molenaar and P. W. Hemker, A Multigrid Approach for the Solution of the 2D Semiconductor Equations, *IMPACT of Computing in Science and Engineering* 2, 219–243 (1990).

In this paper a multigrid method for the solution of the steady semiconductor equations is presented. The discretization is made on an adaptive grid, by means of a mixed finite element method on rectangles, with the trapezoidal quadrature rule. In this way the resulting scheme reduces to the well-known Scharfetter–Gummel discretization. The grid transfer operators are selected in accordance with the discretization. The multigrid solution method is based on a collective, symmetric five-point Vanka relaxation, and—in order to admit very coarse grids—a local damping of the coarse grid correction is applied. It is shown that the convergence rate is independent of the grid size. Since nested iteration is combined with the multigrid iteration, the resulting solution method has optimal efficiency. © 1990 Academic Press, Inc.

1. INTRODUCTION

There has been a great deal of interest recently in the numerical simulation of the electric behavior of semiconductor devices. For a survey see [1–4]. Various programs that solve such problems for an industrial environment are now available. However, it has also become clear that there is still an increasing demand for faster and more flexible and robust programs. The model that describes the distribution of the electric field and the concentration of carriers in a semiconductor, the drift–diffusion model, is a system of three nonlinear elliptic partial differential equations: a nonlinear Poisson equation and two continuity equations.

It has been known for some time now that multigrid (MG) methods are possibly the most efficient for such equations, because the computational effort for solving the large discrete systems can be proportional to the number

0899-8248/90 $3.00
Copyright © 1990 by Academic Press, Inc.
All rights of reproduction in any form reserved.

of unknowns. Therefore various attempts to apply MG to the simulation of semiconductor devices have already been made [5–7]. However, up to now for several reasons, the success was not up to the expectations. It appears that the coarsest level of discretization, used in the sequence of grids, still has to be rather fine, and therefore requires a significant computational effort.

This is due to the fact that there are several difficulties associated with solving these equations. First, the equations are of a singular perturbation character and the dependent variables may vary rapidly over small regions of the device. Second, the system is strongly nonlinear and the equations are badly scaled.

These difficulties require a careful discretization, for which the requirements will include conservation of charge (electrons and holes) and nonnegative solutions. The demands are well known now, but they ask for special attention in the case of a multigrid method, where one wants to construct a sequence of discretizations starting from very coarse meshes. Further, for a multigrid method one needs grid transfer operators between the coarser and finer grids. Such operators usually function best when they are chosen consistently with the discretization method used. These considerations, and the knowledge that the fluxes of the solutions are usually smoother functions in space than the scalar-dependent variables, make it desirable for us to apply a mixed finite element method for the discretization of the equations.

In order to avoid unnecessary computations, to handle irregular geometries, and to obtain the required accuracy in an efficient manner, it is also desirable to have a finer mesh in regions where the solution is varying rapidly, and a coarser one in regions where it is varying slowly. Therefore we introduce an adaptive mesh refinement method that fits with the multigrid method used. Surveys of adaptive procedures are found, e.g., in Babuska and Rheinboldt [8] and Oden [9]. Application in the context of multigrid is found, e.g., in Schmidt and Jacobs [10].

The multigrid method presented in this paper is a further development of earlier work that was done in one dimension [11–13] for the drift–diffusion model and in two dimensions for the nonlinear Poisson equation [14]. New aspects in the present results are the use of Vanka relaxation for this set of equations, the application of a special kind of damping for the coarse grid correction (in the MG method), and the use of appropriate minimizing functionals for the selection of initial estimates.

An outline of the paper is as follows. In Section 2 we present the equations solved, and in Section 3 the grid and data structure used for the adaptive discretization. The discretization itself is explained in Section 4. In sections 5, 6, and 7 we describe the multigrid method and give details about the Vanka relaxation and the adapted coarse grid correction. In Section 8 we describe the construction of the initial estimates, and finally, in Section 9 we report numerical results. First, the convergence of the multigrid iteration is dem-

onstrated for uniform grids, and then an example is shown of a solution on a self-adapted grid.

2. The Equations

A steady semiconductor device can be modeled by

$$-\nabla \cdot \mathbf{J}_\psi = q(p - n + D), \tag{2.1a}$$

$$-\nabla \cdot \mathbf{J}_n = -qR, \tag{2.1b}$$

$$-\nabla \cdot \mathbf{J}_p = +qR, \tag{2.1c}$$

where $\mathbf{J}_\psi$, $\mathbf{J}_n$, and $\mathbf{J}_p$ are defined by

$$\mathbf{J}_\psi = \epsilon \nabla \psi, \tag{2.2a}$$

$$\mathbf{J}_n = q\mu_n\left(\frac{1}{\alpha}\nabla n - n\left(\nabla\psi + \frac{1}{\alpha}\nabla \log n_i\right)\right), \tag{2.2b}$$

$$\mathbf{J}_p = -q\mu_p\left(\frac{1}{\alpha}\nabla p + p\left(\nabla\psi - \frac{1}{\alpha}\nabla \log n_i\right)\right). \tag{2.2c}$$

Equation (2.1a) is Poisson's equation; n and p are the concentrations of electrons and holes, respectively, and the dope function D is a given function of the space variable. The relation between the electric displacement current $\mathbf{J}_\psi$ and the electrostatic potential ψ is given by (2.2a). Equations (2.1b) and (2.1c) are continuity equations; $\mathbf{J}_n$ and $\mathbf{J}_p$ represent the electron and hole current densities, respectively, and R is the recombination rate of electrons and holes. The quantities q, ϵ, α, n_i, μ_n, and μ_p are the electron charge, the permittivity, the inverse of the thermal voltage, the intrinsic concentration of free charge carriers, and the electron and hole mobilities, respectively. For simplicity, in the present paper we only consider constant ϵ, α, n_i, μ_n, μ_p and $R = 0$. The problem, simplified this way, corresponds to the example problem used, e.g., in [2]. It preserves many of the characteristic difficulties found in practical problems, where—based on physical models—different nonlinear functions are used, e.g., for μ_n, μ_p, and R. For details we refer to [2].

In our calculations we use the quasi-Fermi potentials ϕ_n and ϕ_p, which are related to n and p by

$$n = n_i e^{\alpha(\psi - \phi_n)}, \tag{2.3a}$$

$$p = n_i e^{\alpha(\phi_p - \psi)}. \tag{2.3b}$$

Expressed in (ψ, ϕ_n, ϕ_p) the equations are strongly nonlinear, but the range of the values assumed by(ψ, ϕ_n, ϕ_p) is of the same order as the voltages applied to the device. This makes them better suited for numerical computations than, e.g., (ψ, n, p) for which the range of values is much wider (cf. [2]).

Using (2.3) we write (2.2) in terms of (ψ, ϕ_n, ϕ_p),

$$\mathbf{J}_\psi = \epsilon \nabla \psi, \tag{2.4a}$$

$$\mathbf{J}_n = -\bar{\mu}_n e^{\alpha(\psi - \phi_n)} \nabla(\alpha \phi_n), \tag{2.4b}$$

$$\mathbf{J}_p = -\bar{\mu}_p e^{\alpha(\phi_p - \psi)} \nabla(\alpha \phi_p), \tag{2.4c}$$

with

$$\bar{\mu}_n = \frac{n_i q \mu_n}{\alpha}, \qquad \bar{\mu}_p = \frac{n_i q \mu_p}{\alpha}. \tag{2.5}$$

For the discretization of the equations (2.1)–(2.2) we use the Slotboom variables (ψ, Φ_n, Φ_p), which are defined by

$$\Phi_n = e^{-\alpha \phi_n}, \tag{2.6a}$$

$$\Phi_p = e^{+\alpha \phi_p}. \tag{2.6b}$$

Expressed in these variables (2.2) becomes

$$\mathbf{J}_\psi = \epsilon \nabla \psi, \tag{2.7a}$$

$$\mathbf{J}_n = +\bar{\mu}_n e^{\alpha \psi} \nabla \Phi_n, \tag{2.7b}$$

$$\mathbf{J}_p = -\bar{\mu}_p e^{-\alpha \psi} \nabla \Phi_p. \tag{2.7c}$$

The numerical range of the set of variables (ψ, Φ_n, Φ_p) renders them unsuitable for practical calculations, but they are attractive from a theoretical point of view because it makes the individual continuity equations symmetric (without first-order derivatives) and linear in Φ_n and Φ_p.

For an elaborate discussion of the choice of variables, see [2].

3. Grid and Data Structure

It is well known that the semiconductor equations show sharp layers in their solution, so it is attractive to use adaptive grids. In this section we present a method of grid generation that is very suitable for local refinement (cf. [10,

15]) and that can handle a fairly wide range of geometries encountered in device simulation.

It is assumed that the domain $\Omega \subset \mathbb{R}^2$, on which the equations (2.1)–(2.2) have been defined, can be covered by a regular mesh of rectangular blocks. A subset of these blocks should exactly cover Ω and these blocks form the coarsest grid G^0 in a sequence of nested grids for the discretization of (2.1)–(2.2).

On a set of blocks a refinement operator σ is defined as the set-valued mapping, which splits one block Ω_i^l of the grid into four smaller ones (see Fig. 3.1)

The class Q of admissible grids is specified recursively by two rules:

$$\text{i. } G^0 \in Q,$$
$$\text{ii. } G \in Q \Rightarrow \sigma(G) \in Q. \tag{3.1}$$

The level l of a block Ω_i^l is defined as the minimum number of refinement steps between Ω_i^l and a block of G^0. Using this definition we can classify the grids: a grid G^l of level l is the set of all blocks Ω_i^l. If a locally refined grid is used, there are interfaces between grids of a different level (see Fig. 3.1). Following Schmidt and Jacobs [10] such interfaces are called "green" interfaces.

In this way a nested sequence of partitionings of the domain Ω is obtained. Finer meshes may cover parts of Ω, but as soon as a fine level mesh exists in some area, also all coarser levels are available. The data structure used for the implementation, a quad tree, reflects this structure of the grids. In every node of the tree (a block or "cell") there are four pointers to possible offspring. The leaves of the tree correspond to unsplit blocks. In addition, every node contains four pointers to interfaces, representing the sides on the block. Neighboring blocks on the same level are connected by their common interface. These interfaces are also used to distinguish between green interfaces and physical boundaries.

To accommodate general geometries, the root of the tree does not need to represent G^0. So the first (negative) levels in the quad tree may contain entries, which are not necessarily related to a part of the domain. However, there

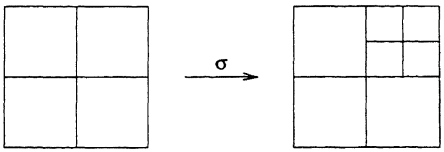

FIG. 3.1. Refining the mesh by a refinement operator σ.

must be a level in the tree which corresponds to G^0 exactly. The different numerical operations on data in the data structure are made by procedures that scan all cells, or all cells that satisfy a specific condition (e.g., all cells on a specified level), and which operate on each cell that is visited.

4. DISCRETIZATION

To discretize (2.1) and (2.4) we use the mixed finite element method, based on lowest order Raviart–Thomas elements for rectangles [16]. By the use of a suitable quadrature rule the discretization is equivalent to the well-known Scharfetter–Gummel scheme.

Boundary conditions are either Dirichlet or Neumann; the corresponding parts of the boundary are denoted by $\delta\Omega_D$ and $\delta\Omega_N$, respectively.

Before we describe the discretization, we note that all three equations, expressed in Slotboom variables, can be written as

$$\nabla \cdot \mathbf{u} = R(\phi, \mathbf{x}), \tag{4.1a}$$

$$a^{-1}\mathbf{u} = \nabla\phi, \tag{4.1b}$$

$$\mathbf{u}\cdot\mathbf{n} = 0, \quad \text{at} \quad \delta\Omega_N, \tag{4.1c}$$

$$\phi = \phi_D, \quad \text{at} \quad \delta\Omega_D, \tag{4.1d}$$

where ϕ is a scalar and $\mathbf{u}$ a vector variable; $\mathbf{n}$ is the outward unit vector normal to $\delta\Omega$.

Let $L_2(\Omega)$ be the space of square integrable functions on Ω, with inner-product

$$(\phi, \tau)_{L_2} = \int_\Omega \phi \cdot \tau d\Omega$$

and let $H(\text{div}, \Omega)$ be defined by

$$H(\text{div}, \Omega) = \{\mathbf{u} | \mathbf{u} \in (L_2(\Omega))^2, \text{ div } \mathbf{u} \in L_2(\Omega), \mathbf{u}\cdot\mathbf{n} = 0, \text{ at } \delta\Omega_N\},$$

with norm

$$\|\mathbf{u}\|^2_{H(\text{div},\Omega)} = \|\mathbf{u}\|^2_{(L_2(\Omega))^2} + \|\text{div } \mathbf{u}\|^2_{L_2(\Omega)}.$$

By introduction of the product space

$$\Lambda(\Omega) = L_2(\Omega) \times H(\text{div}, \Omega),$$

the weak formulation of (4.1) is the following: find $(\phi, \mathbf{u}) \in \Lambda$ such that for all $(\tau, \mathbf{t}) \in \Lambda$

$$(\tau, \operatorname{div} \mathbf{u}) = (\tau, R), \tag{4.2a}$$

$$(a^{-1}\mathbf{u}, \mathbf{t}) + (\phi, \operatorname{div} \mathbf{t}) = \langle \phi_D, \mathbf{t} \rangle, \tag{4.2b}$$

where

$$\langle \phi, \mathbf{t} \rangle = \int_{\delta\Omega_D} \phi \mathbf{t} \cdot \mathbf{n} d\Gamma. \tag{4.3}$$

The Neumann boundary conditions are automatically satisfied by all $\mathbf{u} \in H(\operatorname{div}, \Omega)$.

On each grid G^l, (4.2) is discretized by the lowest order Raviart–Thomas elements. For every block Ω_i^l of grid G^l we define the indicator function $\epsilon_i^l \in L_2(\Omega)$,

$$\epsilon_i^l(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \notin \Omega_i^l, \\ 1, & \mathbf{x} \in \Omega_i^l. \end{cases} \tag{4.4}$$

In addition, according to [16], a vector function $\mathbf{e}_j^l$ is introduced for every edge E_j^l, not part of the Neumann boundary, such that $\mathbf{e}_j^l(\mathbf{x})$ is linear on blocks Ω_i^l and

$$\mathbf{e}_j^l \cdot \mathbf{n}_k^l = \delta_{jk}, \tag{4.5}$$

where δ_{jk} denotes the Kronecker delta, and $\mathbf{n}_k^l$ denotes the outward unit vector normal to E_k^l; this choice ensures $\mathbf{e}_j^l \in H(\operatorname{div}, \Omega)$.

The discrete spaces spanned by $\{\epsilon_i^l\}$ and $\{\mathbf{e}_j^l\}$ are called $L^l(\Omega^l)$ and $H^l(\operatorname{div}, \Omega^l)$, respectively. Their Cartesian product space is

$$\Lambda^l(\Omega^l) = L^l(\Omega^l) \times H^l(\operatorname{div}, \Omega^l).$$

The discrete approximation $(\phi^l, \mathbf{u}^l)$ of the solution $(\phi, \mathbf{u})$ of Eq. (4.1) on grid G^l is

$$\phi^l = \sum_i \phi_i^l \epsilon_i^l, \tag{4.6a}$$

$$\mathbf{u}^l = \sum_j u_j^l \mathbf{e}_j^l. \tag{4.6b}$$

The summation in (4.6a) is over all blocks Ω_i^l in grid G^l, and in (4.6b) over all edges E_j^l, not part of the Neumann boundary.

To discretize (4.2)–(4.3) we replace Λ by its discrete analogue Λ^l. To form the discrete equations we use $\tau = \epsilon_i^l$ for the test functions τ in (4.2a), and for $\mathbf{t}$ in (4.2b) we take $\mathbf{t} = \mathbf{e}_j^l$. Thus we obtain an algebraic system for $(\phi_l, \mathbf{u}_l)$, i.e., the vector of coefficients $\{\phi_i^l, u_j^l\}$:

$$\begin{pmatrix} 0 & A \\ A^T & W \end{pmatrix} \begin{pmatrix} \phi_l \\ \mathbf{u}_l \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}. \tag{4.7}$$

The matrix coefficients in this system are obtained by evaluation of the different integrals; however, we change the discretization by replacing the exact evaluation of the integrals, appearing in the elements of W, by a quadrature based on the four corners of each cell,

$$\int_{\Omega_i^l} a^{-1}(\mathbf{e}_j^l \cdot \mathbf{e}_k^l) d\Omega \simeq \sum_{\nu=1,4} \mathbf{e}_j^l(\mathbf{x}_\nu) \cdot \mathbf{e}_k^l(\mathbf{x}_\nu) \int_{(\Omega_i^l)_\nu} a^{-1} d\Omega, \tag{4.8}$$

where the cell Ω_i^l, with vertices $\mathbf{x}_\nu$, is subdivided into four equal pieces $(\Omega_i^l)_\nu$, as shown in Fig. 4.1. Because of (4.5), repeated use of this quadrature rule approximates W by a diagonal matrix, with elements

$$W_{kj} \simeq \delta_{kj} \left(\sum_{s=1,4} \int_{(\Omega_M^l)_s} a^{-1} d\Omega \right). \tag{4.9}$$

The summation in (4.9) is over the four small pieces $(\Omega_M^l)_s$ adjacent to edge E_M^l (see Fig. 4.1).

For Poisson's equation the coefficient a^{-1} appearing in (4.1b) is the constant ϵ^{-1}; so the relation between the displacement current $(J_\psi)_M^l$ at edge E_M^l (with length h_M) and the potentials ψ_L^l and ψ_R^l in the neighboring blocks Ω_L^l and

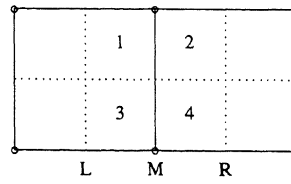

FIG. 4.1. Division of cells for approximation of elements of W.

Ω_R^l (cf. Fig. 4.1) is

$$(J_\psi)_M^l = \epsilon \frac{2h_M}{(a_L + a_R)} (\psi_R^l - \psi_L^l), \tag{4.10}$$

where a_L = area (Ω_L^l) and a_R = area (Ω_R^l).

For the continuity equations a^{-1} is an exponentially varying function. Although ψ^l approximates ψ as a piecewise constant function, we assume for the evaluation of (4.9) that ψ_l can be linearly interpolated between ψ_L^l and ψ_R^l. This leads to the Scharfetter–Gummel discretization of the continuity equations (cf. [2]):

$$(J_n)_M^l = \bar{\mu}_n \frac{2h_M}{(a_L + a_R)} \frac{-\alpha(\psi_R^l - \psi_L^l)}{e^{-\alpha\psi_R^l} - e^{-\alpha\psi_L^l}} (e^{-\alpha(\phi_n)_R^l} - e^{-\alpha(\phi_n)_L^l}), \tag{4.11a}$$

$$(J_p)_M^l = \bar{\mu}_p \frac{2h_M}{(a_L + a_R)} \frac{\alpha(\psi_R^l - \psi_L^l)}{e^{\alpha\psi_R^l} - e^{\alpha\psi_L^l}} (e^{\alpha(\phi_p)_R^l} - e^{\alpha(\phi_p)_L^l}). \tag{4.11b}$$

Dirichlet boundary conditions can be treated consistently by introducing cells with zero area at the boundary.

To treat green interfaces we use the Lagrangian multipliers $(\lambda_\phi)_j^l$, which are defined on edges E_j^l (cf. [17]). The Lagrangian multipliers are calculated by using discontinuous, piecewise linear test functions $\mathbf{t} \notin H^l(\mathrm{div}, \Omega^l)$ in the weak formulation of (4.1b). If quadrature rule (4.8) is used, and the integrals in (4.9) are approximated as before, we obtain (see Fig. 4.1)

$$(\lambda_\psi)_M^l = \frac{a_R\psi_L + a_L\psi_R}{a_L + a_R}, \tag{4.12a}$$

$$e^{-\alpha(\lambda_{\phi n})_M^l} = \frac{e^{-\alpha(\phi_n)_L^l}(e^{-\alpha\psi_R^l} - e^{-\alpha\psi_M^l}) + e^{-\alpha(\phi_n)_R^l}(e^{-\alpha\psi_M^l} - e^{-\alpha\psi_L^l})}{e^{-\alpha\psi_R^l} - e^{-\alpha\psi_L^l}}, \tag{4.12b}$$

$$e^{\alpha(\lambda_{\phi p})_M^l} = \frac{e^{\alpha(\phi_p)_L^l}(e^{\alpha\psi_R^l} - e^{\alpha\psi_M^l}) + e^{\alpha(\phi_p)_R^l}(e^{\alpha\psi_M^l} - e^{\alpha\psi_L^l})}{e^{\alpha\psi_R^l} - e^{\alpha\psi_L^l}}, \tag{4.12c}$$

with

$$\psi_M^l = (\lambda_\psi)_M^l.$$

These Lagrangian multipliers are an approximation of the solution at the edges (cf. [17]). So, at a green interface we calculate the Lagrangian multiplier on the finest grid on which the interface is not green and use this value as a Dirichlet boundary condition on the finer grids.

This concludes our discussion of the discretization of (4.2). By the use of our quadrature rule and the Slotboom variables we finally arrive at a discretization that is equivalent with the Scharfetter–Gummel scheme. The mixed finite element method is useful for the consistent construction of a nested set of discretizations on the different levels. In fact, the interpolation defined by (4.12) corresponds with the nonlinear interpolation introduced for the semiconductor equations in [11]. The sequence of discretizations is used to solve the discretized nonlinear system of equations by a multigrid algorithm.

5. MULTIGRID

The full approximation scheme [18] or nonlinear multigrid (NMG) scheme [21] is a multigrid iterative approach for solving sets of nonlinear equations obtained by discretization. For some classes of elliptic equations it is optimally efficient in the sense that the rate of convergence is independent of the mesh size. Another advantage is that large linear systems need to be neither solved nor stored. Generally, we write the discrete equations on grid G^l as

$$\mathfrak{R}^l(\bar{q}^l) = f^l, \tag{5.1}$$

where $\mathfrak{R}^l$ is the discretized nonlinear operator. Let q^l be an iterative approximation to $\bar{q}^l$. Better approximations can be obtained by classical relaxation methods (Jacobi, Gauss-Seidel, etc.), which reduce the residuals d^l,

$$d^l = f^l - \mathfrak{R}^l(q^l), \tag{5.2}$$

and, in particular, they efficiently damp the high-frequency components of the residuals. The low-frequency components are better reduced by solving the residual equation on a coarser grid G^{l-1}. Let q^{l-1} be some coarse grid approximation of q^l, then solve on grid G^{l-1}

$$\mathfrak{R}^{l-1}\tilde{q}^{l-1} = \mathfrak{R}^{l-1}q^{l-1} + \bar{R}_l^{l-1}d^l, \tag{5.3}$$

with $\bar{R}_l^{l-1}\colon \Lambda^l(\Omega^l) \to \Lambda^{l-1}(\Omega^{l-1})$, a restriction operator. A better approximation $\tilde{q}^l$ to $\bar{q}^l$ is then obtained by

$$\tilde{q}^l = q^l + P_{l-1}^l(\tilde{q}^{l-1}) - P_{l-1}^l(q^{l-1}), \tag{5.4}$$

with $P_{l-1}^l\colon \Lambda^{l-1}(\Omega^{l-1}) \to \Lambda^l(\Omega^l)$, a prolongation operator. Instead of solving (5.3) exactly, we approximate its solution either by a few iteration steps of a relaxation procedure or by a few cycles of the NMG procedure that makes

use of an even coarser grid. In this way the NMG algorithm is recursively defined.

If adaptive grids are used, the residual d^l is not necessarily computed everywhere on Ω; if a grid G^l does not exist in some area, the residual d^l is locally defined to be equal to zero.

As initial approximation q^{l-1} in the iterative process for the solution of (5.3) we do *not* use a restriction of a solution on a finer grid, as described in [18], but we take the last available iterand on the coarse grid. Such iterands are always available, because initial approximations for a finer grid are produced by interpolation from some approximation earlier computed on a coarser grid. Details and modifications of the coarse grid correction will be described in Section 7, whereas in Section 8 the construction of initial estimates on coarse grids is treated.

6. Relaxation

Previous experience with the nonlinear Poisson equation (cf. [14]) indicated that an adapted five-point Vanka-type relaxation (cf. [19]) is a good candidate for a relaxation method. By this method, all cells are successively scanned, first in forward, later in backward lexicographical order, and for each cell Ω_i^l the three nonlinear equations (4.7) are solved for the potentials ϕ_i^l and the fluxes u_j^l corresponding with the four edges E_j^l of the cell Ω_i^l (see Fig. 6.1).

From this 15×15 system the fluxes u_j^l are eliminated by (4.10)–(4.11). The resulting nonlinear 3×3 system could be solved by Newton's method, but it is possibly ill conditioned, if the initial guess is too far from the solution. Gummel's iteration (where the three nonlinear equations are solved sequentially) appears to be a more robust method for solving the nonlinear systems, and robustness is enhanced by solving Poisson's equation exactly in each Gummel step. The continuity equations to be solved in Gummel's iteration are linear if expressed in Φ_n and Φ_p. However, to avoid calculations in Slotboom variables, we calculate corrections $d\phi_n^{(n)}$ and $d\phi_p^{(n)}$ to the quasi-Fermi potentials $\phi_n^{(n)}$ and $\phi_p^{(n)}$ as for Newton's method for each continuity equation,

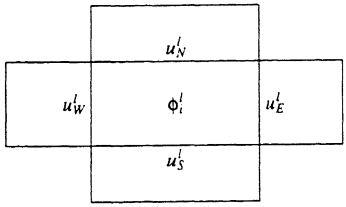

FIG. 6.1. Relaxation subdomain for five-point Vanka relaxation.

and then apply the correction transformation (see [2]):

$$\phi_n^{(n+1)} = \phi_n^{(n)} - \frac{1}{\alpha}\log(1 - \alpha d\phi_n^{(n)}), \tag{6.1a}$$

$$\phi_p^{(n+1)} = \phi_p^{(n)} + \frac{1}{\alpha}\log(1 + \alpha d\phi_p^{(n)}). \tag{6.1b}$$

Without rounding errors, this would solve the continuity equations in a single step; in practice a small number of iterations may be necessary. Large corrections may yield negative arguments for the logarithmic function. If this happens, we damp the correction by replacing the function $\log(x)$ in (6.1) by a $C^1(-\infty, \infty)$ function identical to $\log(x)$, for $x > x_0$. In practice, where the machine accuracy is 15 digits, we use (see [11])

$$\text{modlog}(x) = \log(x_0) + \text{sgn}(x)|\log(|x - x_0| + x_0) - \log(x_0)|, \tag{6.2}$$

with $x_0 = 0.5 \times 10^{-7}$.

In the following we describe how the local Poisson equation is efficiently solved by a modified Newton method. To simplify notation and without loss of generality, we write the Poisson equation, appearing in Gummel's iteration, as

$$a \sinh \bar{\psi} + b\bar{\psi} = 1, \tag{6.3}$$

with $a > 0$. In principle Eq. (6.3) is solved by Newton's method; however, if the Jacobian is dominated by the sinh function, it is better to linearize the equation with $\sinh \psi$ as a new variable. A suitable correction transformation strategy for the iterands $\psi^{(n)}$ in Newton's method, which switches between the two linearizations, is

$$\psi^{(n+1)} = \begin{cases} \text{arcsinh}(\sinh \psi^{(n)} + d\psi \cosh \psi^{(n)}), & \text{if } \left|\dfrac{a \cosh \psi}{b}\right| > 1, \\ \psi^{(n)} + d\psi^{(n)}, & \text{otherwise.} \end{cases} \tag{6.4}$$

The iteration is stopped if $|d\psi^{(n)}|$ is sufficiently small (less than 10^{-12}).

If the last available iterand is taken as the initial guess for Newton's method, we observe that large, untransformed corrections $d\psi^{(n)}$ may cause overflow. To avoid this situation the process is restarted with a better initial estimate as soon as an untransformed correction is too large ($|d\psi^{(n)}| > 1.0$ V). Two possible initial estimates for (6.3) are $\psi^{(0)} = \text{arcsinh}(1/a)$ and $\psi^{(0)} = 1/(a$

$+ b$). To judge the feasibility of these initial estimates, we use the fact that the solution $\bar{\psi}$ of (6.3) minimizes the convex functional

$$F(\psi) = a \cosh \psi + b \frac{\psi^2}{2} - \psi. \tag{6.5}$$

If the initial estimate $\psi^{(0)}$, for which F attains a minimal value, is chosen as the new initial estimate, Newton's method converges rapidly (within four steps in the cases we studied).

This concludes our description of the solution method for the small nonlinear systems appearing in five-point Vanka-type relaxation. To illustrate the robustness of this method, we use a 2D diode test problem (see Section 9), with either a forward biased (-1.0 V) or a reverse biased ($+5.0$ V) applied voltage. The performance of the relaxation process is shown in Table 6.1. Starting from a 4×4 grid, we perform two symmetric relaxation sweeps on every grid, before we interpolate the solution to a next finer grid. (Here, no coarse grid corrections are applied.) The finest grid used is a 64×64 grid. In Table 6.1 we show results for the cases where either Poisson's equation is solved exactly in each Gummel step or the solution of Poisson's equation is approximated by a single step from a Newton iteration, using the last available iterand as initial estimate. In both cases the Gummel iteration is stopped if

$$|d\psi^{(n)}| + |d\phi_n^{(n)}| + |d\phi_p^{(n)}| < 10^{-12}.$$

From Table 6.1 we see that the efficiency of Gummel's iteration is good, even in the forward biased case, in which the equations are strongly coupled.

TABLE 6.1
SOLUTION OF SMALL NONLINEAR SYSTEMS BY GUMMEL'S ITERATION

	Reverse bias		Forward bias	
	1 Newton step	Solve exact	1 Newton step	Solve exact
No. of processes	21.824	21.824	21.824	21.824
Mean No. of Gummel its.	2.8	2.8	4.2	4.1
Max No. of Gummel its.	9	8	9	9
Mean No. of steps for (6.3)	1.0	1.4	1.0	2.0
Max No. of steps for (6.3)	1	6	1	6
Divergent process	27	0	0	0

Note. A "process" is the solution of a 3×3 nonlinear system, by Gummel iteration. The "number of steps for (6.3)" is the number of Newton steps to solve Poisson's equation in Gummel's iteration. A process is divergent if Gummel's iteration does not converge within 25 steps.

Solving Poisson's equation exactly during each step does not improve the efficiency of Gummel's iteration, but robustness is enhanced indeed.

7. The Coarse Grid Correction

In this section the coarse grid correction, mentioned in Section 5, is discussed in more detail. The grid transfer operators P_{l-1}^{l} and $\bar{R}_{l}^{l-1}$ are introduced, and we explain why these simple transfer operators have to be adapted in regions where the solution exhibits sharp shifts.

The prolongation P_{l-1}^{l}, which transfers solutions from coarse to fine grids, is induced by the nesting of the spaces $\Lambda^{l-1}(\Omega^{l-1}) \subset \Lambda^{l}(\Omega^{l})$. This implies that any function $(\phi^{l-1}, u^{l-1}) \in \Lambda^{l-1}(\Omega^{l-1})$ can also be considered an element of $\Lambda^{l}(\Omega^{l})$, with a unique representation given by (4.6). The restriction operator $\bar{R}_{l}^{l-1}$, which transfers residuals from fine to coarse grids, is defined as the transpose of P_{l-1}^{l}.

If the coarse grid problem is solved exactly in (5.3), the errors, before and after the coarse grid correction, are related by

$$\tilde{q}^{l} - \bar{q}^{l} = [I_{l}^{l} - P_{l-1}^{l}(J^{l-1}(q^{l-1}))^{-1}\bar{R}_{l}^{l-1}J^{l}(q^{l})](q^{l} - \bar{q}^{l}) + O(\|\bar{R}_{l}^{l-1}d^{l}\|^{2}), \tag{7.1}$$

where

$$J^{l}(q^{l}) = \frac{\partial \Re^{l}(q^{l})}{\partial q^{l}} \tag{7.2}$$

is the Jacobian matrix and I the identity matrix.

As pointed out by De Zeeuw [13] for a one-dimensional case, the local value of the diagonal elements in the Jacobian matrix J^{l-1} and J^{l} can differ by orders of magnitude in the neighborhood of sharp layers, because q^{l-1} is not a good representation of q^{l}. From (7.1) we see that in these regions problems can be expected. If an element of J^{l-1} is much smaller than the corresponding elements of J^{l}, the error is locally blown up by the coarse grid correction. De Zeeuw proposed to damp the restricted residual in order to avoid such problems. Here we apply a similar technique for the two-dimensional case.

For every cell Ω_{i}^{l-1}, which is split into four cells Ω_{j}^{l}, we determine the damping factors $\theta_{i,k}^{l-1}$ by locally comparing the diagonal elements of the Jacobian matrices J^{l} and J^{l-1}:

$$\tilde{\theta}_{i,k}^{l-1} = \frac{|J_{(i,k)(i,k)}^{l-1}(q^{l-1})|}{\sup_{j=1,4}|J_{(j,k)(j,k)}^{l}(q^{l})|}, \qquad k = \psi, \phi_n, \phi_p, \tag{7.3a}$$

$$\theta_{i,k}^{l-1} = \min(2\tilde{\theta}_{i,k}^{l-1}, 1). \tag{7.3b}$$

The second step (7.3b) is added to avoid damping, if it is not necessary. If Ω_i^{l-1} is not split, we set $\theta_{i,k}^{l-1} = 1$.

By introduction of these damping factors the formulation of the coarse grid problem (cf. (5.3)) becomes

$$\mathfrak{N}^{l-1}\tilde{q}^{l-1} = \mathfrak{N}^{l-1}q^{l-1} + \Theta^{l-1}\bar{R}_l^{l-1}d^l, \tag{7.4}$$

where Θ^{l-1} is a diagonal matrix, with elements

$$\Theta_{(i,k)(i,k)}^{l-1} = \theta_{i,k}^{l-1}, \qquad k = \psi, \phi_n, \phi_p. \tag{7.5}$$

If the mesh becomes fine enough, sharp layers are well resolved, the coarse and fine grid Jacobians gain in similarity, and the damping disappears, as we see from (7.3).

However, only damping the restricted residual does not guarantee that there will locally be no spurious corrections to the fine grid solution, if the grids are relatively coarse. We also find it necessary to suppress the coarse grid correction locally, if layers are not properly resolved. In fact, we suppress the coarse grid correction from a cell Ω_i^{l-1}, split into four cells Ω_j^l, if

$$\sup_{j=1,4} |(2\psi_i^{l-1} - (\phi_n^{l-1})_i - (\phi_p^{l-1})_i) - (2\psi_j^l - (\phi_n^l)_j - (\phi_p^l)_j)| > 1.0. \tag{7.6}$$

This means that the correction is suppressed if the ratios (n/p) on the fine and the coarse grid are much different. In the context of the multigrid algorithm, the need for damping restricted residuals and suppressing coarse grid corrections can be understood as follows.

Locally the coarse grid solution is a bad representation of the fine grid solution, because the grids are too coarse. However, it is known that even very coarse grids still may help to reduce low-frequency error components. By locally damping the interaction between the grids, we are still able to reduce these low-frequency error components in some parts of the solution, without exciting high-frequency error components in other parts. If necessary, additional local relaxation can reduce errors in regions where the interaction between the grids is affected by damping; in our numerical experiments, however, this does not influence the observed convergence behavior.

8. The Initial Estimate

To start the multigrid algorithm, we first have to compute a solution on the coarsest grid. Initial estimates on finer grids are obtained by interpolation

from a coarser one. On the coarsest grid, we use a continuation strategy for the applied voltages at the contacts.

Starting at a voltage that yields a simple problem (e.g., zero voltage at all contacts), we change the boundary condition stepwise to its final value. On the coarse grid moving from one applied voltage to the next, we take the following steps: (i) change boundary conditions; (ii) find an initial approximation for these new boundary conditions; and (iii) improve this approximation iteratively. The iterative improvement of the coarsest grid approximations is done by relaxation only (see Section 6), which is robust and easy to implement.

The initial approximation for the new boundary conditions is obtained by a technique due to Mole and co-workers [20]. Starting from a solution $(\psi^{(0)}, \phi_n^{(0)}, \phi_p^{(0)})$, we first assume that the carrier densities do not change during continuation, and solve the following equations for the corrections $(d\phi_n, d\phi_p)$,

$$-\nabla \cdot (d\mathbf{J}_n) = 0, \tag{8.1a}$$

$$-\nabla \cdot (d\mathbf{J}_p) = 0, \tag{8.1b}$$

with

$$d\mathbf{J}_n = -\bar{\mu}_n e^{\alpha(\psi^{(0)} - \phi_n^{(0)})} \nabla(\alpha d\phi_n), \tag{8.1c}$$

$$d\mathbf{J}_p = -\bar{\mu}_p e^{\alpha(\phi_p^{(0)} - \psi^{(0)})} \nabla(\alpha d\phi_p), \tag{8.1d}$$

where the change in the applied voltage is used for the boundary conditions. The linear equations (8.1) are discretized by the mixed finite element method as described in Section 4. The resulting system is solved iteratively by Vanka relaxation; this iteration is stopped if the largest correction is a factor 10^{-2} less then the change in the applied voltage.

Next, the initial approximation $(\psi^{(1)}, \phi_n^{(1)}, \phi_p^{(1)})$ is found by setting

$$\phi_n^{(1)} = \phi_n^{(0)} + d\phi_n,$$

$$\phi_p^{(1)} = \phi_p^{(0)} + d\phi_p,$$

and $\psi^{(0)}$ is updated in such a way that the density of the majority charge carries does not change, i.e.,

$$\psi^{(1)} = \psi^{(0)} + d\phi_n, \qquad \text{in a } n \text{ region},$$

$$\psi^{(1)} = \psi^{(0)} + d\phi_p, \qquad \text{in a } p \text{ region}.$$

In exceptional cases, with a forward biased diode problem, we observed that the new minority level may temporarily become larger than the new majority

level. However, this caused no problems, because of the robustness of our relaxation procedure.

9. Numerical Experiments

We use a 2D diode problem as a test problem for our adaptive multigrid algorithm. The convergence behavior for uniform grids is shown in Section 9.1, and the power of local refinement is demonstrated in Section 9.2.

The problem is defined on a square $[0, 10^{-3}] \times [0, 10^{-3}]$. The doping profile D describes a quarter-circle n-region diode (see Fig. 9.1):

$$D(\mathbf{x}) = \begin{cases} +10^{18}, & \|\mathbf{x}\| < 0.5 \times 10^{-3}, \\ 0, & \|\mathbf{x}\| = 0.5 \times 10^{-3}, \\ -10^{18}, & \|\mathbf{x}\| > 0.5 \times 10^{-3}. \end{cases} \tag{9.1}$$

At the two contacts, indicated in Fig. 9.1 by double lines, the quasi-Fermi potentials ϕ_n and ϕ_p are given, depending on the applied voltage V_a,

$$\phi_n = \phi_p = \begin{cases} 0, & y = 0, x < 0.25 \times 10^{-3}, \\ V_a, & y = 10^{-3}, \end{cases} \tag{9.2}$$

and ψ is derived from these values, by assuming charge neutrality,

$$p - n + D = 0. \tag{9.3}$$

At the remaining parts of the boundary homogeneous Neumann boundary conditions are assumed for all three equations.

We consider two test problems: a reverse biased ($V_a = +5.0$ V) and a forward biased problem ($V_a = -1.0$ V). The numerical values for the constants appearing in (2.1) and (2.4) are $\epsilon = 1.036 \times 10^{-12}$, $n_i = 1.22 \times 10^{10}$, $q = 1.60 \times 10^{-19}$, and $\alpha = 38.683$.

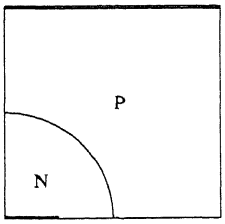

FIG. 9.1. Quarter-circle diode.

9.1. *Uniform Grids*

In this subsection the convergence behavior of the multigrid algorithm is studied for the two problems on uniform grids. The coarsest grid used in the calculations was a 4×4 grid. The solution of this very small coarse grid

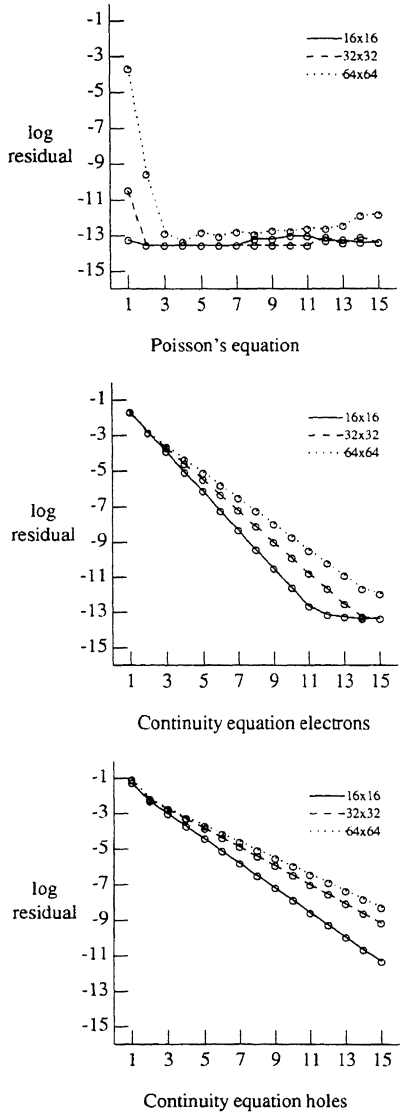

FIG. 9.2. Convergence behavior, reverse biased diode (V cycles).

problem is approximated by executing 50 relaxation sweeps, thus reducing the residual by a factor of 10^{-5}. In all multigrid cycles a single symmetric relaxation sweep is made both before and after the coarse grid correction.

For the reverse biased problem, the convergence behavior for different meshes is shown in Fig 9.2 (V cycles) and Fig. 9.3 (W cycles). The convergence

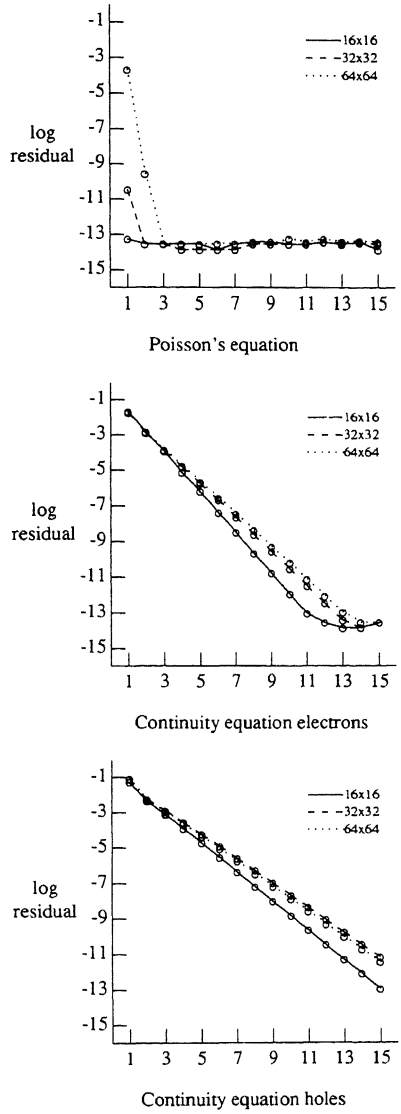

FIG. 9.3. Convergence behavior, reverse biased diode (W cycles).

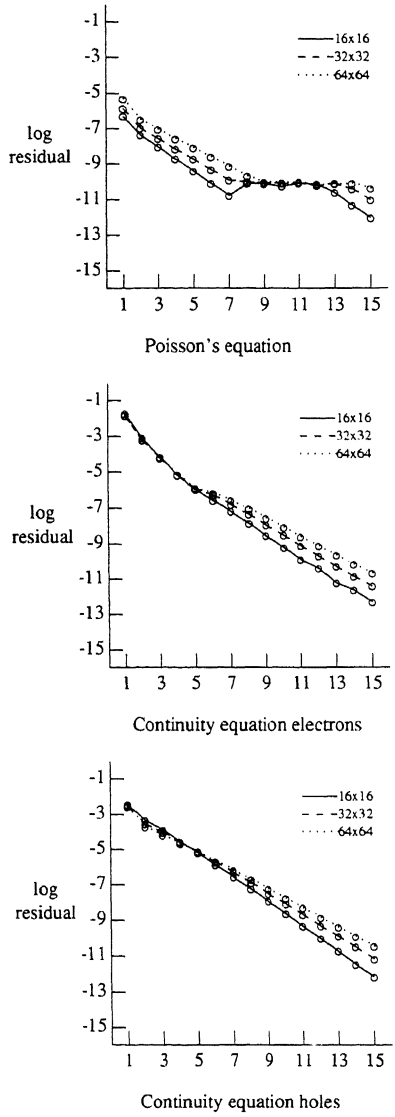

FIG. 9.4. Convergence behavior, forward biased diode (V cycles).

is measured by the sup-norm of the residual, which is scaled by the corresponding diagonal element of the Jacobian. In both cases it appears that Poisson's equation is solved up to machine precision in only a few cycles. If W cycles are used we find a nearly grid-independent convergence behavior.

Figures 9.4 and 9.5 show the convergence behavior for the forward biased

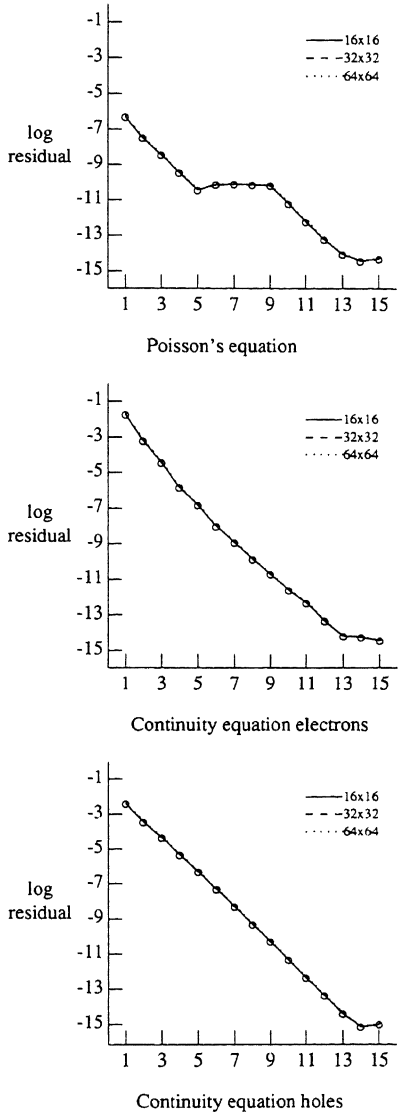

FIG. 9.5. Convergence behavior, forward biased diode (W cycles).

problem, using V and W cycles. The convergence behavior for Poisson's equation looks irregular; it stalls until the continuity equations are solved sufficiently accurate. Again, we find a nearly grid-independent convergence behavior for W cycles.

TABLE 9.1
DAMPING OF INTERACTION BETWEEN GRIDS FOR THE REVERSE BIASED DIODE

Grid	Cells with damping of the restricted residual	Cells with suppression of the correction
4×4	6 (=38%)	1 (=6%)
8×8	10 (=16%)	4 (=6%)
16×16	16 (=6%)	7 (=3%)
32×32	28 (=2%)	15 (=1%)

Finally, in Table 9.1 we see that the interaction between the grids is damped only in a small percentage of the cells. This number decreases if the mesh gets finer. Damping only occurs in the reverse biased problem. This concludes our discussion of results obtained for uniform grids. We find a good, nearly grid-independent, convergence behavior, by locally damping the interaction between the grids.

9.2. *Nonuniform Grids*

Here we show results for calculations on a locally adapted grid. Because well-analyzed a posteriori error estimators are not yet available for the MFEM applied to semiconductor equations, we use an ad hoc refinement criterion, viz. equidistribution of the second derivative of the electrostatic potential ψ. In fact, a cell Ω_i^l (with area a_i^l) is split if

$$a_i^l(|\psi_{i,xx}^l| + 2|\psi_{i,xy}^l| + |\psi_{i,yy}^l|) > \eta, \tag{9.4}$$

for some a priori given constant η. The second-order derivatives in (9.4) are approximated numerically on grid G^l by means of standard three- or nine-point stencils.

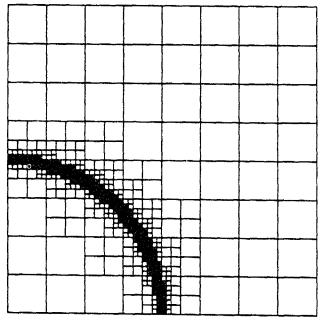

FIG. 9.6. Self-adapted grid for reverse biased diode.

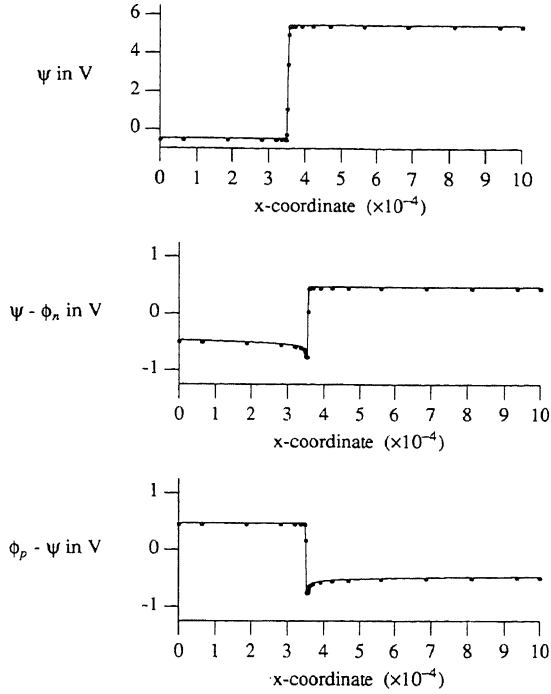

FIG. 9.7. Plot of solution components along diagonal.

As test problem we use the reverse biased ($V_a = +5.0$ V) diode, and take $\eta = 2.5 \times 10^{-2}$. Figure 9.6 shows the final mesh; the finest level corresponds to a uniform 512×512 grid. Indeed, the cells are concentrated in the neighborhood of the junction, where all three solution components have a sharp

TABLE 9.2
NUMBER OF CELLS IN ADAPTIVE GRID

Level	Number of cells in adaptive grid	Uniform grid
0	16	4×4
1	32	8×8
2	64	16×16
3	128	32×32
4	256	64×64
5	512	128×128
6	1080	256×256
7	2656	512×512

interior layer. In Fig. 9.7 a cross section of the solution components along the diagonal $x = y$ is shown. We obtain a good resolution of the interior layer by a limited number of cells, as can be seen from Table 9.2, which gives the number of cells on different levels. As long as the coarser meshes (mesh size h) are unable to resolve the sharp layer, we see that the number of cells is $O(h^{-1})$. Only for finer meshes are more cells introduced. So, by using local refinement, we are able to get a good resolution of the layer, with a restricted number of cells. Because the discrete equations are solved by a multigrid method, our algorithm is highly efficient.

REFERENCES

1. P. A. Markowich, *The Stationary Semiconductor Device Equations.* Springer-Verlag, Wien/New York (1986).
2. S. J. Polak, C. den Heijer, W. H. A. Schilders, and P. Markowich, Semiconductor device modeling from the numerical point of view. *Internat. J. Numer. Math. Engrg.* **24,** 763–838 (1987).
3. J. J. H. Miller (Ed.), *Proceedings, Sixth International NASECODE Conference.* Boole Press, Dublin (1989).
4. G. Baccarani and M. Rudan (Eds.), *Proceedings, 3rd International Conference on Simulation of Semiconductor Devices and Processes,* Vol. 3, *Simulation of Semiconductor Devices and Processes.* University of Bologna (1988).
5. Q. M. Shiekh, Systems of nonlinear algebraic equations arising in simulation of semiconductor devices. Report UIUCDCS-R-83-1133, Urbana, Illinois (1983).
6. A. S. L. Shieh, Solution of coupled systems of PDE by the transistorized multi-grid method. *In Proceedings, Conference on Numerical Solution of VLSI devices, Boston* (1984).
7. R. E. Bank and H. D. Mittelmann, Continuation and multi-grid for nonlinear elliptic systems. *In* W. Hackbusch and U. Trottenberg (Eds.), *Multigrid Methods,* Vol. II, pp. 23–27. Springer-Verlag, New York/Berlin (1986).
8. I. Babuska and W. C. Rheinboldt, Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.* **15,** 736–754 (1978).
9. J. T. Oden, Progress in adaptive methods in computational fluid dynamics. *In* J. E. Flaherty, P. J. Paslow, M. S. Shephard, and J. D. Vasilakis (Eds.), *Adaptive Methods for Partial Differential Equations.* SIAM, Philadelphia, PA (1989).
10. G. H. Schmidt and F. J. Jacobs, Adaptive local grid refinement and multi-grid in numerical reservoir simulation. *J. Comput. Phys.* **77,** 140–165 (1988).
11. P. W. Hemker, A nonlinear multigrid method for one-dimensional semiconductor device simulation. *In* Guo Ben Yu, J. J. H. Miller, and Shi Zhong-Ci (Eds.), *BAIL V, Proceedings, 5th International Conference on Boundary and Interior Layers.* Boole Press, Dublin (1988).
12. P. W. Hemker, A nonlinear multigrid method for one-dimensional semiconductor device simulation: Results for the diode. *J. Comput. Appl. Math.* **30,** 117–126 (1990).
13. P. M. De Zeeuw, Nonlinear multigrid applied to a 1D stationary semiconductor model. Report NM-R8905, Department of Numerical Mathematics, Centre for Mathematics and Computer Science, Amsterdam (1989).
14. J. Molenaar, Non-linear multigrid in 2-D semiconductor device simulation: The zero current

case. Report NM-R8917, Department of Numerical Mathematics, Centre for Mathematics and Computer Science, Amsterdam (1989).

15. W. C. Rheinboldt, On a theory of mesh-refinement processes. *SIAM J. Numer. Anal.* **17**, 766–778 (1980).
16. P. A. Raviart and J. M. Thomas, A Mixed finite element method for second order elliptic problems. *In Mathematical Aspects of the Finite Element Method,* Lecture Notes in Mathematics, Vol. 606. Springer-Verlag, New York/Berlin (1977).
17. D. N. Arnold and F. Brezzi, Mixed and non-conforming finite element methods: Implementation, post-processing and error estimators, *MMAN* **19**, 7–32 (1985).
18. A. Brandt, Guide to multigrid development. *In Multigrid Methods,* Lecture Notes in Mathematics, Vol. 960. Springer-Verlag, New York/Berlin (1982).
19. S. P. Vanka, Block-implicit multigrid solution of Navier–Stokes equations in primitive variables. *J. Comput. Phys.* **65**, 138–158 (1986).
20. S. P. Edwards, A. M. Howland, and P. J. Mole, Initial guess strategy and linear algebra techniques for a coupled two-dimensional semiconductor equation solver. *In Proceedings, NASECODE IV,* pp. 272–280. Boole Press, Dublin (1985).
21. W. Hackbusch, *Multigrid Methods and Applications,* Springer Series in Computational Mathematics, Vol. 4, Springer-Verlag, Berlin (1985).