Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

B.P. Sommeijer, W.H. Hundsdorfer, C.T.H. Everaars, P.J. van der Houwen, J.G. Verwer

A numerical study of a 1D stationary semiconductor model

Department of Numerical Mathematics Note NM-N8702 May

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

A Numerical Study of a 1D Stationary Semiconductor Model

B.P. Sommeijer, W.H. Hundsdorfer,
C.T.H. Everaars, P.J. van der Houwen, J.G. Verwer
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Based on a 1D model describing the stationary basic semiconductor device equations, several numerical methods have been investigated to solve this type of problems. The main part of this work discusses the solution of the nonlinear systems which result from discretization of the PDEs. Emphasis is placed upon finding suitable initial approximations when iterative processes are employed.

1980 Mathematics Subject Classification: 65N20, 65H10

Note: Contract research carried out by order of Drs. S.J. Polak, Nederlandse Philips Bedrijven, ISA-ISC-TIS/CARO, Gebouw SAQ 2, 5600 MD Eindhoven, The Netherlands

1. Introduction

Parallel to the interest in semiconducter device simulation, the number of papers on this subject has shown an explosive growth during the last decades. Due to the never-ending demand for more speed in technical environments, the nowadays computers have made a tremendous progress and, consequently, allow for a rather sophisticated numerical simulation of semiconductors. Even the treatment of (transient) three-dimensional problems has started recently.

However, in these solution processes, not all the elements are completely understood and for some elements it is not clear in advance which solution technique is to be preferred.

On the basis of a *concrete* 1D stationary example problem (as commissioned by Philips) we aim to provide insight into choosing a proper algorithm for the various elements in the solution process.

Although the real (3D) situation is drastically simplified in this 1D model, it still possesses many of the characteristic difficulties, usually encountered in the numerical simulation of semiconductor devices.

After discretizing the elliptic PDE, the solution of the resulting nonlinear system of equations will be discussed. Here, our starting point is Newton's method. In spite of its quadratically convergence behaviour for sufficiently close starting guesses, this method has the disadvantage of being not globally convergent. It is this last property which will obtain special attention in this report; in other words, we will investigate techniques to provide 'sufficiently accurate' initial approximations to let the Newton process converge. Hereby, we note that these techniques are tailored to the particular semiconductor problem at hand.

Finally, it is emphasized that much of the material presented in this report is already known, and extensively discussed in the expository paper by Polak et al. [8]. New aspects can be found in Section 4.1 (Continuation type of methods) and in Section 4.2 (Multiple grid methods).

Report NM-N8702
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

2. The basic equations

The basic mathematical model describing the behaviour of a semiconductor is given by [6,9]

$$-\operatorname{div}(\epsilon \nabla \psi) = q(p - n + D), \tag{2.1a}$$

$$\operatorname{div} J_p = -qR\,, \tag{2.1b}$$

$$\operatorname{div} J_n = qR\,. \tag{2.1c}$$

In (2.1a) (Poisson's equation), ψ represents the electrostatic potential, ϵ denotes the permittivity, which is assumed to be a scalar constant, q is the elementary charge and p and n represent the concentration of holes and electrons, respectively. The (doping) function D will be specified in the next subsection.

The continuity equations (2.1b) and (2.1c) describe the relations between the current densities J_p (for holes) and J_n (for electrons) and a function $R = R(p,n)$ which is called the recombination-generation rate.

The current densities J_p and J_n are defined by the so-called current relations

$$J_p = -q\mu_p\left(\frac{1}{\alpha} \nabla p + p \nabla \psi\right), \tag{2.2a}$$

$$J_n = q\mu_n\left(\frac{1}{\alpha} \nabla n - n \nabla \psi\right), \tag{2.2b}$$

where α is a constant and μ_p and μ_n are the mobilities of holes and electrons.

In the set of equations (2.1)-(2.2), the quantities $\epsilon, q, D, R, \alpha$ are prescribed and p, n and ψ represent the unknowns.

2.1. A 1D model problem

We will confine our numerical study to the one-dimensional version of {(2.1), (2.2)}. The specification of the problem is based on a note which we received from Dr. W.H.A. Schilders, Philips; June 3, 1986.

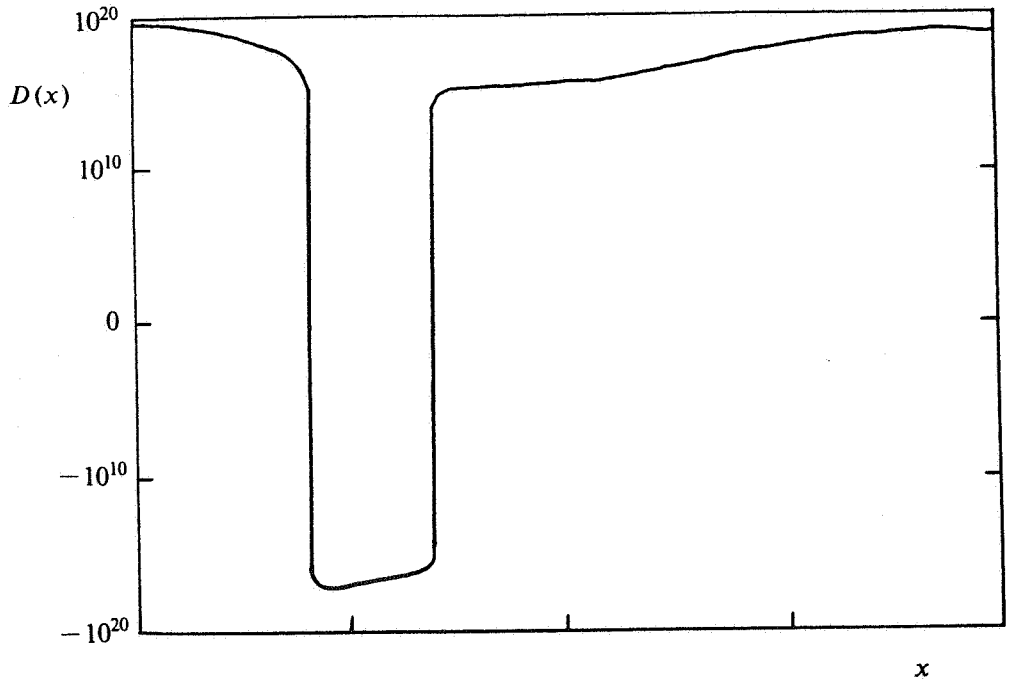
The data used are given by

$$\begin{aligned} &\epsilon = 1.035918_{10}{}^{-12}\,,\ q = 1.6021_{10}{}^{-19};\\ &\mu_p = \mu_n = 500;\\ &\alpha = q/kT,\ k = 1.38054_{10}{}^{-23}\,,\ T = 300;\\ &R = \frac{pn - n_i^2}{\tau(p + n + 2n_i)}\,,\ \tau = {}_{10}{}^{-6}\,,\ n_i = 1.22_{10}{}^{10}. \end{aligned} \tag{2.3}$$

The doping function D is specified by

$$\begin{aligned} D(x) = {} & 6_{10}{}^{15} + 6_{10}{}^{19} \exp\{-(x/7.1_{10}{}^{-5})^2\}\\ & - 2.15_{10}{}^{18} \exp\{-(x/1.15_{10}{}^{-4})^2\}\\ & + 1.1_{10}{}^{19} \exp\{-((x - 8_{10}{}^{-4})/1.3_{10}{}^{-4})^2\}\,, \end{aligned} \tag{2.4}$$

where $x \in I = [0, 8_{10}{}^{-4}]$, the interval on which the problem is defined. In the following figure the strongly varying behaviour of the doping function $D(x)$ is shown.



Doping profile (2.4)

2.2. Boundary conditions

In general, we have three contacts to a semiconductor device, called the emitter (E), the basis (B) and the collector (C) (see Figure 1). In our case, the boundary conditions are given by

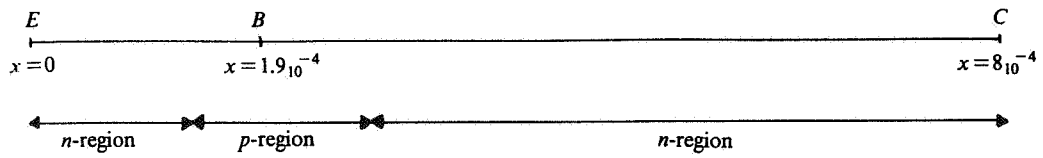


FIGURE 1. Definition of the contacts in a 1D transistor problem: A p-region (n-region) is a domain with a high concentration of holes (electrons) (see also the figure of the dope function).

Emitter: vanishing space charge, i.e. $p-n+D=0$,
$$J_p = 0, \qquad \phi_n = V_E; \tag{2.5a}$$

Basis:
$$\phi_p = 0; \tag{2.5b}$$

Collector: vanishing space charge, i.e. $p-n+D=0$,
$$\phi_p = V_C, \qquad \phi_n = V_C, \tag{2.5c}$$

where V_E and V_C are given voltages, respectively applied at the emitter and collector, and the functions ϕ_p and ϕ_n are defined by

$$\begin{aligned} \phi_p &= \psi + \frac{1}{\alpha}\ln(p/n_i), \\ \phi_n &= \psi - \frac{1}{\alpha}\ln(n/n_i). \end{aligned} \tag{2.6}$$

The functions ϕ_p and ϕ_n are usually termed 'the hole and electron quasi-Fermi potentials'. These functions play an important role in the *numerical* solution of the problem.

2.3. Change of variables

For the problem {(2.1), (2.2)} it is well known, that the dependent variables p and n assume values in a very wide range (e.g. $[10^{-1},10^{19}]$) and possess a strongly exponential behaviour. In numerical simulation, this may cause serious problems.

Therefore, we decided to perform our investigations in terms of the variables $\{\psi,\phi_p,\phi_n\}$, the latter being defined in (2.6). This has the additional advantage that the applied voltages V_E and V_C are simple Dirichlet conditions for the ϕ-variables (cf. (2.5)).

The range of values assumed by these new variables depends on V_E and V_C, but is rather modest in a typical case, e.g. [-1,1].

However, it should be observed that, by the change of variables, the nonlinearity of the problem to be solved is strongly increased.

Fortunately, it will turn out that the advantages of the (p,n)-variables (viz. a mildly nonlinear operator) and those of the ϕ-variables (viz. a modest range of values) can be combined (see [8] and also Section 4.1.3, Correction transformation).

2.4. Summary of the problem definition

In terms of the variables ψ,ϕ_p and ϕ_n, our 1D test model is given by

$$\epsilon\psi'' + qn_i(e^{\alpha(\phi_p-\psi)} - e^{\alpha(\psi-\phi_n)}) + qD = 0\,, \tag{2.7i}$$

$$\mu_p n_i(\phi_p' e^{\alpha(\phi_p-\psi)})' - R = 0\,, \tag{2.7ii}$$

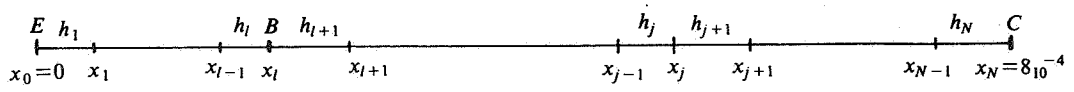
$$\mu_n n_i(-\phi_n' e^{\alpha(\psi-\phi_n)})' - R = 0\,. \tag{2.7iii}$$

Apart from the boundary condition $J_p=0$ (at the emitter), which is replaced by $\phi_p'=0$, all other boundary conditions are straightforwardly adapted.

3. The spatial discretization

To discretize the equations (2.7), we define the nonequidistant grid

$$\Delta_h := \{x_j\colon x_0 = 0,\ x_j = x_{j-1}+h_j,\quad j = 1,\ldots,N-1,\ x_N = 8_{10}{}^{-4}\}\,.$$



Furthermore, we introduce the abbreviations ψ_j, ϕ_j^+ and ϕ_j^-, respectively denoting approximations to $\psi(x_j)$, $\phi_p(x_j)$ and $\phi_n(x_j)$.

Now, we can setup the system of nonlinear equations, henceforth shortly written as

$$\mathbf{F}(\mathbf{U}) = \mathbf{0}\,, \tag{3.1}$$

with $\mathbf{F}=(F_{1,j},F_{2,j},F_{3,j})_j^T$ and $\mathbf{U}=(\psi_j,\phi_j^+,\phi_j^-)_j^T$, where j runs through the grid points (i.e., $j=0,1,...,N$) and the $F_{i,j}(i=1,2,3)$ correspond to the equations (2.7i), (2.7ii) and (2.7iii), respectively.

The discretization of Poisson's equation (2.7i) is standard and will be performed by the usual three-point difference formula. This results in

$$F_{1,j} := \epsilon\left(\frac{\psi_{j+1}-\psi_j}{h_{j+1}} - \frac{\psi_j-\psi_{j-1}}{h_j}\right) + q\cdot\frac{h_j+h_{j+1}}{2}\left(n_i e^{\alpha(\phi_j^+-\psi_j)} - n_i e^{\alpha(\psi_j-\phi_j^-)} + D(x_j)\right) = 0\,, \quad j=1,...,N-1. \tag{3.2a}$$

For the Poisson equation the boundary conditions are converted into

$$F_{1,j} := n_i e^{\alpha(\phi_j^+-\psi_j)} - n_i e^{\alpha(\psi_j-\phi_j^-)} + D(x_j) = 0\,, \quad j=0,N. \tag{3.2b}$$

The discretization of the continuity equations (2.7ii) and (2.7iii) needs special care. As pointed out in [6, p. 141] the standard finite difference approximation does not provide reasonable approximations, unless the mesh sizes are very severely restricted.

To circumvent this undesirable situation, we applied a nowadays common discretization technique, introduced by Scharfetter and Gummel, which is based on exponential fitting. For a detailed discussion of this method, we refer to [9, p. 158] or [6, p. 148].

Using this technique we arrive at

$$F_{2,j} := \frac{\mu_p}{\alpha} e^{\alpha(\phi_j^+-\psi_j)}\left[\frac{B(\alpha(\psi_{j+1}-\psi_j))E(\alpha(\phi_{j+1}^+-\phi_j^+))}{h_{j+1}} + \frac{B(-\alpha(\psi_j-\psi_{j-1}))E(-\alpha(\phi_j^+-\phi_{j-1}^+))}{h_j}\right] - \frac{h_{j+1}+h_j}{2\tau}\frac{e^{\alpha(\phi_j^+-\phi_j^-)}-1}{e^{\alpha(\phi_j^+-\psi_j)}+e^{\alpha(\psi_j-\phi_j^-)}+2}\,, \quad j=1,...,l-1,l+1,...,N-1. \tag{3.3a}$$

The Neumann condition at the emitter is simply discretized as

$$F_{2,0} := \phi_0^+ - \phi_1^+ = 0 \tag{3.3b}$$

and the Dirichlet conditions at the basis and collector yield

$$F_{2,l} := \phi_l^+ = 0 \quad \text{and} \quad F_{2,N} := \phi_N^+ - V_C = 0. \tag{3.3c}$$

The function B, occurring in (3.3a), is the Bernoulli function defined by

$$B(z) := \frac{z}{e^z-1} \tag{3.4}$$

and the function E is given by

$$E(z) := e^z-1. \tag{3.5}$$

It should be remarked that, for small values of z, a straightforward evaluation of B and E may yield cancellation or even zero-division. Therefore, in our implementation, we set

$$B(z) = 1-\frac{z}{2}\left(1-\frac{z}{6}\left(1-\frac{z^2}{60}\right)\right) \tag{3.4'}$$

and

$$E(z) = z(1+\frac{z}{2}(1+\frac{z}{3}(1+\frac{z}{4}))) \tag{3.5'}$$

if $|z|<10^{-4}$.

Finally, the discretization of the second continuity equation (2.7iii) is very similar to (3.3a), with the exception that now the point x_l (i.e. the basis) is not excluded from the general formula. Hence, we find

$$F_{3,j} := \frac{\mu_n}{\alpha} e^{\alpha(\psi_j-\phi_j^-)} \left[\frac{B(-\alpha(\psi_{j+1}-\psi_j))E(-\alpha(\phi_{j+1}^- - \phi_j^-))}{h_{j+1}} + \frac{B(\alpha(\psi_j-\psi_{j-1}))E(\alpha(\phi_j^- - \phi_{j-1}^-))}{h_j} \right] - \frac{h_{j+1}+h_j}{2\tau} \frac{e^{\alpha(\phi_j^+ - \phi_j^-)}-1}{e^{\alpha(\phi_j^+ - \psi_j)} + e^{\alpha(\psi_j - \phi_j^-)} + 2}, \quad j=1,...,N-1 \tag{3.6a}$$

and

$$F_{3,0} := \phi_0^- - V_E = 0 \quad , \quad F_{3,N} := \phi_N^- - V_C = 0 . \tag{3.6b}$$

The system (3.1) is now completely defined by assembling the equations (3.2)-(3.6).

4. Numerical algorithms

In this section we will discuss several possible ways to solve the system (3.1) numerically. First, we remark that due to the nonlinearity of the problem, any applicable method must be of the iterative type [7]. This leads us directly to the most famous iterative scheme, i.e. Newton's method, defined by

$$U^{n+1} = U^n - [F'(U^n)]^{-1}F(U^n), \quad n=0,1,..., \tag{4.1}$$

with U^0 given. Note that vector symbols are omitted, as no ambiguity has to be feared.

Usually, the problem (3.1) has to be solved for several sets of applied voltages; let us say, for $\{V_{E_j}, V_{C_j}\}, j=0,1,....$ A straightforward approach then is to start up Newton's method for solving the problem for $\{V_{E_j}, V_{C_j}\}$, with the solution obtained for $\{V_{E_{j-1}}, V_{C_{j-1}}\}$. Unfortunately, the nature of semiconductor problems is such that a small change in $\{V_E, V_C\}$ may result in a dramatic change in the solution. As a consequence, the approach often results in a divergent process, as Newton's method is not globally convergent. Therefore, the main part of this section is devoted to finding close enough initial approximations.

4.1. Continuation type of methods

Suppose we have a solution U^0 of (3.1) corresponding to the set of boundary conditions $\{V_E^0, V_C^0\}$ and we want to find the solution U^* corresponding to $\{V_E^*, V_C^*\}$. For that purpose, we introduce the continuation parameter t and define the family of problems

$$F(U(t),t) = 0, \qquad t\in[0,1] \tag{4.2}$$

where it is assumed that (4.2) has a unique solution for each $t\in[0,1]$. The parameter t should hereby be interpreted as a continuation of the discrete boundary values. We shall take $U(t)$ to be the solution of (3.1) with boundary conditions $\{(1-t)V_E^0+tV_E^*, (1-t)V_C^0+tV_C^*\}$, so that $U(0)=U^0$ and $U(1)=U^*$. In this way we have defined a solution curve which gradually leads to U^*.

Remark. In this connection we remark that the continuation parameter t is not necessarily restricted to the boundary values. It can also be associated with other problem-parameters. For example,

quantities which are responsible for (the location of) the sharp gradients in the solution. These possibilities are not considered in this report but may be subject of further research. Here we restrict our attention to the continuation of the boundary values.

4.1.1. Davidenko's method

One possible way to define the solution curve is to transform (4.2) into an initial value problem by differentiating to t, yielding

$$F_t(U(t),t) + F_u(U(t),t)\dot{U}(t) = 0, \quad t\in[0,1]$$
$$U(0) = U^0. \tag{4.3}$$

If we assume that F_u is invertible, then (4.3) can be written as an explicit ODE which can be integrated by standard ODE methods. It should be observed, that in our example - where t is only involved in the boundary conditions - this approach results in an F_t-vector with nonvanishing elements in the first and last component only.

If the ODE (4.3) is *exactly* solved, then we do find the exact solution U^*. However, a numerical integration to obtain a solution of sufficiently high accuracy, would result in a very time-consuming process. Therefore, we approximately solve this ODE and regard the numerical solution at $t=1$ as a -hopefully- better initial approximation for the Newton process (4.1).

We actually implemented this approach for a simplified set of equations and applied various standard ODE codes based on explicit time integration techniques, using several tolerances to control the local errors. The result of these experiments were not very promising, in the sense that a more stringent local error criterion did not systematically yield an initial approximation which resulted in a better convergence behaviour for Newton's method.

On the basis of these experiments we decided to reject this type of continuation methods.

4.1.2. Discrete imbedding

A different way to find $U^*(=U(1))$ is successively approximate the solutions of

$$F(U(t_j),t_j) = 0, \quad j = 0,1,2,... \tag{4.2'}$$

for a sequence of points $t_0=0,t_1,t_2,...,1$. This type of continuation methods is usually called *discrete imbedding.* For each of the problems in (4.2'), in principle, any iterative method can be used. We will consider Newton's method for this purpose.

In this subsection we will describe a strategy which automatically selects the sequence of points $t_j(j>1)$ for each of which the Newton process converges, starting with the result of the previous point. Thus, the crucial task is to select stepsizes Δt such that the strategy performs best in terms of total number of iterations. A theoretical discussion on this topic can be found in [4]. For the development of such a strategy, we introduce a few criteria:

A. *divergence criterion:* first, we need a robust mechanism which immediately signals divergence in the iteration process; this is absolutely necessary, since iterating with unrealistic values - for example, obtained from a too large Newton-correction - will very quickly result in overflow.
   To avoid this situation, we decided to introduce an allowed range for each of the variables $\psi,\phi^+(=\phi_p)$ and $\phi^-(=\phi_n)$. That is, we provide minimal and maximal values for these variables and the algorithm ensures to keep all iterates within these ranges. As an indication, a reasonable range is provided by twice the extreme values the variables are expected to assume. (However, we have experienced that the performance of the algorithm does not critically depend on these values).
   Henceforth, these ranges will be called the *bandwidth.*

B. *stopping criterion:* in any iterative process, we continuously have to answer the question: "have we ground to a halt?". It is an essential part of our algorithm to specify different "grounds", depending on the value of t. Only for $t=1$, i.e. when (4.2) coincides with the original problem, we impose a stringent stopping criterion; for $t<1$, we are iterating on artificially introduced

problems, the "solution" of which merely serves to obtain a close enough initial approximation for the next problem. Therefore, at these intermediate points we use a stopping criterion which is far less stringent, thus saving iterations. The actual stopping criterion, we implemented, reads

$$\| \text{absolute difference between two successive iterates} \|_\infty < \text{TOL}(t) \tag{4.4a}$$

where

$$\text{TOL}(t) = \begin{cases} 10^{-2} & \text{if } t<1 \\ 10^{-11} & \text{if } t=1. \end{cases} \tag{4.4b}$$

We remark that it is not necessary to perform a *relative* test on the difference of the iterates since they are $\mathcal{O}(1)$ in our example and that testing on the residue (i.e., $\|F(U^{n+1})\|_\infty$ small enough) would need an appropriate scaling of the equations. We have successfully employed the stopping criterion (4.4).

C. *fitting criterion:* a consequence of our stepsize strategy may be that the t-parameter assumes values very close to 1. Obviously, this is not efficient. To avoid this situation we require, in each step, that the remaining t-interval is an integer multiple of the current stepsize.

The stepsize strategy we have implemented can be formulated by the following rules:

(i) first, we set $\Delta t = 1$, i.e. we directly try to solve (4.2) for the endpoint $t = 1$.
(ii) after each iteration, we test on criterion A; if this criterion is not satisfied, we abandon the iteration process, halve the stepsize and restart.
(iii) if the iteration passes this test, we apply criterion B, i.e. we test on "convergence" (cf. (4.4).
(iv) if the process has "converged", we increase the stepsize by a factor γ (subject to criterion C) and try to solve the next (sub)problem.

In Appendix B we give a flowchart which may elucidate the above strategy.

4.1.3. Correction transformation

As already mentioned in Section 2.3, this numerical study is performed in terms of the variables $\{\psi,\phi^+(=\phi_p),\phi^-(=\phi_n)\}$. This choice is motivated by our wish to avoid complications due to the extremely wide value range of p and n. However, the price we have to pay is that the system to be solved is strongly nonlinear, whereas the system in terms of $\{\psi,p,n\}$ is mildly nonlinear.

As pointed out in [8], it is possible to take advantage of the benefits of both approaches. Here, we shortly reproduce this technique:

Let $U=(u_j)$ with $u_j=(\psi_j,\phi_j^+,\phi_j^-)^T$, where $j=0,1,...,N$, the number of grid points (cf. (3.1)); similarly, let $V=(v_j),v_j=(\psi_j,p_j,n_j)^T$, where the relation between ϕ_j^+,ϕ_j^- and p_j,n_j is defined in (2.6).

Let us write the transformation $V\mapsto U$ as $U=\Omega(V)=(\omega(v_0),\omega(v_1),...,\omega(v_N))^T$ with $\omega:\mathbb{R}^3\to\mathbb{R}^3$. Using (2.6) it is readily verified that

$$\omega'(v_j)=\begin{bmatrix} 1 & 0 & 0 \\ 1 & \dfrac{1}{\alpha p_j} & 0 \\ 1 & 0 & -\dfrac{1}{\alpha n_j} \end{bmatrix}. \tag{4.5}$$

Furthermore, let

$$G(V) := F(\Omega(V)) = 0 \tag{4.6}$$

denote the system for the original variables V, so that both systems are identical.

Now, the idea of the correction transformation is to employ the operator G, which is much less nonlinear than the operator F, but still performing the calculations in terms of U. Applying Newton's methods to (4.6) we obtain

$$V^{n+1} - V^n = -[G'(V^n)]^{-1}G(V^n). \tag{4.7}$$

Substituting

$$\begin{aligned} G'(V) &= F'(U).\Omega'(V) \\ &= F'(U).\ \mathrm{Diag}(\omega'(v_0),\omega'(v_1),...,\omega'(v_N)) \end{aligned}$$

into (4.7) yields

$$\Omega'(V^n)[V^{n+1}-V^n] = -[F'(U^n)]^{-1}F(U^n). \tag{4.8}$$

The final step is to express the left-hand side of (4.8) in terms of the corrections on U. Observe that this left-hand side can be partitioned into vectors r_j in $\mathbb{R}^3$, $r_j = (r_{1,j},r_{2,j},r_{3,j})^T = \omega'(v_j^n)[v_j^{n+1}-v_j^n]$, $j = 0,1,...,N$; thus, the three variables in V can be transformed for each grid point, separately.

Let

$$\Delta v_j = (v_j^{n+1}-v_j^n) = (\Delta\psi_j,\Delta p_j,\Delta n_j)^T.$$

We find

$$r_j = \omega'(v_j^n)\Delta v_j = \begin{pmatrix} 1 & 0 & 0 \\ 1 & \dfrac{1}{\alpha p_j^n} & 0 \\ 1 & 0 & \dfrac{-1}{\alpha n_j^n} \end{pmatrix} \begin{pmatrix} \Delta\psi_j \\ \Delta p_j \\ \Delta n_j \end{pmatrix} = \begin{pmatrix} \Delta\psi_j \\ \Delta\psi_j + \Delta p_j/(\alpha p_j^n) \\ \Delta\psi_j - \Delta n_j/(\alpha n_j^n) \end{pmatrix} = \begin{pmatrix} \Delta\psi_j \\ \Delta\psi_j + \dfrac{1}{\alpha}\exp[\alpha(\Delta\phi_j^+ - \Delta\psi_j)] - \dfrac{1}{\alpha} \\ \Delta\psi_j - \dfrac{1}{\alpha}\exp[\alpha(\Delta\psi_j - \Delta\phi_j^-)] + \dfrac{1}{\alpha} \end{pmatrix}.$$

Now, the corrections $\Delta u_j = u_j^{n+1} - u_j^n = (\Delta\psi_j,\Delta\phi_j^+,\Delta\phi_j^-)^T$ in terms of the U-variables are easily found to be

$$\Delta u_j = \theta(r_j) := \begin{pmatrix} r_{1,j} \\ r_{1,j} + \dfrac{1}{\alpha}\ln(1+\alpha(r_{2,j}-r_{1,j})) \\ r_{1,j} - \dfrac{1}{\alpha}\ln(1-\alpha(r_{3,j}-r_{1,j})) \end{pmatrix}. \tag{4.9}$$

Defining $\Theta(R) := (\theta(r_0),\theta(r_1),...,\theta(r_N))^T$ for $R=(r_j)$, each $r_j \in \mathbb{R}^3$, the resulting process in terms of the U-variables can thus be written as

$$U^{n+1} = U^n + \Theta(-[F'(U^n)]^{-1}F(U^n)). \tag{4.10}$$

We observe that, *mathematically,* this process is equivalent to (4.7) (with $U^n = \Omega(V^n)$ for all n). Or, in other words, the present method is just a reformulation of Newton's method (4.7) for the original variables ψ,p and n.

However, *in actual application,* there is an important difference between the processes (4.7) and (4.10): the Jacobian $F'(U^n)$ in (4.10) is easily invertible, whereas the $G'(V^n)$ in (4.7) turned out to be numerically singular in our experiments. This is due to the fact that $G'(V) = F'(U)\Omega'(V)$, where $\Omega'(V)$ is ill-conditioned since p and n assume very large values (cf. (4.5)).

To sum up, instead of working with (4.1), in each iteration step we first compute the values $(r_{1,j},r_{2,j},r_{3,j})$ by solving the linear systems in the right-hand side of (4.8). Then the corrections in terms of the U-variables are computed from (4.9).

REMARK 1. Clearly, if one of the logarithms in (4.9) has a nonpositive argument, the correction transformation in its present form is not feasible. If this situation is encountered, one of the following remedies can be applied:

R1: if one of the arguments is nonpositive, then the transformation is omitted for the corresponding grid point, i.e., $\Delta u_j=(r_{1,j},r_{2,j},r_{3,j})^T$.

R2: $\Delta\phi_j^+ = r_{1,j}+\frac{1}{\alpha}\ln[\max(1+\alpha(r_{2,j}-r_{1,j}),\epsilon)]$,
$\Delta\phi_j^- = r_{1,j}-\frac{1}{\alpha}\ln[\max(1-\alpha(r_{3,j}-r_{1,j}),\epsilon)]$,
where ϵ is a small positive number.

R3: if $1+\alpha(r_{2,j}-r_{1,j})\leqslant 0$ then $\Delta\phi_j^+ = r_{1,j}+\max(r_{2,j}-r_{1,j},-.026)$,
if $1-\alpha(r_{3,j}-r_{1,j})\leqslant 0$ then $\Delta\phi_j^- = r_{1,j}+\max(r_{3,j}-r_{1,j},+.026)$.

This last remedy is in use at Philips. The number .026 is very close to $1/\alpha$, hence R_3 is approximately equivalent with:

$$\text{if } 1+\alpha(r_{2,j}-r_{1,j})\leqslant 0 \text{ then } \Delta\phi_j^+ = r_{1,j}-1/\alpha$$
$$\text{and if } 1-\alpha(r_{3,j}-r_{1,j})\leqslant 0 \text{ then } \Delta\phi_j^- = r_{1,j}+1/\alpha.$$

We have implemented all remedies; their influence on the performance will be discussed in Section 5.1.

REMARK 2. We end up this section with a brief discussion on the above remedies. First of all, it is easy to show that one of the logarithms in (4.9) will have a nonpositive argument iff some p_j^{n+1} or n_j^{n+1} would come out nonpositive in (4.7). Here, we recall that p and n denote concentrations and are nonnegative by definition. Hence, a nonpositive argument in one of the logarithms corresponds to a nonphysical situation. Bearing this in mind, the remedies R_1 and R_2 have been constructed. They allow for the following interpretation: Remedy R_1 simply means: if one of the arguments in the logarithm, say in the j^{th} component, is nonpositive then the component u_j^{n+1} is computed from (4.10) by replacing $\theta(r_j)$ by the identity operator.

Process (4.10) in combination with remedy R_2 is equivalent with the Newton process (4.7) in which p_j^{n+1} and n_j^{n+1} are "cut off" if they become smaller than ϵp_j^n and ϵn_j^n, respectively. This can be shown by considering the following modification of (4.7)

$$V^{n+1} - V^n = -\Lambda^n[G'(V^n)]^{-1}G(V^n), \tag{4.11}$$

where $\Lambda^n = \text{Diag}(\Lambda_0^n,\Lambda_1^n,\ldots,\Lambda_N^n)$, with each Λ_j^n a 3×3 diagonal matrix, say diag $(\xi_j^n,\eta_j^n,\zeta_j^n)$. In the same way as before, this can be written as (cf. (4.8))

$$\Omega'(V^n)[\Lambda^n]^{-1}(V^{n+1}-V^n) = -[F'(U^n)]^{-1}F(U^n).$$

Setting $R=(r_j)=-[F'(U^n)]^{-1}F(U^n)$, $r_j=(r_{1,j},r_{2,j},r_{3,j})^T (j=0,1,...,N)$ we obtain, similar to (4.9),

$$\Delta u_j = \begin{pmatrix} \Delta\psi_j \\ \Delta\phi_j^+ \\ \Delta\phi_j^- \end{pmatrix} = \begin{pmatrix} \xi_j^n r_{1,j} \\ \xi_j^n r_{1,j} + \frac{1}{\alpha}\ln(1+\alpha\eta_j^n(r_{2,j}-r_{1,j})) \\ \xi_j^n r_{1,j} - \frac{1}{\alpha}\ln(1-\alpha\zeta_j^n(r_{3,j}-r_{1,j})) \end{pmatrix}.$$

By using $r_j = \omega'(v_j^n)[\Lambda_j^n]^{-1}(v_j^{n+1}-v_j^n)$ we obtain for the arguments in the logarithms the expressions

$$1 + \alpha\eta_j^n(r_{2,j}-r_{1,j}) = p_j^{n+1}/p_j^n,$$

$$1 - \alpha\zeta_j^n(r_{3,j}-r_{1,j}) = n_j^{n+1}/n_j^n.$$

It follows that remedy R_2 (replacing $\ln(x)$ by $\ln(\max(x,\epsilon))$) is equivalent with (4.11), where $\xi_j^n \equiv 1$, and $\eta_j^n, \zeta_j^n \in (0,1]$ are as large as possible such that $p_j^{n+1} \geqslant \epsilon p_j^n$ and $n_j^{n+1} \geqslant \epsilon n_j^n$. Finally, we note that remedy R_3 does not seem to allow for such an interpretation.

4.1.4. Additional information

We conclude this section with some remarks:

(i) a frequently used variant of Newton's method consists in "freezing" the Jacobian matrix during all or a number of iterations (modified- or quasi-Newton method). Anticipating the description of the numerical results, we remark that we have experimented with a lot of variants of this type. However, the overall efficiency was not increased, when compared with the classical Newton process. Therefore, in all experiments reported in Section 5, the Jacobian was updated in each iteration, like we assumed so far.

(ii) another commonly used technique is the so-called damped Newton process, that is, not the full Newton-correction but only a fraction is used in updating the last iterate (cf. (4.1)):

$$U^{n+1} = U^n - \lambda_n[F'(U^n)]^{-1}F(U^n),$$

for some suitably chosen $\lambda_n > 0$. The idea is to improve the global convergence behaviour. However, in order to retain the excellent local convergence of Newton's method, the procedure for choosing λ_n should allow $\lambda_n = 1$ near the solution. It is certainly not a trivial task to select an optimal λ-sequence; moreover, a lot of additional F-evaluations will be needed for properly choosing these damping factors.

Our numerical experiments have indicated that the global convergence behaviour of Newton's method is strongly improved by the correction transformation (see Section 4.1.3). Therefore, we think that this technique, in combination with a judicious choice of the continuation parameter t, makes damping redundant.

4.2. Multiple grid methods

Another class of methods which can successfully be applied in semiconductor device simulation consists of schemes which obtain information from more than one spatial grid (see also Bank et al. [1]).

A well-known example is the multigrid method. Here, the spatial domain is covered by a sequence of grids with increasing mesh sizes. Then, some iteration process is applied (e.g. a defect correction process) and the iteration errors are expanded in discrete Fourier series. In terms of these Fourier expansions we can roughly distinguish between smooth (low-frequency) and nonsmooth (high-frequency) error components. Usually, some relaxation technique is used to quickly damp the nonsmooth components on a particular grid. The low-frequency modes, which are responsible for the slow asymptotic convergence, are damped on a coarser grid, where these frequencies are high(er) frequencies, relative to that grid.

An obvious advantage of this method is that the major part of the work can be performed on the coarser grids. Moreover, in many applications, the total number of iterations does not increase as the

finest-grid discretization parameter h tends to zero. For a profound discussion on multigrid methods, we refer to [2,3].

To get some insight in its performance, we implemented a very simple version in which only two grids are involved, viz. the grid on which the solution is requested, called the fine grid, and one additional (coarse) grid by alternatingly deleting one grid point. For the coarse-grid operator we simply chose the fine-grid operator, restricted to the coarse grid. We remark that in a real multigrid approach a much more sophisticated coarse-grid operator is employed.

Thus we obtain two nonlinear systems (cf. (3.1))

$$F_h(U_h) = 0 \quad \text{and} \quad F_H(U_H) = 0$$

for the fine- and coarse-grid solutions U_h and U_H, respectively.

Now, again introducing a continuation parameter t and using the same strategy as described in Section 4.1.3, we "solved" the coarse-grid problem $F_H(U_H(t),t)=0$ if $t<1$. Only, if t has reached its final point (i.e. $t=1$), that is, the solution U_h on the original (finest) grid is desired, we changed to the finest-grid problem, taking the last obtained coarse-grid solution as an initial approximation. Here we need an operator which transforms U_H into U_h, usually called a prolongator. In the literature, various sophisticated prolongation techniques have been proposed; we implemented the simplest one, viz. linear interpolation. An alternative might be, for example, to exploit the exponential behaviour of the solution and to fit the coarse-grid solution by exponential functions from which a fine-grid approximation can be obtained. This may be the topic of future research.

In spite of all these simplifications, we obtained a significant speedup in the overall performance of the method (see Section 5.3).

We think that this type of methods is very promising and has a great future in the context of the present application.

Finally, we remark that the above multiple grid modification was obtained by minimal programming effort. As a matter of fact, most of the alterations were restricted to implementing in a number of DO-loops a variable stride which took the value 2 (for $t<1$) or 1 (for $t=1$). In doing so, the data structures could be maintained.

5. Numerical results

We will present the numerical results obtained by applying the various methods described in the preceding section. First, we shall specify some quantities, not yet defined and we shall introduce some nomenclature to facilitate the description of the experiments.

All numerical experiments were restricted to one fixed nonuniform spatial grid Δ_h with number of grid points $N=130$. The mesh sizes $h_j, j=1,...,N$ are given by (see also Section 3)

$$h_j := \begin{cases} .95_{10}{}^{-5}, & j=1,...,10 \\ .6991_{10}{}^{-5}, & j=11,...,20 \\ .37_{10}{}^{-7}, & j=21,...,30 \\ .129_{10}{}^{-6}, & j=31,...,40 \\ .2343_{10}{}^{-5}, & j=41,...,50 \\ .3427_{10}{}^{-5}, & j=51,...,60 \\ .536_{10}{}^{-5}, & j=61,...,70 \\ .2731_{10}{}^{-5}, & j=71,...,80 \\ .1_{10}{}^{-4}, & j=81,...,120 \\ .9485_{10}{}^{-5}, & j=121,...,130 \end{cases} \tag{5.1}$$

and the grid points are obtained from $x_j=x_{j-1}+h_j$, $j=1,...,130$, $x_0=0$. Hence, the basis which is

positioned at $x = 1.9_{10}{}^{-4}$ corresponds to grid point 50.

Next, we specify the set of applied voltages $\{V_E, V_C\}$ where a solution is required. In fact, each given set of these "output points" defines what we call "a Problem." We have tested four such Problems, defined by:

Problem I:

case nr.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
V_E	0	0	0	0	0	0	-.2	-.4	-.6	-.7	-.8	-.85	-.9	-.95	-1.0
V_C	0	.2	.4	.6	.8	1	1	1	1	1	1	1	1	1	1

Problem II:

case nr.	0	1	2
V_E	0	0	-1
V_C	0	1	1

Problem III:

case nr.	0	1	2
V_E	0	0	-1
V_C	0	.5	5

Problem IV:

case nr.	0	1	2
V_E	-1	0	0
V_C	1	1	0

The nature of our test model is such that increasing the collector voltage does not offer serious problems. However, decreasing the emitter voltage gives rise to substantial difficulties. Especially for V_E in the range [-.8,-1] the solution rapidly varies, if this boundary value is slightly changed. This motivates the clustering of the output points in this region, as specified in Problem I (in Appendix A, plots are given of the solution of this problem). However, if one is only interested in a solution for $\{V_E = -1, V_C = 1\}$, then the input as specified in Problem II is relevant. It should be observed that this problem is a severe test for the stepsize strategy, since now the algorithm has to pass the difficult range automatically. Problem III is very similar to the second problem but now the voltage jump at the collector equals 5. Problem IV has the same set of output points as Problem II but in reversed order. This means that the most difficult part of the solution process is encountered in the initial phase, which may have consequences in designing an optimal strategy.

To have some form of reference, we solved Problem I with the classical Newton method. That is, without strategy and without correction transformation. Moreover, we added (by trial and error) a number of output points if the solution of case i was outside the contraction region for case $i+1$. The total number of Newton iterations to reach the solution of case 14 ($V_E = -1, V_C = 1$) was 424. Obviously, using the classical Newton method is not an efficient way to tackle this particular problem, but is provides some insight in the merits of the correction transformation technique as well as in the strength of the strategy.

We mention that the linear systems arising in the Newton process possess a bandstructure with 5 lower and 3 upper codiagonals. To solve these systems we used the routine LEQT1B from the IMSL-

library [5].

In testing the problems I, II and IV, we used a bandwith (see criterion A, Section 4.1.2) specified by

$$\psi \in [-0.7, 2.0],$$
$$\phi^+ \in [-1.0, 1.5],$$
$$\phi^- \in [-1.5, 1.5].$$

For problem III these ranges are enlarged to

$$\psi \in [-0.8, 7.5],$$
$$\phi^+ \in [-1.0, 7.5],$$
$$\phi^- \in [-1.5, 7.5].$$

Although these numbers require some preknowledge on the solution, we remark that they can be chosen rather arbitrary. Provided that they safely enclose the solution, the performance of the algorithm is not significantly influenced by their values.

Finally, a few remarks on the solution corresponding to case 0. For the problems I, II and III, we have $\phi_j^+ \equiv 0$, $\phi_j^- \equiv 0$, $\forall j$. The potentials ψ_j are easily obtained by applying Newton's method to the equations (3.2) starting with the initial approximation $\psi_j = \frac{1}{\alpha} \operatorname{arc\,sinh} (\frac{1}{2} D(x_j)/n_i)$, i.e. by assuming space charge neutrality (see [8] and Section 2.2).

The final solution from Problem I was used as case 0-solution in Problem IV.

The effort involved in finding these case 0-solutions is not taken into account in the tables of result.

5.1. Effect of the correction transformation

First of all, we observe that the correction transformation has a positive effect on the robustness of the Newton method. By this we mean that the global convergence behaviour is strongly improved. For example, if we repeat the experiment with Newton's method including the correction transformation (but still without the continuation strategy), then the total number of iterations decreases from 424 to 163. This gain is mainly due to the fact that fewer additional points were needed to proceed from one case to the other.

Next, we implemented the strategy with respect to the choice of the continuation parameter t. For the time being, we fixed the strategy parameter γ to 1.25 and TOL is defined in (4.4b). On the basis of the Problems I and II, we have tested the various remedies R1, R2 and R3 as defined at the end of Section 4.1.3. The results of this test can be found in the Tables 1 and 2. These results give rise to the following conclusions and observations:

1) the second and third remedy are of comparable efficiency, whereas $R1$ falls far behind. We decided to select $R2$, since this has a simple interpretation (cf. Section 4.1.3, Remark 2); hence, in the sequel, all experiments will be performed using this remedy.
2) we separately tested the influence of ϵ on the second remedy. For ϵ within the range $[10^{-1}, 10^{-5}]$ we did not notice a difference in the number of iterations; therefore ϵ is set to 10^{-4}.
3) another observation is that the strategy is working well, especially for Problem II. To get some insight into its performance, especially in the way the stepsizes Δt have been chosen and how many rejections have been occurred (due to criterion A), we add Figure 2. This figure illustrates the behaviour of the strategy (with remedy $R2$). The accumulated number of iterations is plotted as a function of t if the algorithm is applied to Problem II and proceeds from case 1 to case 2, i.e., $V_C \equiv 1$ and V_E: $0 \rightarrow -1$ as t: $0 \rightarrow 1$.
   From this picture we may conclude that the algorithm is capable of detecting the difficult region and that only 9 iterations are spent to detect that the stepsize was too large, resulting in a rejection. We consider this an acceptable performance for an automatic mechanism.

4) Finally, we deduce from Table 2 that the initial phase of the problem (i.e. increasing the collector voltage) offers no problems at all: the solution of case 0 is inside the convergence region of case 1.

TABLE 1. Number of Newton iterations required to solve Problem I (in parentheses the accumulated numbers).

case nr.	$\{V_E, V_C\}$	$R1$	remedy $R2(\epsilon=10^{-4})$	$R3$
0	0,0	-	-	-
1	0,.2	5(5)	6(6)	6(6)
2	0,.4	5(10)	5(11)	5(11)
3	0,.6	5(15)	5(16)	5(16)
4	0,.8	5(20)	5(21)	5(21)
5	0,1	5(25)	5(26)	5(26)
6	-.2,1	5(30)	5(31)	5(31)
7	-.4,1	5(35)	5(36)	5(36)
8	-.6,1	16(51)	6(42)	6(42)
9	-.7,1	6(57)	6(48)	6(48)
10	-.8,1	7(64)	7(55)	7(55)
11	-.85,1	8(72)	8(63)	8(63)
12	-.9,1	20(92)	16(79)	16(79)
13	-.95,1	21(113)	16(95)	16(95)
14	-1,1	23(136)	18(113)	19(114)

TABLE 2. Number of Newton iterations required to solve Problem II (in parentheses the accumulated numbers)

case nr.	$\{V_E, V_C\}$	$R1$	remedy $R2(\epsilon=10^{-4})$	$R3$
0	0,0	-	-	-
1	0,1	18(18)	7(7)	7(7)
2	-1,1	89(107)	62(69)	63(70)

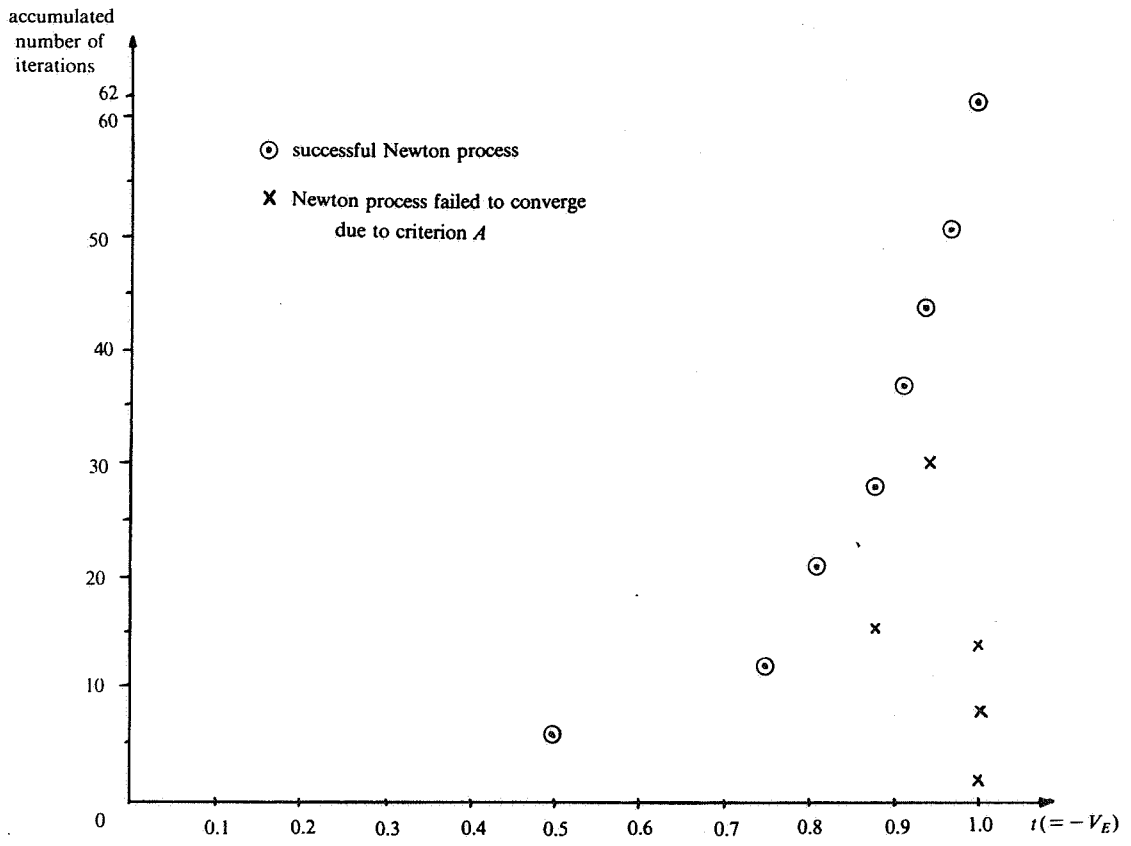


FIGURE 2. Performance of the strategy

5.2. The influence of γ *and TOL*

Next, we will test the influence of the strategy parameters γ (the growing-factor for the stepsize) and TOL (the stopping criterion in subproblems). For several values of these parameters, we solved the Problems I and II.

TABLE 3. Number of Newton iterations as a function of the strategy parameters γ and TOL

TOL	Problem I $\gamma=1$			Problem II $\gamma=1.25$		
3	101	101	101	*	*	*
1	101	101	101	*	*	*
3_{10}^{-1}	107	107	107	53	57	58
10^{-1}	111	111	111	61	61	67
3_{10}^{-2}	112	112	112	66	66	70
10^{-2}	113	113	113	69	69	72
3_{10}^{-3}	114	114	114	71	71	75
10^{-3}	115	115	115	71	71	78
10^{-4}	115	115	115	76	76	81
10^{-11}	120	120	120	86	86	94

Table 3 contains the results of this test from which we conclude:

1° the *growing-factor* γ is of no influence in problems with a dense set of output points which, a priori, is adapted to the difficult regions in the process (Problem I); or, in other words, before the factor γ can manifest itself, a next output point has been reached. We recall that, additionally, criterion C is imposed which prevents the strategy to exhibit optimal performance for this problem.

However, in the second problem the situation is quite different. Especially if we realize that the step from case 0 to case 1 (i.e. the jump in the collector voltage) is performed without subdivision, i.e. independent of γ. Hence, the strategy is only active when the process proceeds from case 1 to case 2. The results indicate that a growing-factor $\gamma = 1.5$ is too optimistic in this case, since the most-difficult part of the problem is obtained for t close to 1.

TABLE 4. Number of Newton iterations as a function of γ and TOL for problem IV

TOL	$\gamma=1$	$\gamma=1.25$	$\gamma=1.5$	$\gamma=2.0$
1	*	*	*	*
3_{10}^{-1}	77	67	57	61
10^{-1}	139	85	75	81
3_{10}^{-2}	140	86	75	82
10^{-2}	141	87	76	85
10^{-4}	159	96	82	88
10^{-11}	192	110	92	99

Therefore, we also tested Problem IV, in which the output points are specified in reversed order, thus transforming the difficult part to the initial phase of the process. Therefore, once this difficult phase has been traversed, the algorithm should be able to increase its stepsizes. Consequently, a relatively large growing-factor is expected to result in a superior behaviour. The results, collected in Table 4, confirm these expectations (note especially the results for $\gamma=1$), and indicate that $\gamma=1.5$ is optimal for this problem.

In conclusion, we think that $\gamma \in (1.25, 1.5)$ is a good choice.

2° Next, we consider the influence of the *parameter TOL*. We observe that the results show the

expected behaviour: the more stringent the stopping criterion, the more expensive the process. For very large values of TOL, however, the algorithm failed to reach the endpoint $t=1$; such a failure is indicated by an * in the tables. It turned out that, due to such a crude tolerance, the accepted "solution" in an intermediate point is too far from the true solution. Then, proceeding with this solution, the way back to the solution curve cannot be found anymore, and the strategy has completely lost the control. Hence, for the sake of robustness, one should be a bit conservative in choosing TOL.

Another observation is that, in situations where the strategy is hardly active (e.g. in Problem I) the effect of TOL is not very pronounced. For the Problems II and IV, however, there is a substantial difference between the extreme values TOL = 0.3 and TOL = 10^{-11}.

Taking into account that the solution is $\mathcal{O}(1)$, a value from the range $(10^{-2}, 10^{-1})$ seems to be a reasonable choice for the parameter TOL.

Finally, we applied the algorithm to Problem III. The difference with Problem II merely consists in an increased jump of the collector voltage. Again, the solution for case 1 could be obtained from the case 0-solution without activating the strategy ($\Delta t=1$ was successful). For this part, 9 iterations are required, whereas Problem II needs 7 iterations to drop the Newton corrections below 10^{-11}. The conclusions with respect to the parameters γ and TOL are similar to those obtained in the previous tests. Table 5 contains the complete results.

TABLE 5. Number of Newton iterations as a function of γ and TOL for problem III

TOL	$\gamma=1$	$\gamma=1.25$	$\gamma=1.5$	$\gamma=2$
1	*	*	*	*
3_{10}^{-1}	66	63	71	73
10^{-1}	77	72	76	76
3_{10}^{-2}	88	76	81	81
10^{-2}	85	79	84	84
10^{-4}	94	86	91	91
10^{-11}	112	108	116	116

5.3. Effect of using a coarse grid

To conclude this numerical section, we will shortly discuss the modification suggested in Section 4.2. This comprises the use of information obtained on coarser grids.

Here we will give the result of a preliminary test achieved by the modification as outlined in Section 4.2. We solved the second part of Problem II (i.e. from case 1 to case 2), since the first part does not need a strategy. As a reference, we consider the single-grid approach, which required 62 iterations (cf. Table 2, $R2(\epsilon=10^{-4})$).

The two-grid variant needed 53 coarse-grid iterations and 16 fine-grid iterations; hence, the total amount of work is approximately equivalent with $53/2+16=43$ fine-grid iterations.

Thus, a substantial reduction of computer time is obtained by this extremely simple modification.

6. CONCLUSIONS AND RECOMMENDATIONS

In this report we have described our experiences in computing stationary solutions of a 1D model problem for semiconductor devices. In solving the nonlinear system, emphasis was placed upon Newton's method, which was extended with a technique called "correction transformation" (cf. [8] and Section 4.1.3). The resulting scheme was studied in the context of continuation type methods and imbedded in such an environment, where the continuation parameter was associated with the boundary values.

A strategy was designed for properly choosing the steplength of the continuation parameter. Several

tests are described which served to tune the strategy parameters. We believe that, for this choice of the continuation parameter, the resulting algorithm is close to the optimum, and no much gain can be achieved by improving upon this strategy. However, we have not considered to employ other problem parameters for the continuation process (see also the remark in Section 4.1). It may turn out that there exists a better choice for this parameter which, in combination with an adequate strategy for its steplength, results in an improved overall performance. This possibility may be the subject of future research.

Another conclusion is that the correction transformation technique in combination with a reliable strategy, strongly improves the global convergence behaviour of Newton's method. This approach is, in our opinion, a more elegant (and less expensive) way in obtaining a robust algorithm than the frequently used damped form of Newton's method (see Section 4.1.4., (ii)).

In higher-dimensional, real-life problems, for example in designing VLSI structures, the calculation of the static behaviour of a single device should be standard. For example, it should be possible, preferably interactively, to repeat such calculations many times. Depending on the computer facilities, of course, it may turn out that the above class of methods still behaves unsatisfactorily to meet this requirement.

In order to achieve a real break-through, for that purpose, multigrid methods are serious candidates. In this study we had a glimpse on a first attempt in that direction. Of course, much more numerical evidence is required to judge the merits of this type of methods in the present application. However, due to the very positive results obtained in other fields, we think that these methods are very promising and worth to be subject of extensive further research.

Another important aspect to mention is that the present problems possess a moving front in their solutions. This behaviour strongly indicates that the use of a so-called "adaptive-grid technique" will be beneficial. The grid that we used in this study has some refinement near $1.6_{10}{}^{-4}$, but this region did not always coincide with the region where the solution exhibits its largest gradients. The development of adaptive-grid software has not yet reached a high level of sophistication (especially for higher-dimensional problems) and it certainly will require a lot of research to reach an acceptable level. Nonetheless, for a substantial increase in the efficiency of numerical simulation of semiconductor devices, these techniques should be examined as well, since they offer an attractive alternative to fixed-grid methods.

References

1 R.E. Bank, D.J. Rose and W. Fichtner, *Numerical methods for semiconductor device simulation,* SIAM J. Sci. Stat. Comput. 4, 1983, pp. 416-435.

2 A. Brandt, *Multi-level adaptive solutions to boundary value problems,* Math. Comp. 31, 1977, pp. 333-390.

3 P.W. Hemker, *Introduction to multigrid methods,* Nieuw Archief voor Wiskunde (3) XXIX, 1981, pp. 71-101.

4 C. den Heijer and W.C. Rheinboldt, *On steplength algorithms for a class of continuation methods,* SIAM J. Numer. Anal. 18, 1981, pp. 925-948.

5 IMSL library, Revised Edition 9.2, 1984, Houston, Texas.

6 P.A. Markowich, *The Stationary Semiconductor Device Equations,* Springer-Verlag, 1986.

7 J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables,* New York, Academic Press, 1970.

8 S.J. Polak, C. den Heijer, W.H.A. Schilders and P.A. Markowich, *Semiconductor device modelling from the numerical point of view,* Int. J. Numer. Meth. Engng 24, 1987, pp. 763-838.

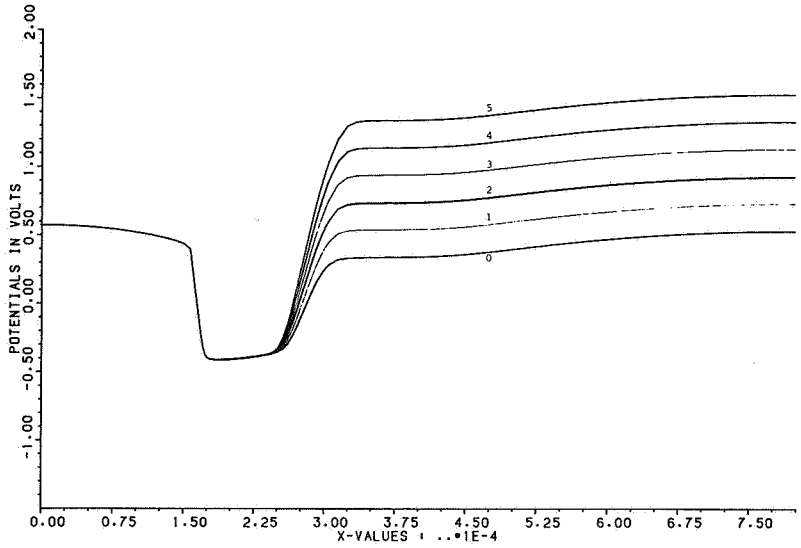
9 S. Selberherr, *Analysis and Simulation of Semiconductor Devices,* Springer-Verlag, 1984.

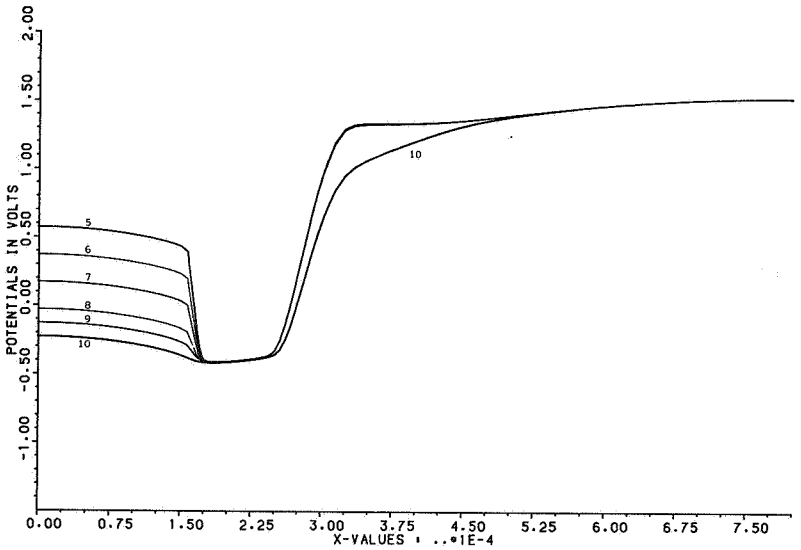
APPENDIX A

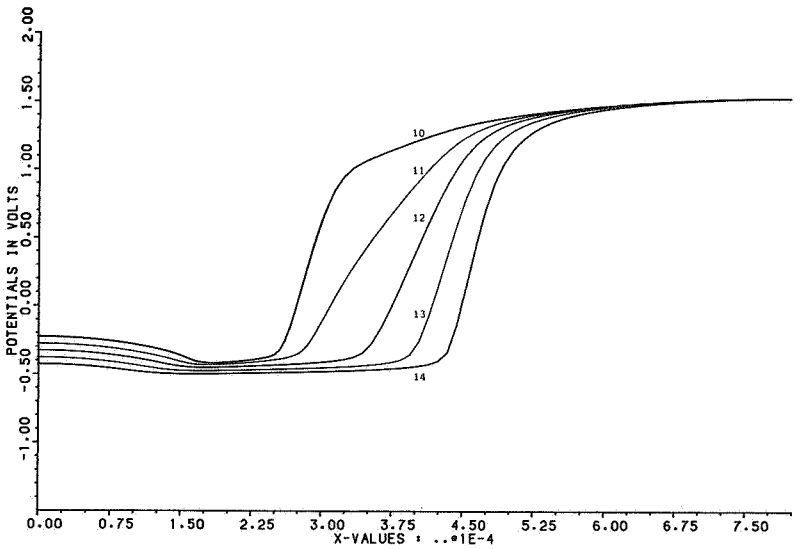
To provide some insight into the behaviour of the solution as a function of the boundary values, we add the following plots. They show the solution of Problem I; the numbers along the lines refer to the various cases, i.e. 0,1,...,14.

In order to illustrate the tremendous range of values assumed by the variables p and n, we include plots for these variables. The plots were supplied by Dr. W.H.A. Schilders, Philips.

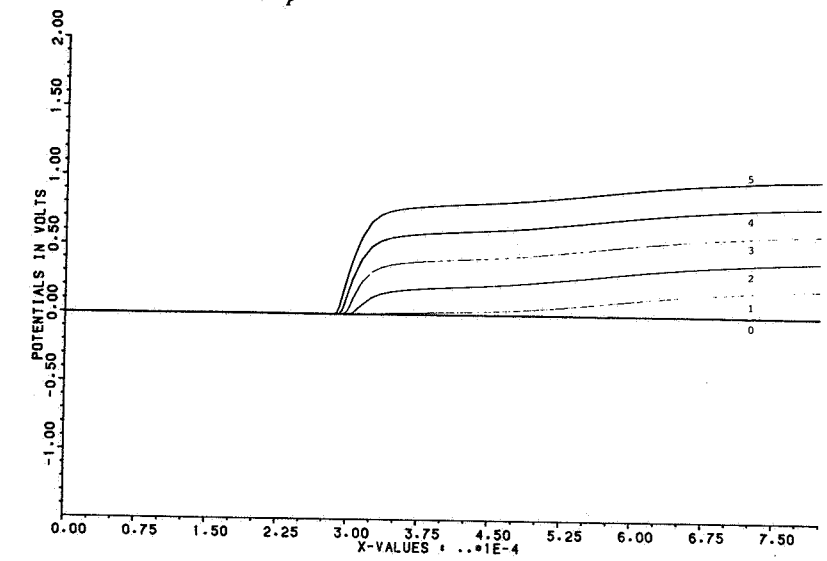
THE ELECTROSTATIC POTENTIAL ψ:

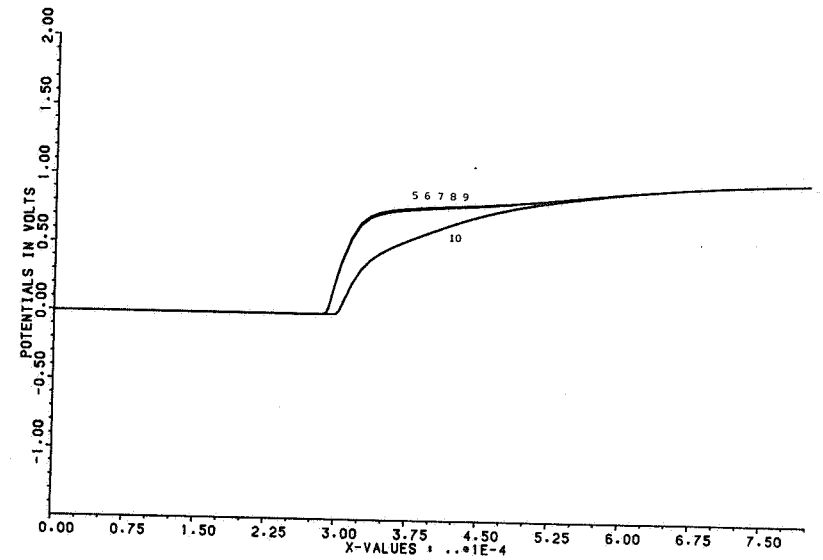


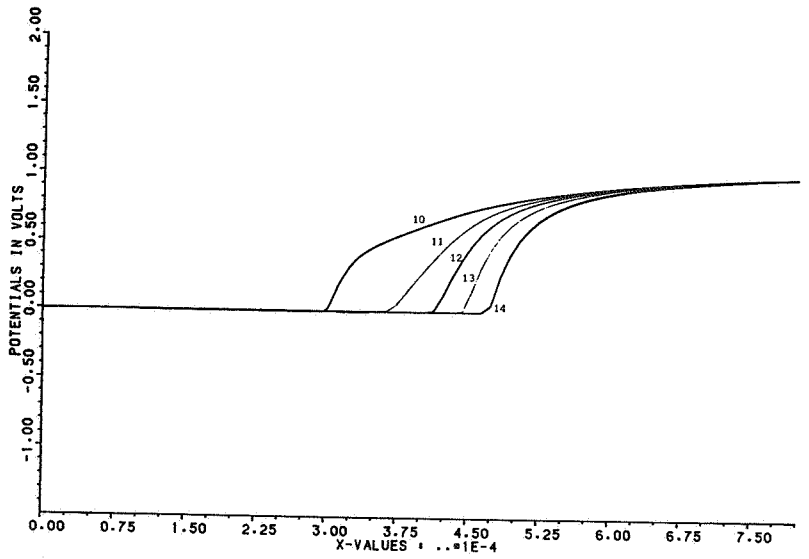




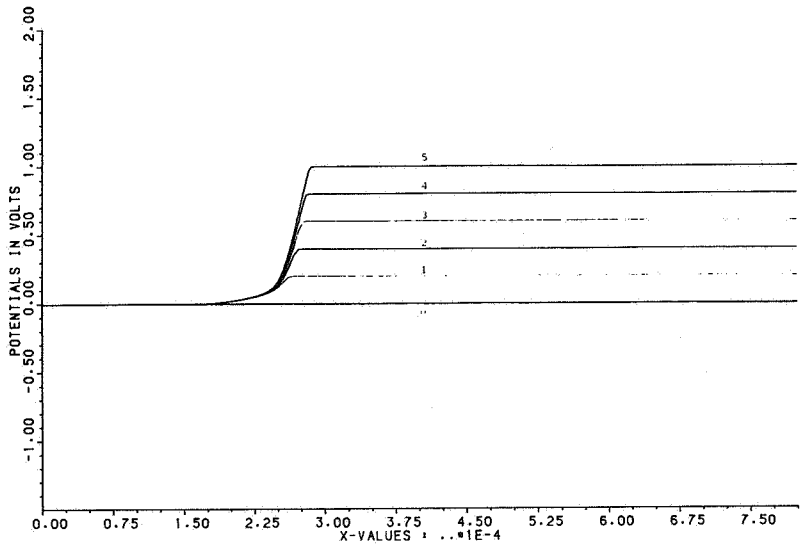
THE HOLE QUASI-FERMI POTENTIAL ϕ_p:

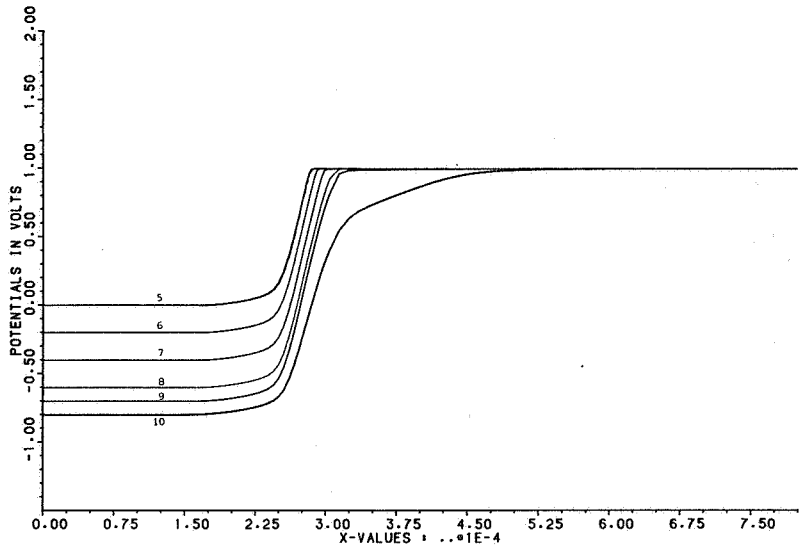


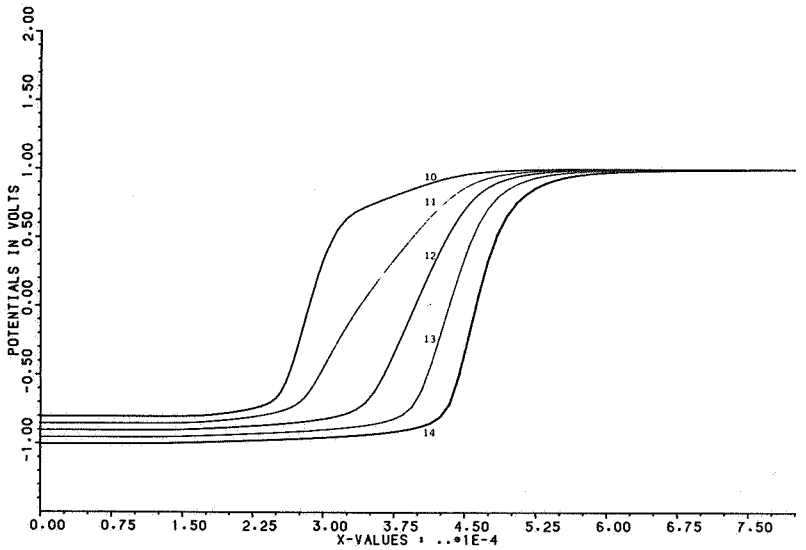




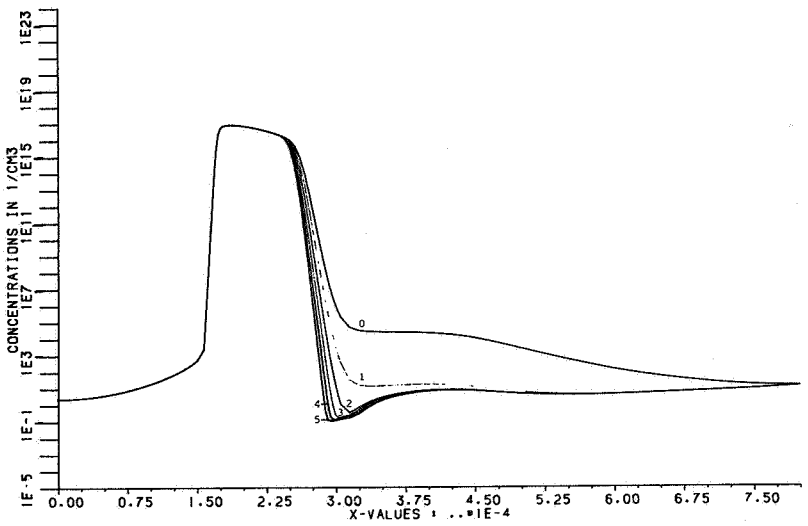
THE ELECTRON QUASI-FERMI POTENTIAL ϕ_n:

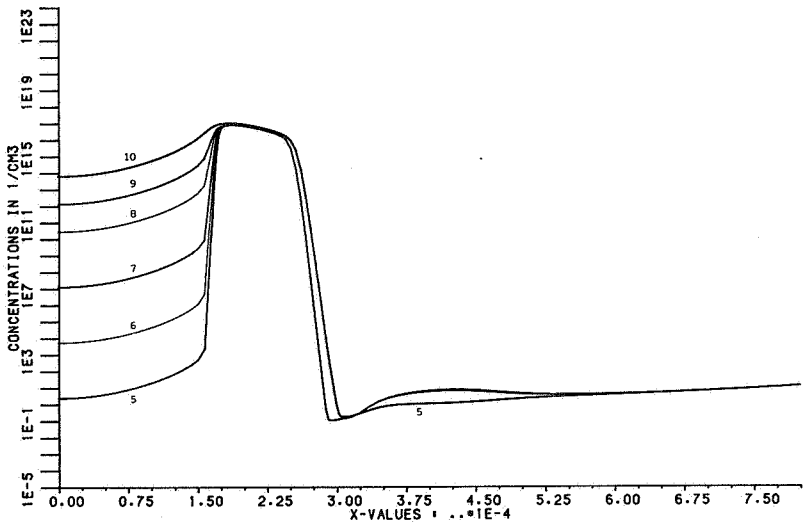


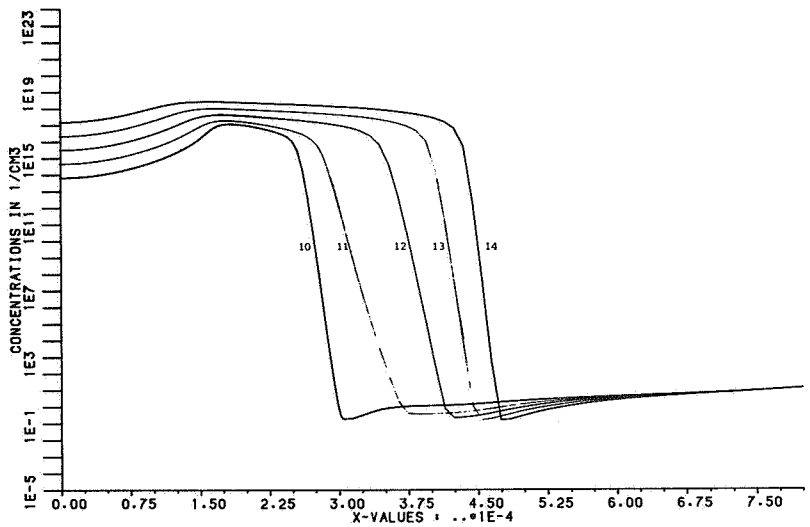




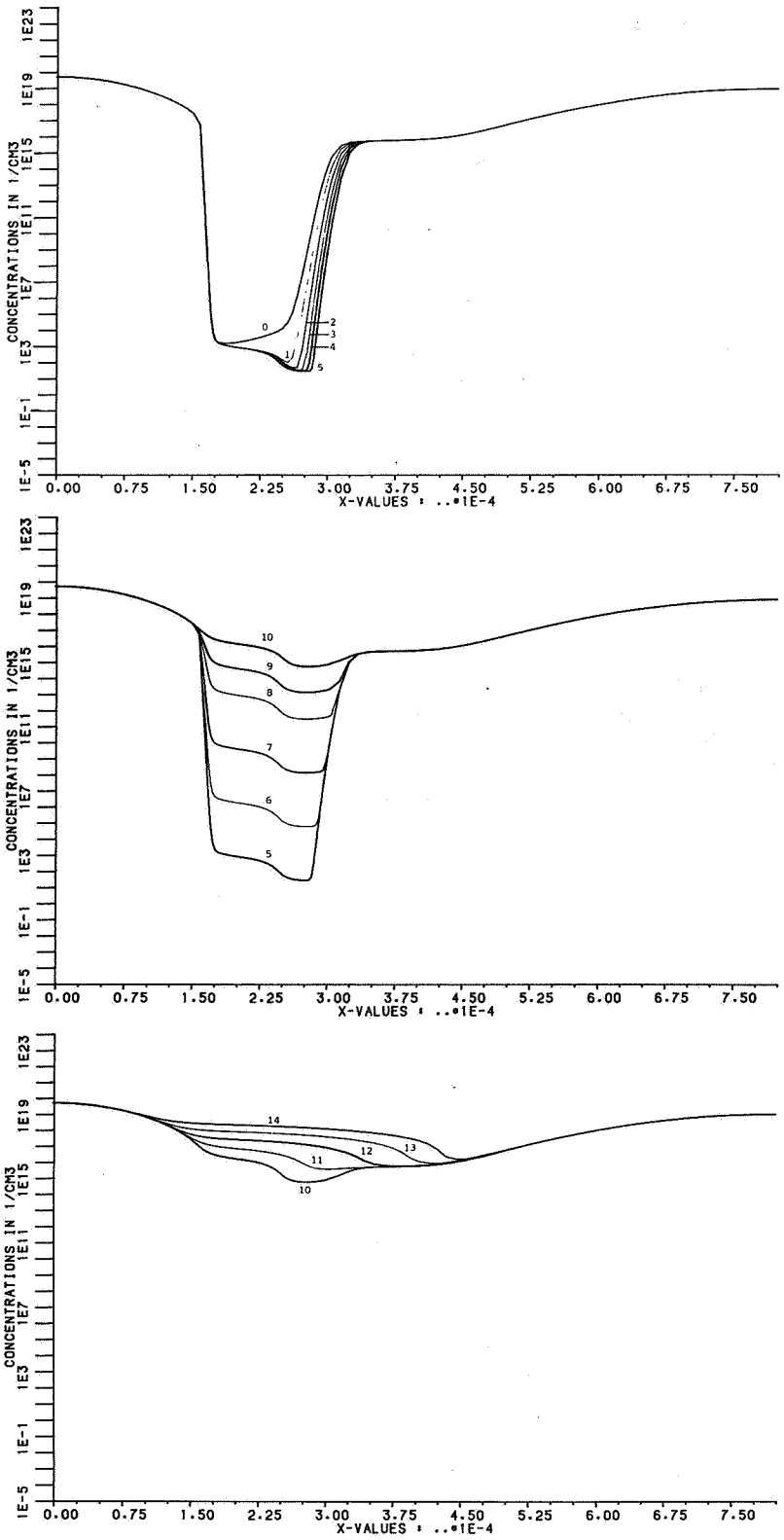
THE HOLE CONCENTRATION p:







THE ELECTRON CONCENTRATION n:



APPENDIX B

Here we present a flowchart showing the strategy as defined in Section 4.1.2: