

**Using Multigrid for Semiconductor Device
Simulation in 1-D**

by
Stephen E. Adams and Uri M. Ascher

Technical Report 88-13

September 1988

Using Multigrid for Semiconductor Device Simulation in 1-D

by

Stephen E. Adams and Uri M. Ascher

Abstract

This paper examines the application of the multigrid method to the steady state semiconductor equations in one dimension. A number of attempts reported in the literature have yielded only limited success in applying multigrid algorithms to this sensitive problem, suggesting that a more careful look in relatively simple circumstances is worthwhile.

Several modifications to the basic multigrid algorithm are evaluated based on their performance for a one-dimensional model problem. It was found that use of a symmetric Gauss-Seidel relaxation scheme, a special prolongation based on using the difference operator, and local relaxation sweeps near junctions, produced a robust and efficient code. This modified algorithm is also successful for a wide variety of cases, and its performance compares favourably with other multigrid algorithms that have been applied to the semiconductor equations.

§1. Introduction

The construction of semiconductor devices is a complex process. While the mass production of semiconductors is inexpensive, the creation of a few prototypes can be expensive and time consuming. The use of numerical simulations reduces the number of prototypes constructed, with a corresponding reduction of costs and time commitments. Finding techniques for solving the corresponding mathematical models reliably and efficiently has therefore emerged as an important task. This paper examines the application of the multigrid method to the numerical simulation of semiconductor devices.

The equations which govern the behaviour of semiconductors were first developed by van Roosbroeck in the early 1950's [40]. They form a system of nonlinear partial differential equations which describe the electrostatic potential, current flow, and carrier concentrations. Once scaled, these equations can be viewed as singularly perturbed. They are subject to both Dirichlet and Neumann boundary conditions. Sharp layers, where the solution varies rapidly, arise at junctions in the interior of the domain.

The numerical solution of the semiconductor equations has proved to be a challenging problem. Several surveys on this topic have been published, including [32], [31] and [7]. A nonstandard discretization of the equations must be used, and the equations are poorly scaled. Transformations between certain naturally arising sets of dependent variables can become extremely ill-conditioned [4]. The sharp internal layers require that a highly nonuniform mesh be used to provide a uniformly accurate resolution of the solution efficiently. The iterative procedure which is needed to deal with the nonlinearities must be both robust and efficient, since simulations are usually performed for a wide range of operating conditions.

The multigrid method [9], [10], [11], [19] is an efficient numerical algorithm for solving the algebraic equations which arise from the discretization of boundary value differential problems. Relaxation sweeps are combined with coarse grid correction steps to produce a scheme which is, theoretically, optimally efficient. The method has been successfully used for many problems, but it is not always obvious how it should be implemented to attain the maximum efficiency for a given application. Previous attempts to apply the multigrid method to the semiconductor equations (e.g. [8], [16] and [20]) have met with limited success, and do not always appear to be as efficient as desirable. In particular, most efforts that we are aware of (except [20]) must use a rather fine coarsest grid, limiting the utility of the multigrid iteration.

This paper evaluates several modifications to the basic multigrid algorithm, and demonstrates that the multigrid method can be successfully used for semiconductor device modelling in one space dimension. With the most successful variants derived, an agreeably small number of multigrid iterations is needed for a variety of tests tried, with the coarsest grid being as coarse as possible. While the successful solution of the one-dimensional problem does not guarantee an immediate similar success for problems in more than one space variable, it is clear that what does not work in one dimension will not work in the multi-dimensional case. Our study offers both promise and some concrete directions for developing efficient multigrid solvers of more general instances

of the semiconductor device simulation problem.

In §2 we recall the semiconductor device equations and their stable discretization. Some scaling, transformations and a discussion on anticipated difficulties are briefly given as well.

We have divided the design of a successful algorithm, somewhat arbitrarily perhaps, into two stages. The first stage is described in §3, where a reasonable basic multigrid algorithm is derived. It is based on the FAS algorithm of Brandt (see, e.g. [10]), and includes some additional treatment at junctions. The behaviour of this still straightforward implementation is described.

Improvements to the basic algorithm are described in §4. Symmetric relaxation iterations, prolongations based on the difference operator, and local relaxation sweeps all significantly improve the convergence of the method. These modifications, and their effects, are described first. The most successful variants are collected into one program, SC-1, which is described at the beginning of §4.2. We then demonstrate that SC-1 is successful for a wide variety of instances of the problem, including the modelling of a thyristor. Further, we compare SC-1 to an adaptation of a multigrid implementation described by Hemker [20]. For the problems considered, SC-1 is the more efficient and more robust program.

§2. The problem and its discretization

A complete discussion of the physics of semiconductor devices can be found in [32], [39] and [35]. Under appropriate physical conditions the semiconductor device equations can be written [40] as

$$\epsilon \Delta \psi = q(n - p - C) \quad (1)$$

$$\operatorname{div} J_n - q \frac{\partial n}{\partial t} = qR \quad (2)$$

$$\operatorname{div} J_p + q \frac{\partial p}{\partial t} = -qR \quad (3)$$

$$J_n = qD_n \nabla n - q\mu_n n \nabla \psi \quad (4)$$

$$J_p = -qD_p \nabla p - q\mu_p p \nabla \psi. \quad (5)$$

Substitution of (4), (5) into (2), (3) results in a system of three nonlinear partial differential equations for ψ, n and p . These equations are subject to mixed Dirichlet and Neumann boundary conditions. The meaning of the various variables is as follows:

ψ : electrostatic potential

$n > 0$: electron concentration

$p > 0$: hole concentration

J_n : electron current density

J_p : hole current density.

The quantities q , ϵ , D_n and D_p are assumed to be constants, representing the elementary charge, the permittivity, and the electron and hole diffusion coefficients, respectively. The other functions appearing above are R : the recombination-generation rate; C : the doping profile; and μ_n (μ_p) the electron (hole) mobility rate.

We consider here the steady state problem in one space variable x . Moreover, we follow Markowich, Ringhofer and others (see [25] and references therein) in considering the problem in singular perturbation scaling. With this scaling, all lengths are scaled by a characteristic device length l , and the potentials are scaled by the thermal voltage U_T . The doping profile $C(x)$, and the carrier concentrations, are scaled by C_m , the maximum magnitude of $C(x)$. Finally the carrier mobilities μ_n and μ_p are scaled to have order of magnitude 1. See [25] for typical scaling values.

The resulting scaled equations, with the scaled quantities denoted by the same symbols as the unscaled ones, are:

$$\lambda^2 \psi'' = (n - p - C) \quad (6)$$

$$J_p' = -R \quad (7)$$

$$J_n' = R \quad (8)$$

$$J_n = \mu_n (n' - n\psi') \quad (9)$$

$$J_p = -\mu_p (p' + p\psi') \quad (10)$$

where

$$\lambda = \sqrt{\frac{\epsilon U_T}{q C_m l^2}}$$

is a very small value, typically of the order of magnitude 10^{-3} to 10^{-5} . Dirichlet conditions, corresponding to Ohmic contacts, are prescribed on the boundary ∂D . In scaled form, these boundary conditions are:

$$\begin{aligned}\psi|_{\partial D} &= \ln\left(\frac{C + \sqrt{C^2 + 4\delta^4}}{2\delta^2}\right)|_{\partial D} + \begin{cases} 0 & \text{at left interval end} \\ V_0 & \text{at right interval end} \end{cases} \\ n|_{\partial D} &= 1/2(C + \sqrt{C^2 + 4\delta^4})|_{\partial D} \\ p|_{\partial D} &= 1/2(-C + \sqrt{C^2 + 4\delta^4})|_{\partial D},\end{aligned}$$

where $\delta^2 = n_i/C_m$ is another small parameter (n_i is the intrinsic carrier concentration) and V_0 is the applied bias.

A complete discussion of the parameter models is contained in [32]. This part of the modelling process is far from trivial, but for the purposes of our study it suffices to take $\mu_n \equiv \mu_p \equiv 1$ and $R \equiv 0$. The doping profile $C(x)$ is assumed to be piecewise constant, with jump discontinuities at junctions. While not physical, the resulting models still retain all the major features which potentially cause numerical difficulties.

The equations are singularly perturbed since a small parameter λ^2 multiplies the highest derivative in Poisson's equation. The solution components are expected to have sharp transition layers across junctions, since where ψ is slowly varying $n - p \sim C(x)$ (as $\lambda \rightarrow 0$) and the doping profile $C(x)$ has jump discontinuities at the junctions (see Figures (1 - 4) in §3.1). A numerical solution of the equations should account for this behaviour: a fine discretization will be needed near the layers if the solution is to be accurately approximated there, but a coarse grid will suffice elsewhere. Still, the discretization scheme has to be stable, especially in the multigrid context, even when the grid is coarse at junctions.

For both analytic and computational work there are possible advantages to using sets of dependent variables other than ψ , n and p [31], [8]. One transformation in common use replaces the scaled carrier concentrations n and p by the quasi-Fermi levels ϕ_n and ϕ_p as follows:

$$n = \delta^2 e^{\psi - \phi_n} \quad (11)$$

$$p = \delta^2 e^{\phi_p - \psi}. \quad (12)$$

The basic equations (6) - (10) are transformed into :

$$\lambda^2 \psi'' - (\delta^2 e^{\psi - \phi_n} - \delta^2 e^{\phi_p - \psi} - C(x)) = 0 \quad (13)$$

$$(\delta^2 \mu_n e^{\psi - \phi_n} \phi_n')' + R = 0 \quad (14)$$

$$(\delta^2 \mu_p e^{\phi_p - \psi} \phi_p')' - R = 0, \quad (15)$$

so the continuity equations are now in divergence form.

The second transformation to the Slotboom variables u and v reads:

$$n = \delta^2 e^{\psi} u \quad (16)$$

$$p = \delta^2 e^{-\psi} v, \quad (17)$$

or, equivalently:

$$u = e^{-\phi_n} \quad (18)$$

$$v = e^{\phi_p}. \quad (19)$$

Both transformations “regularize” the problem, in that ϕ_n , ϕ_p , u and v are all slow variables (they vary slowly through the junctions, as do the fluxes J_n and J_p , but unlike ψ , n and p), and the equations are no longer singular, singularly perturbed (cf. [4]). The Slotboom variables are attractive analytically because the continuity equations are linear in u and v , while (14) and (15) are quasilinear in ϕ_n and ϕ_p ; further the continuity equations are self adjoint for given values of ψ and R . The dynamic ranges of (ϕ_n, ϕ_p) and (n, p) are moderate, and successful numerical computations have been performed using both sets of variables [8], [31]. The Slotboom variables are poorly scaled; this formulation is generally only useful for analytic investigations.

Some feeling for anticipated difficulties with this problem is given by the observation that the problem in terms of n and p is stable under certain restrictions [4], but the transformation (11), (12) is not necessarily well-conditioned. It has been observed in practice [31], [20], that the linearization in terms of the concentration variables appears to be better suited for numerical purposes than in the other variables. Thus it appears that no one set of dependent variables satisfies all desires, and this is a crucial reason for the sensitivity of this problem to multigrid treatment.

There are several classical methods which can be used to discretize the semiconductor equations [7]. Finite elements, finite differences and the finite box method (a generalization of the finite difference method) have all been successfully used to generate discretizations. The most commonly used approach is a specialized finite difference scheme known as the Scharfetter-Gummel discretization [33]. This approach will now be recalled.

We consider a set of grid points $\{x_i\}_{i=0,1..N}$ with $x_0 = 0$ and $x_N = 1$ the boundary points. For simplicity, assume that the grid is uniform, denoting $h = x_{i+1} - x_i$. The discretization of Poisson’s equation with a small parameter is a standard problem which is treated in many publications (see for instance [30]). We use the standard three point scheme:

$$\frac{\lambda^2}{h^2}(\psi_{i+1} - \psi_i) - \frac{\lambda^2}{h^2}(\psi_i - \psi_{i-1}) = n_i - p_i - C(x_i). \quad (20)$$

Note that as $\lambda \rightarrow 0$ the reduced solution away from the junctions is reproduced.

The treatment of the continuity equations is less standard. Although the small parameter λ^2 appears explicitly only in Poisson’s equation, the entire system is singularly perturbed (see, for example [4]). The standard symmetric differences do not damp local errors generated at junctions, so are not sufficiently stable. The transformation to the Slotboom variables (u, v) is regularizing but numerically unusable. However, it is useful for generating a discretization. Writing

$$J_n = \delta^2 \mu_n e^\psi \frac{du}{dx} \quad (21)$$

and discretizing the current conservation relation using a symmetric scheme between $x_i - \frac{h}{2}$ and $x_i + \frac{h}{2}$ we have

$$\frac{J_{n,i+1/2} - J_{n,i-1/2}}{h} = R(x_i). \quad (22)$$

Assuming that μ_n , J_n and $\frac{du}{dx}$ are constant on the interval $[x_i, x_{i+1}]$, (21) gives

$$\frac{du}{dx} = \frac{J_n e^{-\psi}}{\delta^2 \mu_n}, \quad (23)$$

which can be integrated exactly between x_i and x_{i+1} . Substituting this into (22) we have:

$$\begin{aligned} & \frac{\mu_{n,i+1/2}}{h^2} (B(\psi_{i+1} - \psi_i) n_{i+1} - B(\psi_i - \psi_{i+1}) n_i) - \\ & \frac{\mu_{n,i-1/2}}{h^2} (B(\psi_i - \psi_{i-1}) n_i - B(\psi_{i-1} - \psi_i) n_{i-1}) = R(n_i, p_i), \end{aligned} \quad (24)$$

where $B(x) = \frac{x}{e^x - 1}$.

A similar treatment of the continuity equation for holes gives

$$\begin{aligned} & \frac{\mu_{p,i+1/2}}{h^2} (B(\psi_i - \psi_{i+1}) p_{i+1} - B(\psi_{i+1} - \psi_i) p_i) - \\ & \frac{\mu_{p,i-1/2}}{h^2} (B(\psi_{i-1} - \psi_i) p_i - B(\psi_i - \psi_{i-1}) p_{i-1}) = R(n_i, p_i). \end{aligned} \quad (25)$$

The local truncation error for the discretized equations is proportional to the grid spacing. The Scharfetter-Gummel scheme is an instance of exponential fitting (see e.g. [5] and references therein).

§3. The Basic Multigrid Algorithm

Let us briefly recall the multigrid idea. Extensive discussions about the multigrid method and its applications can be found in [19], [11], [10] and [37]. First, consider a problem $Lu = f$ on some domain D , where L is an elliptic, scalar, linear differential operator which incorporates the boundary conditions. The discretization on a uniform grid with spacing h produces a system of linear algebraic equations

$$L_h u_h = f_h, \tag{26}$$

where L_h is a large, sparse matrix with additional properties inherited from the ellipticity of L . The multigrid method solves these equations iteratively, by mixing relaxation sweeps and coarse grid corrections.

Let U_h be the current approximation to u_h . A multigrid algorithm with a fixed strategy defined by three integers ν_1, ν_2 and γ , proceeds as follows: First, ν_1 relaxation sweeps are performed, aimed at smoothing the residual error. Then the defect is calculated and restricted to a coarser grid. The equations for a correction on the coarse grid are approximately solved using γ applications of the multigrid algorithm (they are solved exactly on the coarsest grid), and this solution is then prolonged to the fine h -grid to form the correction V_h which approximates $v_h = u_h - U_h$. A new approximation of the solution can then be found as $U_h := U_h + V_h$. Finally, ν_2 additional error smoothing relaxation sweeps are performed.

While rigorous convergence theory is available only for very simple model problems (certainly not for problems like the one considered here), a useful basic observation is that the multigrid method uses relaxations to eliminate the high frequency components of the error (high frequency, that is, relative to the grid spacing on which they are performed) and a coarse grid correction (CGC) to deal with the low frequency error components. Still, the effectiveness of the multigrid algorithm is determined by the details of the implementation, and the choice of its components — the relaxation operator, the restriction and prolongation, and the coarse grid operator — is often problem dependent. We will describe our choices for the basic algorithm below.

When the differential problem

$$N_h u_h = 0 \tag{27}$$

is nonlinear and includes several differential equations for several unknowns, as is the case for the semiconductor equations, a number of additional factors enter in the design of a successful solver. Recall that to handle nonlinear problems there are two approaches which use the multigrid method. One approach applies the linear multigrid algorithm to a linearization of the problem, obtained using Newton's method, or quasi-linearization. This approach has the advantage of modularity, since the new, nonlinear problem is reduced to a simpler, linear one. Also, the theory for this approach is better understood. On the other hand, it can be cumbersome and expensive in terms of storage. Moreover, in the case of the semiconductor equations we have nonlinear, ill-conditioned pointwise decoupling transformations which are used to derive stable

discretizations, so an early linearization seems ill-advised. We have therefore chosen to use a nonlinear multigrid algorithm. In particular, the FAS algorithm introduced by Brandt [10] is employed. Here the correction $v_h = u_h - U_h$ will satisfy the residual equation

$$\tilde{N}_h v_h \equiv N_h(U_h + v_h) - N_h U_h = r_h \quad (28)$$

where in general $\tilde{N}_h \neq N_h$. A new iterate is obtained as $U_h := U_h + V_h$ where V_h approximately solves $\tilde{N}_h v_h = r_h$.

For the relaxation operator in the basic code, we have used the Gauss-Seidel iteration with red-black ordering (see, e.g. [19], [10]). This is one of the most common, popular and simple relaxation schemes. The resulting systems of three nonlinear equations for three unknowns at each grid point are approximately solved simultaneously for the three unknowns using one Newton iteration. Near junctions, where the solution changes rapidly, we have found it beneficial to apply two steps of Newton's method to obtain a more accurate solution of the relaxation equations. So this is a modified collective Gauss-Seidel-Newton relaxation scheme.

The restriction operator I_k^{k-1} transfers the residuals from a finer grid G^k to a coarser grid G^{k-1} by using some form of averaging. We have used throughout this paper the full weighting

$$(I_k^{k-1} U^k)_j = 1/4(U_{2j-1}^k + 2U_{2j}^k + U_{2j+1}^k) \quad (29)$$

where U_j^k denotes solution values on grid G^k . The simpler injection scheme, which directly transfers information between the grids at common points, is well-known to be unsafe theoretically as well as practically for sensitive problems. And indeed, it did not work in our experiments either.

For the coarse grid operator we have used throughout the study the same difference operator as used on the fine grid, described in §2. This again is a usual and reasonable choice in practice.

The prolongation I_{k-1}^k transfers the correction (coarse grid solution) from a coarser grid G^{k-1} to a finer grid G^k . The most common, and simplest, prolongations are piecewise linear interpolations. These assume that the coarse grid solution is linear between grid points, so a simple averaging is used to find the interpolated values at the fine grid points. For a one-dimensional problem, using linear interpolation with a standard coarsening,

$$\begin{aligned} (I_{k-1}^k U^{k-1})_{2j} &= U_j^{k-1} \\ (I_{k-1}^k U^{k-1})_{2j+1} &= 1/2(U_j^{k-1} + U_{j+1}^{k-1}). \end{aligned} \quad (30)$$

This is the operator used in our basic algorithm. Note that, while it is not necessarily ideal for our problem (improvements will be discussed in the next section), it is the adjoint of our restriction operator, i.e. $I_k^{k-1} = (I_{k-1}^k)^*$. This is sometimes advantageous for theoretical purposes [26], [19].

Finally, in addition to the standard FAS algorithm, we have also implemented a nested iteration, or FMG version (see [10], [19]). Linear interpolation is used to

transfer the solution from a coarser grid to form an initial iterate on a finer grid in the FMG code. To accurately resolve the interior layer near a junction this scheme avoids interpolation across the junction, i.e. discontinuities in the doping profile are treated here as interval ends.

Both the FAS and FMG versions of the code were applied to the test problem described in §3.1 below. The programs use a series of uniform grids to discretize the domain, with $N_f = 64$ interior points used on the finest level and $N_c = 2$ points on the coarsest level. The grids are nested one within the next, with the grid size h_m doubled at each successive grid, so $h_{m-1} = 2h_m$ for $m = 2, 3, \dots, 6$. Generally, the finest grid is not sufficient to accurately resolve the junction layers.

§3.1 Description of a Model Problem — Test 1

To evaluate various modifications of the multigrid method a simple one dimensional model problem has been used. The two endpoints are assumed to be Ohmic contacts, so that only Dirichlet boundary conditions arise. Further, only uniform grids are used in the solution process.

The test problem used in this section (Test 1) consists of an np -diode on the interval $[-1, 1]$. The (scaled) doping profile has an np -junction at $x = 0$:

$$C(x) = \begin{cases} 1 & -1 \leq x \leq 0 \\ -10^{-6} & 1 \geq x > 0 \end{cases}$$

with $\lambda^2 = 0.4 \times 10^{-6}$ and $\delta^2 = 0.1 \times 10^{-6}$. Recombination and generation are assumed to be negligible, so $R \equiv 0$. Computations were performed for the thermal equilibrium case (i.e., $V_0 = 0$) and for $V_0 = 30$ (forward bias). (Note that in terms of the scaled variables, 1 Volt corresponds to $V_0 \approx 40$.)

Dirichlet conditions hold at the boundaries $x = -1$ and $x = 1$, as described in §2. The initial estimate used by the multigrid procedure was

$$u(x) = \begin{cases} u(-1) & x \leq 0 \\ u(1) & x > 0 \end{cases}$$

for $u = (\psi, n, p)$. This initial estimate satisfies the boundary conditions.

The various modifications were evaluated based on the number of iterations required to reach convergence. An iteration was terminated when two successive iterates u^{k+1} and u^k satisfied

$$|u_j^{k+1} - u_j^k| < 10^{-3} |u_j^{k+1}| \quad (31)$$

for each component u_j^{k+1} .

The solution components for both the equilibrium and forward biased problems are shown in Figures 1 - 4.

The application of our basic programs to this test problem has met with some success. Solutions were obtained even though the coarsest grid is very coarse and no continuation or damping of the nonlinear iteration were used. The number of multigrid iterations obtained for various ν_1, ν_2 and γ values is recorded in the Tables in §4.1. It

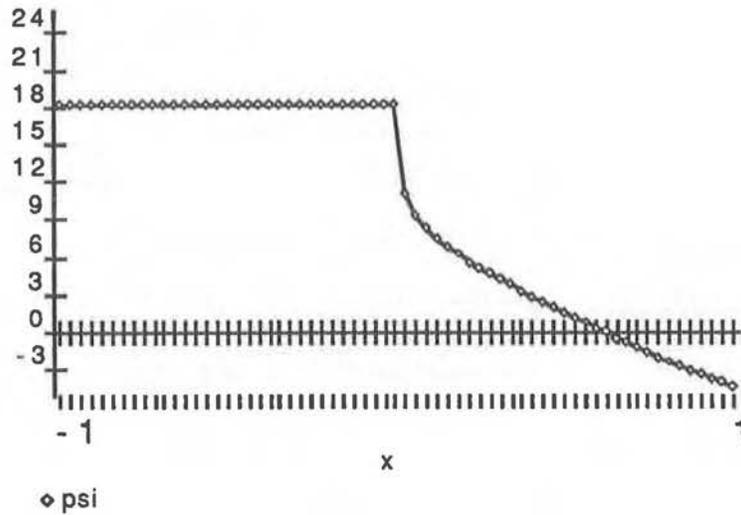


Figure 1: Potential ψ with $V_0 = 0$ (Test 1)

was felt, however, that greater efficiency can be achieved. The convergence rates were not impressive, and typically slowed down as the iteration proceeded. In addition to trying to improve the choice of relaxation, to which multigrid performance is often sensitive, our choice of prolongation operator raises questions here as well, because it does not prolongate the smoothest quantities across junctions, and it does not relate well to the principles used to derive the difference scheme. The following section further discusses the causes of the poor performance of the algorithm and presents modifications which greatly improve the convergence behaviour.

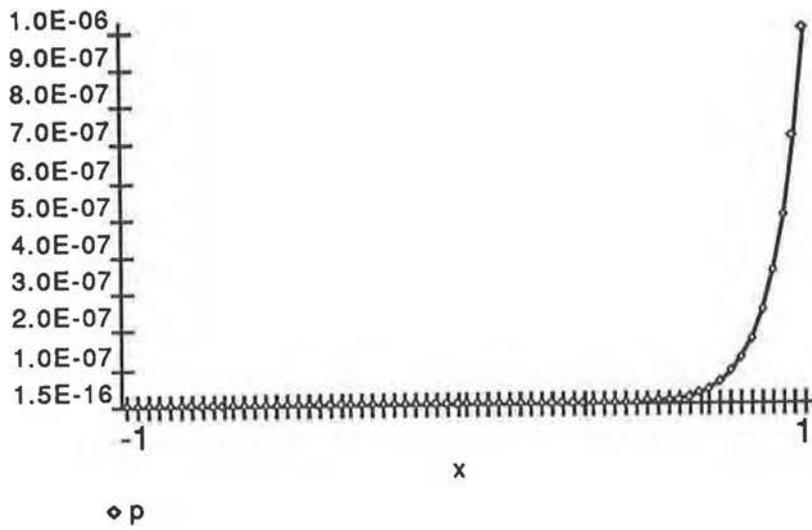
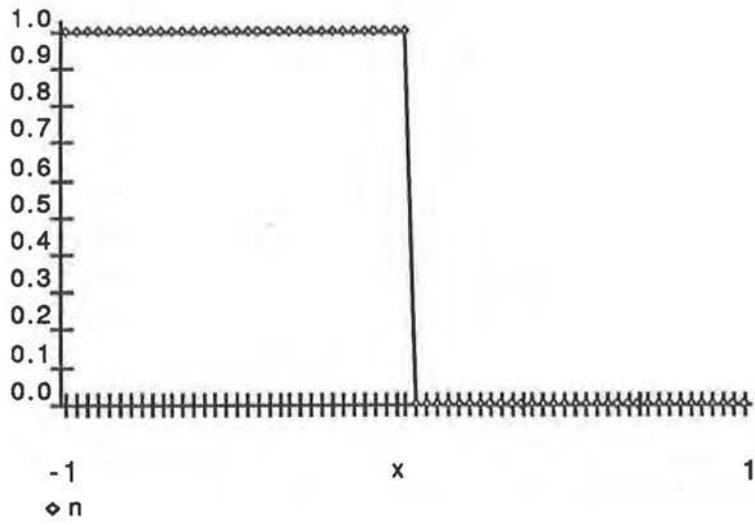


Figure 2: Carrier concentrations with $V_0 = 0$ (Test 1)

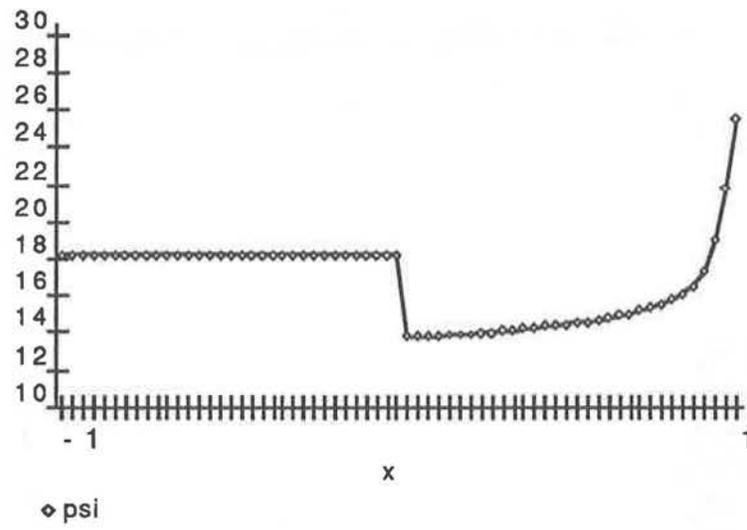


Figure 3: Potential ψ with $V_0 = 30$ (Test 1)

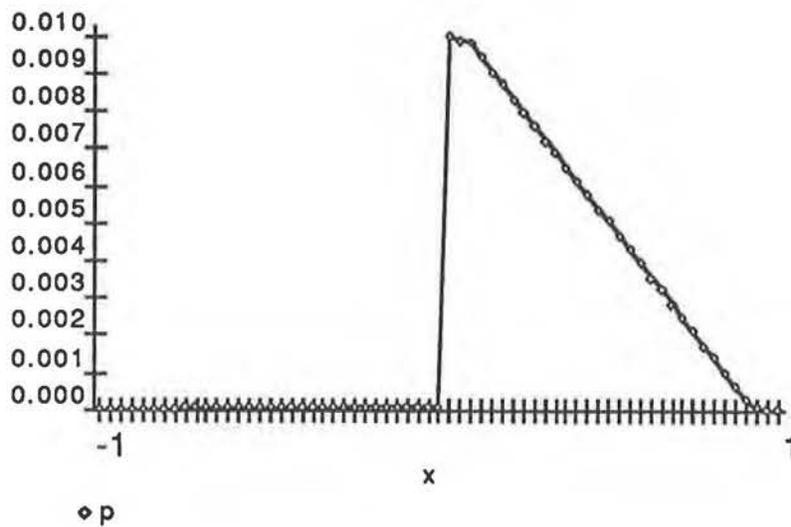
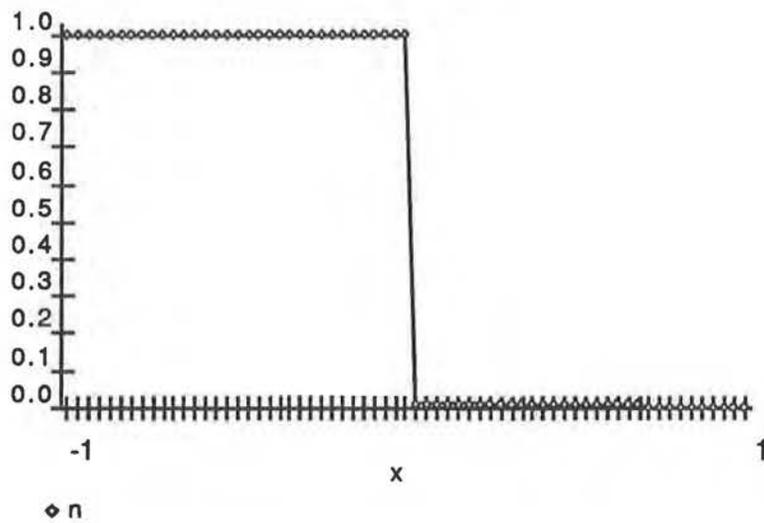


Figure 4: Carrier concentrations with $V_0 = 30$ (Test 1)

§ 4. An Improved Multigrid Implementation

§4.1 Modifications to the Basic Code

This section describes modifications to the basic codes which improve the convergence behaviour of the algorithm. The changes were evaluated based on their performance on the model problem described in §3.1. All modifications were made independently of one another, with the remainder of the code unchanged; the basic codes described in §3 were used as a framework for these modifications. Additional tests and comparisons, omitted here for brevity, are described in [1].

§4.1.1 Improvements to the Relaxation Scheme

Model problem analysis [37] shows that the relaxation procedure must balance two competing goals: it should reduce the high frequency errors but also must avoid exciting any smooth, low frequency components which cannot be quickly eliminated by relaxations. Similarly the coarse grid correction step should eliminate low frequency errors without creating any nonsmooth errors.

The rapid changes of the solution components near the np -junction introduce large errors in the interior layer. In particular, high frequency errors, which cannot be resolved on coarser grids and so must be eliminated on the finer grids, are generated. These errors can be damped by relaxations, but since the solution is only slowly varying throughout the rest of the domain additional relaxation sweeps are needed only near the junction.

Local relaxation sweeps were introduced which performed the usual red-black Gauss-Seidel relaxation on the three grid points neighbouring the junction, with the solution treated as fixed at all other points. The local sweeps were performed on all grids which contained enough points.

The effect of the local relaxation sweeps for the FMG implementation was remarkable, particularly for the forward bias case (Table 1). Although the added expense was minimal since the extra work was done on just a few grid points, the number of iterations required was reduced by a factor of one-half for some cases.

(γ, ν_1, ν_2)	Number of Iterations	
	$V_0 = 0$	$V_0 = 30$
(1, 1, 1)	6(7)	8(18)
(1, 2, 2)	5(5)	17(18)
(2, 1, 1)	3(6)	3(6)
(2, 2, 2)	4(4)	3(5)

Table 1: Number of iterations required for Test 1, using the standard FMG implementation, with local relaxations performed near the np -junction.

(In this and the following tables which list the number of iterations required for convergence, the bracketed numbers give the number of iterations taken by the basic

code; (–) means that the iteration failed to converge.)

The local relaxations made the FAS code more robust, but did not improve the speed of convergence in most cases.

As expected the local relaxation sweeps were most beneficial when just one full relaxation step was performed at each step ($\nu_1 = \nu_2 = 1$). With $\nu_1 = \nu_2 = 2$ the errors are already sufficiently smoothed so the extra relaxations are not that useful. Further, the local relaxations do not avoid the problem of exciting low frequency errors, and can cause the performance to degenerate in some instances.

The order in which the grid points are visited is important to the performance of the algorithm. It is particularly important for singularly perturbed problems that the relaxation sweep reflect the characteristic directions defined by the problem [10]. The carrier continuity equations depend on both $+\nabla\psi$ and $-\nabla\psi$ so information is being transmitted in both directions, and a symmetric relaxation scheme is appropriate [20].

The symmetric Gauss-Seidel relaxation (SYMGS) visits the grid points first in lexicographic order, and then in the reverse order. SYMGS reduced the number of iterations required for the test problems (Tables 2 and 3) for both the FAS and FMG implementations. The symmetric relaxation involves twice as much work as a red-black scheme since each point is visited twice during a sweep, but the performance improvement is not due only to this extra work: FMG using SYMGS requires 12 iterations with $\nu_1 = \nu_2 = 1$, while using the red-black scheme with $\nu_1 = \nu_2 = 2$ (which performs the same amount of work) takes 18 iterations. The relaxation ordering improved the convergence of the FMG implementation more than the FAS version, suggesting that the FAS error is dominated by low frequency errors which are significantly reduced by the FMG procedure on the coarser grids before the finest level is reached.

(γ, ν_1, ν_2)	Number of Iterations	
	$V_0 = 0$	$V_0 = 30$
(1, 1, 1)	5(11)	23(34)
(1, 2, 2)	4(6)	9(21)
(2, 1, 1)	5(7)	7(–)
(2, 2, 2)	4(6)	6(7)

Table 2: Number of iterations required for Test 1, using the standard FAS implementation with a symmetric Gauss-Seidel relaxation.

Local relaxation sweeps near the np -junction further reduced the number of iterations needed to reach convergence using SYMGS (Table 4). The effect for $(\gamma, \nu_1, \nu_2) = (1, 1, 1)$ is most noticeable — the local relaxations have much the same effect as the full relaxation sweeps $\nu_1 = \nu_2 = 2$. Also, the number of iterations needed for the FAS code using the symmetric relaxation with local sweeps is comparable to the number required by the FMG code.

(γ, ν_1, ν_2)	Number of Iterations	
	$V_0 = 0$	$V_0 = 30$
(1, 1, 1)	5(7)	12(18)
(1, 2, 2)	4(5)	10(18)
(2, 1, 1)	4(6)	5(6)
(2, 2, 2)	4(4)	4(5)

Table 3: Number of iterations required for Test 1, using the standard FMG implementation with a symmetric Gauss-Seidel relaxation.

(γ, ν_1, ν_2)	Number of Iterations	
	$V_0 = 0$	$V_0 = 30$
(1, 1, 1)	5(11)	13(34)
(1, 2, 2)	4(6)	12(21)
(2, 1, 1)	4(7)	6(-)
(2, 2, 2)	4(6)	5(7)

Table 4: Number of iterations required for Test 1, using the standard FAS implementation with a symmetric Gauss-Seidel relaxation and local relaxations performed near the junction.

(γ, ν_1, ν_2)	Number of Iterations	
	$V_0 = 0$	$V_0 = 30$
(1, 1, 1)	4(7)	10(18)
(1, 2, 2)	4(6)	10(18)
(2, 1, 1)	4(6)	4(6)
(2, 2, 2)	4(4)	4(5)

Table 5: Number of iterations required for Test 1, using the standard FMG implementation with a symmetric Gauss-Seidel relaxation, and local relaxation sweeps performed near the junction.

The collective Gauss-Seidel-Newton relaxation is very effective, but it involves solving a (3×3) system of linear equations at each grid point and for each iteration. We have therefore considered using the nonlinear Gauss-Seidel iteration known as Gummel's method (described in [32], for example) to approximately solve the nonlinear equations at each point. Four iterations of Gummel's method were used, with the points visited in a symmetric sweep. The number of multigrid iterations required is similar to the symmetric (collective) Gauss-Seidel relaxation for all cases. However, fewer iterations of Gummel's method at each point were not sufficient.

§4.1.2 Modified Prolongations

The sharp interior layer which forms near the np -junction creates difficulties for the prolongation operator. The errors are expected to vary rapidly near the junction, and the simple linear interpolation scheme used by the basic codes is not able to resolve these changes. This section describes three more effective prolongation operators. The first prolongation, PROL-1, is based on using the difference operator, while a second version, PROL-2, is a less expensive approximation of PROL-1. Both PROL-1 and PROL-2 significantly speed up the convergence of the code. PROL-3 handles the junction layer by implicitly prolongating the corrections in terms of the well-behaved quasi-Fermi variables. PROL-3 is of special interest because it can be more readily extended to multidimensional problems.

The idea of prolongating by first transferring values at grid points common to the coarse and fine grids and then satisfying the difference equations on the grid points which are on the fine grid but not on the coarse one has been frequently raised (see, e.g., §3.4.4 and §10.3 of [19]). The Algebraic Multigrid ideas in [26], [38] and the analyses in [22], [23] also lead to similar suggestions. These are based on an analysis of the range and nullspace of the multigrid operators for a linear problem which suggests that, after smoothing, the error should ideally lie in the range of the prolongation, so $v_h = I_H^h w_H$ for some w_H . For a red-black Gauss-Seidel relaxation scheme applied to the linear problem $L_h u_h = f_h$ this leads to the requirement that the prolonged correction $I_H^h w_H$ should satisfy the homogeneous residual equation at all black (odd) points, so

$$(L_h I_H^h w_H)_i = 0, \quad i \text{ odd.}$$

(Recall that, using a standard coarsening, the black (odd) points are those points which appear on the fine grid but not on the coarser grid.) This can be achieved by directly transferring the correction between grids at red points (which are common to both grids), and following this by a Gauss-Seidel sweep of the black points only.

The algebraic analysis requires not only a special prolongation operator but also the Galerkin form for the coarse grid operator and a special restriction operator as well. However, we found, as did others, that the choice of the prolongation is the crucial element in practice. This analysis is not directly applicable to the nonlinear problem $N_h u_h = 0$ anyway. Still, it is reasonable to again require that the prolonged

correction should satisfy the homogeneous residual equation at all odd points, so

$$(\tilde{N}_h I_H^h w_H)_i = 0, \quad i \text{ odd}$$

(see (27), (28)). This can be achieved in the same manner as for the linear problem, by performing a relaxation sweep over the black points. The prolongation PROL-1 is an implementation of these ideas. At each black point i , PROL-1 solves the following equations for the new values ψ_i^n , n_i^n and p_i^n :

$$h^{-2}(\psi_{i+1} - 2\psi_i^n + \psi_{i-1}) + n_i^n - p_i^n - C(x_i) = 0, \quad (32)$$

$$\begin{aligned} B(\psi_{i+1} - \psi_i^n)n_{i+1} - (B(\psi_i^n - \psi_{i+1}) + B(\psi_i^n - \psi_{i-1}))n_i^n + \\ B(\psi_{i-1} - \psi_i^n)n_{i-1} = 0, \end{aligned} \quad (33)$$

and

$$\begin{aligned} B(\psi_i^n - \psi_{i+1})p_{i+1} - (B(\psi_{i+1} - \psi_i^n) + B(\psi_{i-1} - \psi_i^n))p_i^n + \\ B(\psi_i^n - \psi_{i-1})p_{i-1} = 0, \end{aligned} \quad (34)$$

with updated values used at the neighbouring points $i \pm 1$. Newton's method was used to solve these equations. No damping was needed for any of the cases tried.

The prolongation PROL-1 greatly reduced the number of iterations required to reach convergence, compared to the basic code (Tables 6 and 7). With $(\gamma, \nu_1, \nu_2) = (1, 1, 1)$, PROL-1 reduces the number of iterations required by a factor of three. This prolongation does not seem to excite low frequency errors, so the results for $\gamma = 1$ and $\gamma = 2$ are much the same. Further, the FMG implementation provides no significant improvement over the FAS version, but FAS does not require the extra work to generate an initial guess on the finest level. (However, for more difficult problems the FMG version might be expected to be more robust since it produces a better starting point on the finest grid unless a continuation method is used.)

(γ, ν_1, ν_2)	Number of Iterations	
	$V_0 = 0$	$V_0 = 30$
(1, 1, 1)	7(11)	11(34)
(1, 2, 2)	4(6)	7(21)
(2, 1, 1)	5(7)	7(-)
(2, 2, 2)	4(6)	6(7)

Table 6: Number of iterations required for Test 1 using the standard FAS implementation, and the prolongation PROL-1.

While PROL-1 is effective, it is also expensive. At each odd point encountered by the prolongation a nonlinear system of equations needs to be solved. A simple

(γ, ν_1, ν_2)	Number of Iterations	
	$V_0 = 0$	$V_0 = 30$
(1, 1, 1)	6(7)	12(18)
(1, 2, 2)	4(5)	7(18)
(2, 1, 1)	5(6)	5(6)
(2, 2, 2)	4(4)	4(5)

Table 7: Number of iterations required for Test 1 using the standard FMG implementation, and the prolongation PROL-1.

approximation is to apply just one Newton iteration to (32) – (34) instead of solving this system very accurately at each odd point. This yields good results for FMG (see [1]) but poorer results for FAS, suggesting that the nonlinear equations do have to be solved accurately, but that one Newton iteration may often suffice for this purpose when starting from the good initial iterates provided by the FMG procedure.

We now derive PROL-2, another cheaper approximation of PROL-1. PROL-2 uncouples equations (32) – (34), which are solved by PROL-1, by approximating ψ values first. Analogously to the derivation of the Scharfetter-Gummel discretization for the continuity equations, the value ψ_i^n is found by using a linear interpolation of the correction at neighbouring points. Once ψ_i^n is known (33) and (34) determine n_i^n and p_i^n . The corrections dn_i and dp_i satisfy:

$$B(\psi_{i+1}^n - \psi_i^n)dn_{i+1} - (B(\psi_i^n - \psi_{i+1}^n) + B(\psi_i^n - \psi_{i-1}^n))dn_i + B(\psi_{i-1}^n - \psi_i^n)dn_{i-1} = 0 \quad (35)$$

and

$$B(\psi_i^n - \psi_{i+1}^n)dp_{i+1} - (B(\psi_{i+1}^n - \psi_i^n) + B(\psi_{i-1}^n - \psi_i^n))dp_i + B(\psi_i^n - \psi_{i-1}^n)dp_{i-1} = 0. \quad (36)$$

The superscript n emphasizes that the new (corrected) solution values are used at the neighbouring points.

The prolongation PROL-2 can also be viewed as an interpolation of the corrections to the current densities J_n and J_p . For $R \equiv 0$ J_n and J_p are constant. Moving from a coarse grid to a finer grid divides each interval into two new intervals. To enforce the condition that J_n be equal on these two intervals, at an odd point i , n_i^n , the new value of n_i , must satisfy

$$B(\psi_{i+1} - \psi_i)n_{i+1} - B(\psi_i - \psi_{i+1})n_i^n = B(\psi_i - \psi_{i-1})n_i^n - B(\psi_{i-1} - \psi_i)n_{i-1}, \quad (37)$$

Updated values of ψ , obtained by using a linear interpolation, are used in (37). Interpolating the correction dn_i (not the solution!) leads to (35). A similar treatment of J_p produces (36) for interpolating corrections to p .

Without using local relaxations PROL-2 is not effective. However, with additional local relaxation sweeps performed only in the neighbourhood of the np -junction the implementation is much more robust. The increased rate of convergence is particularly noticeable for the forward biased problem. With FAS and $(\gamma, \nu_1, \nu_2) = (1, 1, 1)$ the number of iterations required is less than half that required by the standard implementation using a linear interpolation (Table 8), while for FMG the number of iterations is reduced by a factor of three (Table 9).

(γ, ν_1, ν_2)	Number of Iterations	
	$V_0 = 0$	$V_0 = 30$
(1, 1, 1)	5(11)	16(34)
(1, 2, 2)	5(6)	11(21)
(2, 1, 1)	5(7)	8(-)
(2, 2, 2)	5(6)	6(7)

Table 8: Number of iterations required for Test Problem 1, using the standard FAS implementation, with the prolongation PROL-2 and local relaxations performed near the junction.

(γ, ν_1, ν_2)	Number of Iterations	
	$V_0 = 0$	$V_0 = 30$
(1, 1, 1)	5(7)	6(18)
(1, 2, 2)	5(5)	7(18)
(2, 1, 1)	5(6)	5(6)
(2, 2, 2)	5(4)	4(5)

Table 9: Number of iterations required for Test 1, using the standard FMG implementation, with the prolongation PROL-2 and local relaxations performed near the junction.

The two prolongations just described depend on the difference scheme, and their extension to multidimensional problems is not immediate (though possible). On the other hand, like the currents J_n and J_p the quasi-Fermi variables ϕ_n and ϕ_p are also smooth near the junction, and the transformation from n and p to ϕ_n and ϕ_p is pointwise and hence independent of the space dimension. This suggests a prolongation PROL-3 which interpolates the calculated corrections to n and p as though they were corrections to ϕ_n and ϕ_p .

The scaled variables n and ϕ_n are related by $n = \delta^2 e^{\psi - \phi_n}$, so that $\phi_n = \psi - \ln(n\delta^{-2})$.

Denoting the uncorrected values by n^o and ϕ_n^o , and the corrected values by n^n and ϕ_n^n ,

$$d\phi_n = \phi_n^n - \phi_n^o = \ln\left(\frac{n^o}{n^o + dn}\right) \quad (38)$$

where $d\phi_n$ and dn are the corrections to ϕ_n and n respectively. The corrections to ϕ_n are interpolated linearly at an odd point i , so

$$\begin{aligned} d\phi_{n_i} &= 1/2(d\phi_{n_{i-1}} + d\phi_{n_{i+1}}) \\ &= 1/2 \ln\left(\frac{n_{i+1}^o n_{i-1}^o}{n_{i+1}^n n_{i-1}^n}\right). \end{aligned} \quad (39)$$

This change in ϕ_n produces a new value of n_i :

$$n_i^n = n_i^o \sqrt{\frac{n_{i+1}^n n_{i-1}^n}{n_{i+1}^o n_{i-1}^o}}. \quad (40)$$

The original value of n_i is scaled by a geometric average of the ratio of new to old values at neighbouring points. A similar derivation produces the following interpolation for p

$$p_i^n = p_i^o \sqrt{\frac{p_{i+1}^n p_{i-1}^n}{p_{i+1}^o p_{i-1}^o}}. \quad (41)$$

Since n and p are smoothly varying away from the junctions, PROL-3 uses the simpler linear interpolation throughout most of the domain, with the above relations used only in the neighbourhood of the junction.

Unless local relaxation sweeps were used near the pn -junction to provide extra smoothing of the solution, numerical overflow occurred when PROL-3 was used. With local relaxations the results were much the same as for the standard implementation when using local relaxations (Table 10). PROL-3 reduces the number of iterations needed for the forward biased problem using FAS, but there is no marked difference for either FAS or FMG.

§4.2 Summary of Successful Modifications and Further Testing

The most successful of the modifications described above have been incorporated into one program, SC-1, and this code has been tested for several problems. The main features of the code are listed below.

1. The FMG procedure is used to generate an initial guess on the finest level. A linear interpolation which avoids interpolating across junctions is used to transfer the solution to form the initial iterate at the next finer level.
2. A symmetric, collective Gauss-Seidel relaxation procedure is used. One iteration of Newton's method is used to solve the nonlinear equations, except near junctions where two iterations are used.

(γ, ν_1, ν_2)	Number of Iterations	
	$V_0 = 0$	$V_0 = 30$
(1, 1, 1)	8(11)	18(34)
(1, 2, 2)	6(6)	17(21)
(2, 1, 1)	5(7)	7(-)
(2, 2, 2)	4(6)	6(7)

Table 10: Number of iterations required for Test 1, using the standard FAS implementation, and the prolongation PROL-3, with local relaxations performed near the junction.

(γ, ν_1, ν_2)	Number of Iterations	
	$V_0 = 0$	$V_0 = 30$
(1, 1, 1)	6(7)	16(18)
(1, 2, 2)	5(5)	11(18)
(2, 1, 1)	4(6)	6(6)
(2, 2, 2)	4(4)	4(5)

Table 11: Number of iterations required for Test 1, using the standard FMG implementation, and the prolongation PROL-3 with local relaxations performed near the junction.

3. Local relaxation sweeps are used in the neighbourhood of the junctions.
4. The prolongation PROL-2 is used. While PROL-1 appears to be more effective, it is also more expensive, so the more efficient version has been chosen.
5. The restriction operator is a full weighting scheme, and the same difference operator is used on all grids.

The convergence of SC-1 was studied for three different problems, Test 1 (described in §3.1), Test 2, consisting of the standard diode test problem presented in [31], and Test 3, modelling a thyristor. In scaled form, the doping profile of the standard diode is:

$$C(x) = \begin{cases} -1 & -1 \leq x < 0 \\ 0 & x = 0 \\ +1 & 0 < x \leq 1. \end{cases}$$

Note that, in contrast to Test 1, this represents a *pn*-junction, so applying a positive potential at $x = 1$ is a reverse biasing, while a negative potential is a forward bias. The doping profile which models the thyristor is

$$C(x) = \begin{cases} +1 & -1 \leq x < -0.5 \\ -10^{-3} & -0.5 \leq x < 0 \\ 10^{-5} & 0 \leq x < 0.5 \\ -1 & 0.5 < x \leq 1. \end{cases}$$

The applied potentials ranged from -400 to $+400$ with $\lambda^2 = 1.67 \times 10^{-7}$, $\delta^2 = 1.22 \times 10^{-8}$ and $R \equiv 0$. The convergence criterion used here is stricter than that used in §3.1. The solution was judged to be converged when each component u_j of two successive iterates u^{k+1} and u^k satisfied

$$|u_j^{k+1} - u_j^k| < 10^{-6} |u_j^{k+1}|.$$

For Test 1, with reverse biases of $V_0 = -200$ and $V_0 = -400$, SC-1 converges rapidly (Figure 5). The convergence rate does not significantly change as the size of the finest grid changes (Figure 6).

The convergence rate is also independent of the finest grid size for forward biasing with $V_0 = 200$ (Figure 7). This convergence rate is approximately the same when a larger potential of $V_0 = 400$ is used (Figure 8).

For the reverse biased standard diode (Test 2) with $V_0 = 200$ the convergence of SC-1 is almost immediate, for all grid spacings (Figure 9). SC-1 also converges quickly for very large reverse biases, with $V_0 = 4000$ (roughly 100 Volts). For a small forward biasing, ($V_0 = -40$) there is still rapid convergence but the convergence is slower on the finer grids (Figure 10). This trend is more apparent when the larger forward bias $V_0 = -200$ is used (Figure 11).

The thyristor of Test 3 presents special difficulties since the problem is not always well-conditioned [4] and the solution is not unique — for a given voltage, there are

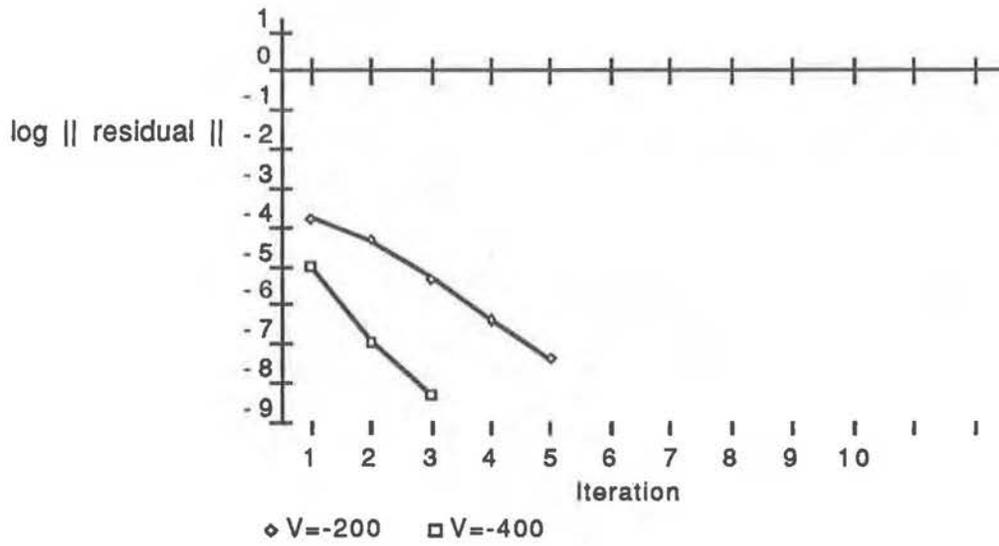


Figure 5: Convergence of SC-1, $V_0 = -200, -400$, $N_f = 64$ (Test 1)

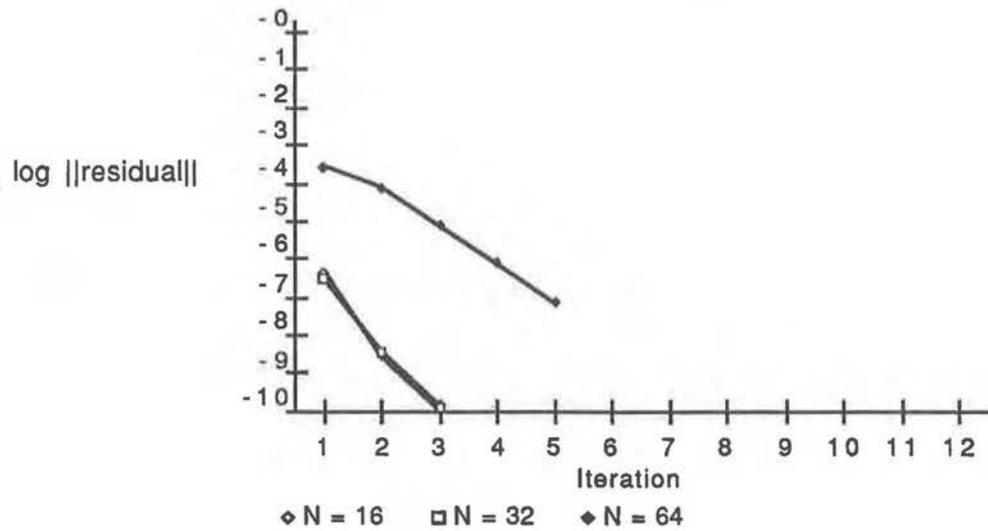


Figure 6: Convergence of SC-1, $V_0 = -200$, $N_f = 16, 32, 64$ (Test 1)

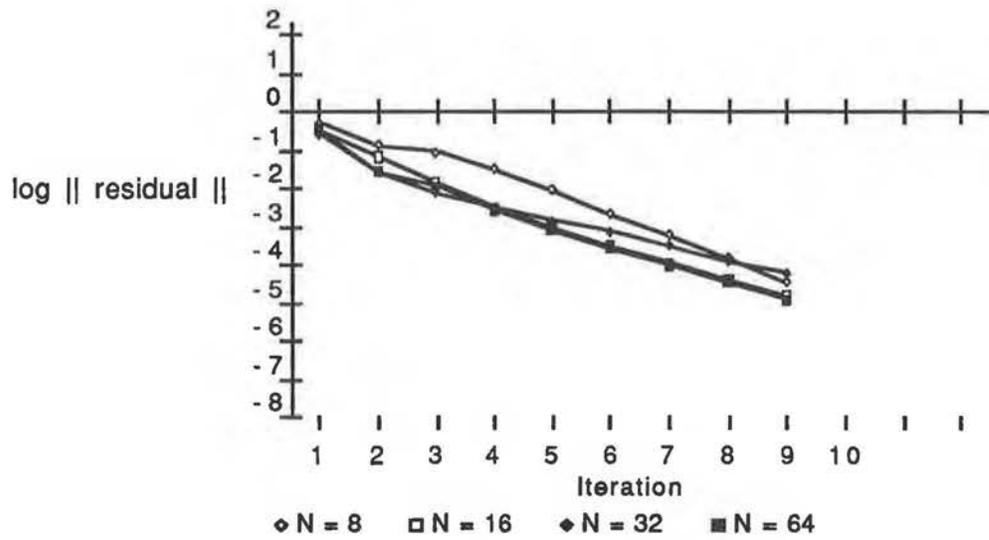


Figure 7: Convergence of SC-1, $V_0 = 200$, $N_f = 8, 16, 32, 64$ (Test 1)

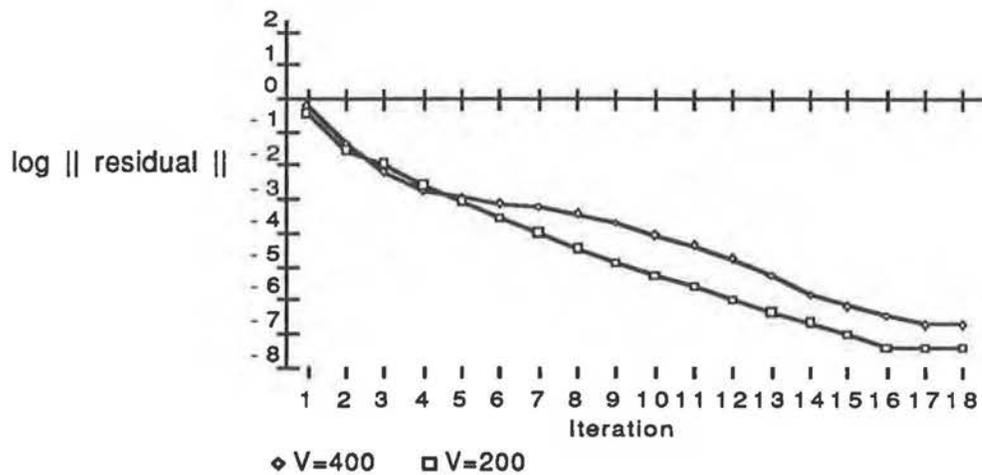


Figure 8: Convergence of SC-1, $V_0 = 200, 400$, $N_f = 64$ (Test 1)

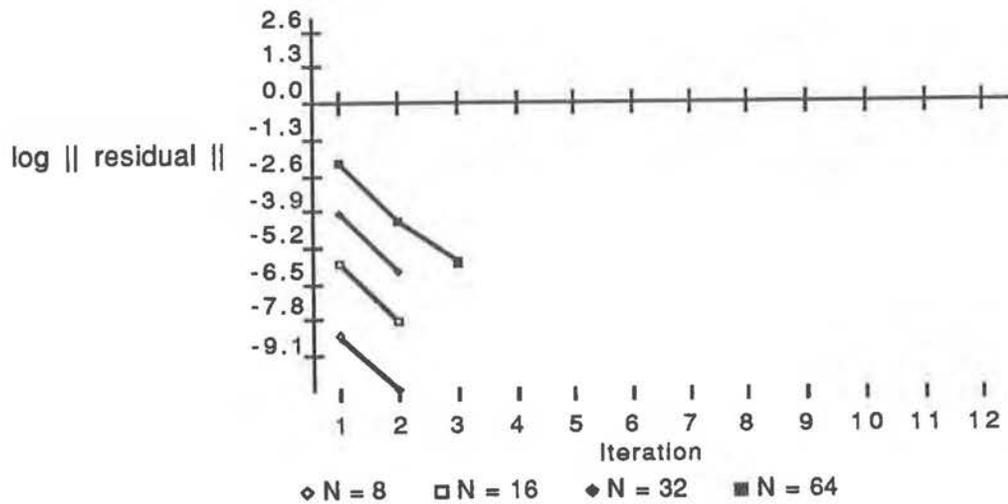


Figure 9: Convergence of SC-1, $V_0 = 200$, $N_f = 8, 16, 32, 64$ (Test 2)

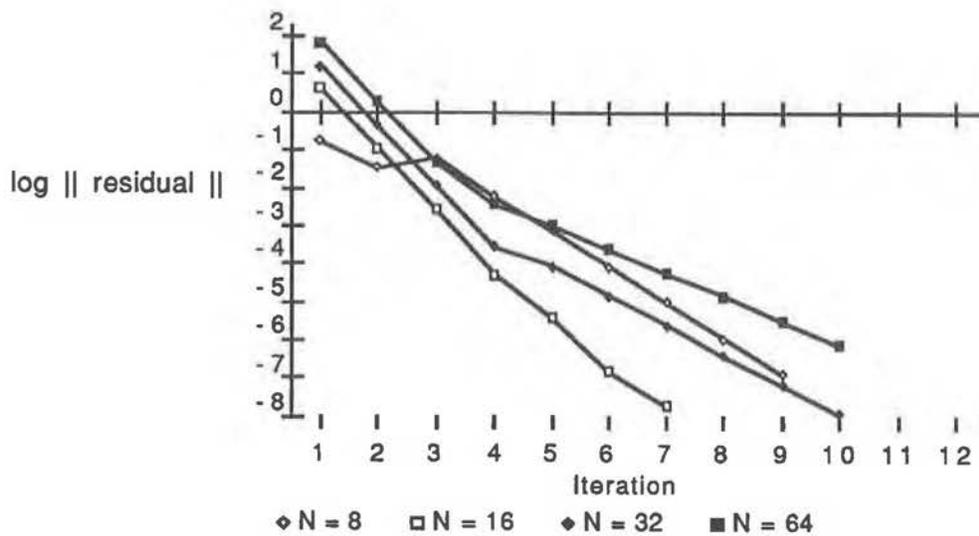


Figure 10: Convergence of SC-1, $V_0 = -40$, $N_f = 8, 16, 32, 64$ (Test 2)

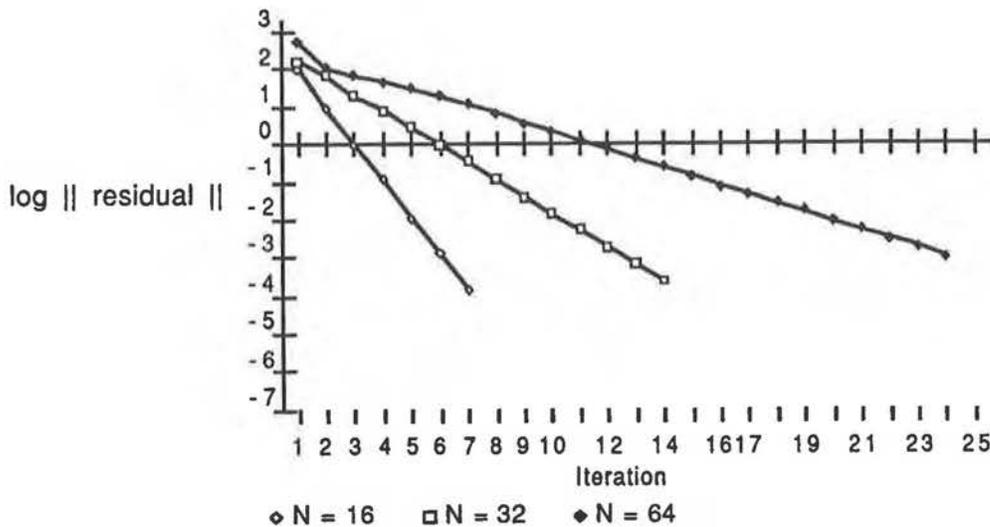


Figure 11: Convergence of SC-1, $V_0 = -200$, $N_f = 16, 32, 64$ (Test 2)

several possible currents which may result. This nonuniqueness allows for a physical phenomenon known as latch-up [39]. To find all solution branches would require that continuation be used, but solutions can be found for relatively small applied potentials. With a forward bias of $V_0 = 40$, SC-1 converges very quickly (Table 12), but for $V_0 = 0$ the convergence is very slow unless extra coarse grid correction steps are used ($\gamma = 2$). The FMG procedure did not converge directly for a reverse biased problem ($V_0 = -50$) because a sufficiently accurate initial guess was not used. The code converged properly when better starting values were obtained by using a simple continuation in terms of the applied potential. Four continuation steps were required to reach $V_0 = -50$.

(γ, ν_1, ν_2)	Number of Iterations	
	$V_0 = 0$	$V_0 = 40$
(1, 1, 1)	39	6
(1, 2, 2)	30	4
(2, 1, 1)	10	5
(2, 2, 2)	8	4

Table 12: Number of iterations required for Test 3, using SC-1.

§4.3 The code SC-1 and Hemker's Algorithm

The only application of the multigrid method to the semiconductor equations that we are aware of which has successfully used very coarse grids in the solution procedure is described by Hemker [20]. A multigrid code based on Hemker's ideas has been implemented and its performance is compared to the program SC-1.

Hemker uses the multigrid method to solve the semiconductor equations in terms of the unscaled quasi-Fermi variables (equations (11) – (12)), with $R \equiv 0$. The prolongation and restriction operators are based on interpolations of the current densities J_n and J_p . To facilitate the construction of appropriate operators a hierarchy of staggered grids is used, as described below.

The standard discretization of the carrier continuity equations (21) – (25) uses expressions for J_n and J_p at grid interval midpoints, which are then expressed in terms of the unknowns at the grid points i and $i+1$. When the multigrid method uses standard coarsening (doubling the grid size on the coarser levels) the positions where J_n and J_p can be estimated change, while the unknowns are calculated at corresponding positions on coarser levels. To facilitate the use of prolongation and restriction operators which are based on J_n and J_p , Hemker uses a series of staggered grids so that J_n and J_p are estimated at corresponding points on the coarse and fine grids. The unknowns are calculated at the midpoints of a uniform grid, allowing J_n and J_p to be estimated at the grid points. In this version the unknowns are calculated at different positions on the different grids, allowing the currents to be known at fixed points.

The interpolation and restriction operators are designed to conserve the estimates of J_n and J_p when transferring between levels. This leads to a piecewise exponential interpolation of ϕ_n and ϕ_p , while linear interpolation is used for prolongating corrections to ψ . The coarse grid operator N_H is the finite difference analog of N_h , but due to the choice of prolongation and restriction operators, N_H is also the Galerkin form of N_h : $N_H = I_h^H N_h I_H^h$. A symmetric, collective Gauss-Seidel relaxation scheme is used, with Newton's method employed to solve the nonlinear equations which arise at each point. The correction transformation technique described in [31] is also used: while the calculations use ϕ_n and ϕ_p , the linearization is performed in terms of n and p . The FMG procedure is used to generate an initial guess on the finest level. Solutions are found on the coarsest level using a combination of techniques. The solution strategy uses a continuation process in terms of the applied potential V_0 , starting from equilibrium and moving in steps to the desired voltage. Each stage of the continuation process uses a combination of a damped Newton's method and Gauss-Seidel relaxations.

To compare Hemker's algorithm to the code SC-1 another code (SC-2), based on Hemker's suggestions, was prepared. SC-2 performs calculations in terms of the carrier concentrations n and p instead of ϕ_n and ϕ_p . The grid transfer operators were changed to correspond to this formulation, so that the new prolongation, restriction and differential operators satisfied

$$N_H = I_h^H N_h I_H^h. \quad (42)$$

This version removed the necessity (experienced with the algorithm of [20]) for damped nonlinear iterations and continuation on the coarsest level for the range of parameters used.

SC-2 was tested on the problems described earlier, describing a standard symmetric diode and a nonsymmetric doping profile. The (scaled) applied potentials ranged from

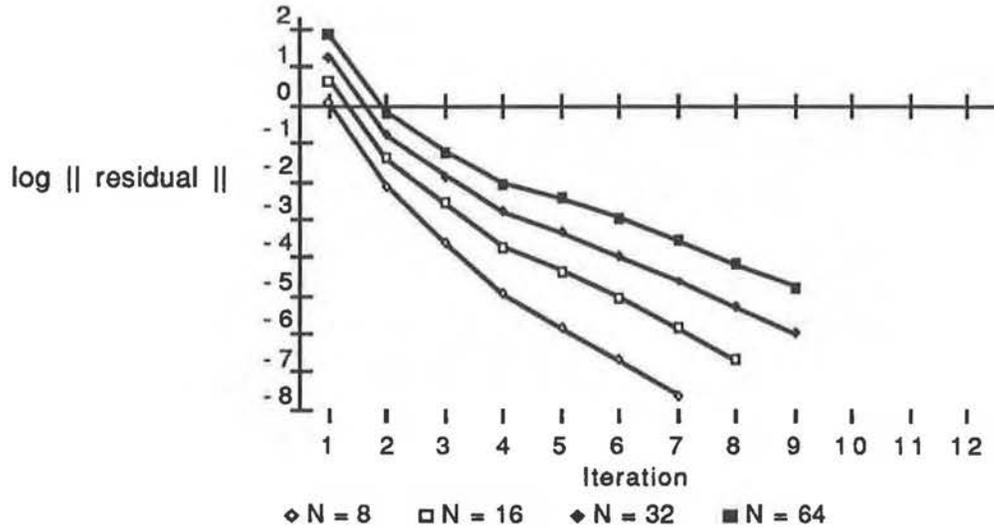


Figure 12: Convergence of SC-2, $V_0 = -40$, $N_f = 8, 16, 32, 64$ (Test 2)

a reverse bias of -400 to a forward bias of $+400$, with $\lambda^2 = 1.67 \times 10^{-7}$, $\delta^2 = 1.22 \times 10^{-8}$ and $R \equiv 0$.

Both programs used the FMG procedure to generate the starting values on the coarsest grids. They are compared on the basis of the reduction of the residual on the finest level only. The program parameters (γ, ν_1, ν_2) were fixed at $(1, 1, 1)$ throughout the testing.

For the symmetric doping profile (Test 2) with a small forward bias ($V_0 = -40$) SC-1 and SC-2 behaved in much the same manner, with the norm of the residual reduced by a constant factor at each iteration (comparing Figures 12 and 10). This convergence rate is independent of the finest grid size. As the magnitude of the applied potential increased the performance of both SC-1 and SC-2 deteriorated as the finest grid spacing decreased (Figures 11 and 13). SC-1 proved to be the more robust code, converging for both $V_0 = -200$ and $V_0 = -400$ with $\gamma = 1$. SC-2 diverged for both cases with $\gamma = 1$; with the more accurate coarse grid solutions obtained using $\gamma = 2$, SC-2 converged for $V_0 = -200$ (Figure 13) but not for $V_0 = -400$. Convergence for the reverse bias cases ($V_0 = 200, 400$) was almost immediate for both SC-1 and SC-2 (Figures 9 and 14), and the convergence rates were independent of the grid size (Figures 9 and 15).

The performance of SC-2 was very different when the nonsymmetric doping profile was used. With a reverse bias ($V_0 = -200$) SC-2 consistently converged but the behaviour was erratic, with the norm of the residual sharply increasing at the first iteration (Figure 16). A similar behaviour was observed for $V_0 = -400$. SC-1 had a much better convergence pattern (Figures 5 and 6). At convergence the residual norm is much smaller when SC-1 is used, compared to SC-2.

SC-1 is more successful than SC-2 for forward bias problems. For $V_0 = 200$ SC-2 converges only when the finest grid is extremely coarse ($N_f = 4$), but SC-1 demon-

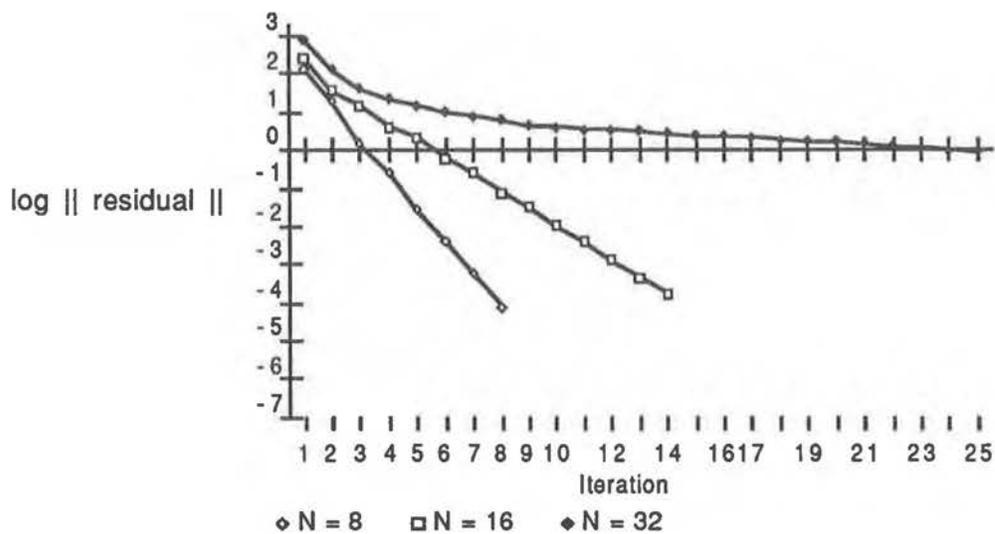


Figure 13: Convergence of SC-2, $V_0 = -200$, $N_f = 8, 16, 32$ (Test 2)

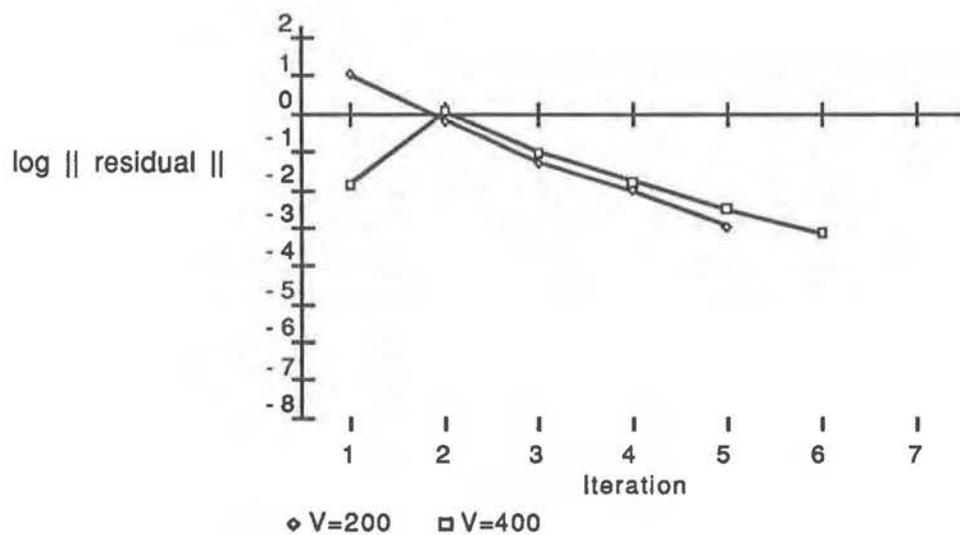


Figure 14: Convergence of SC-2, $V_0 = 200, 400$, $N_f = 64$ (Test 2)

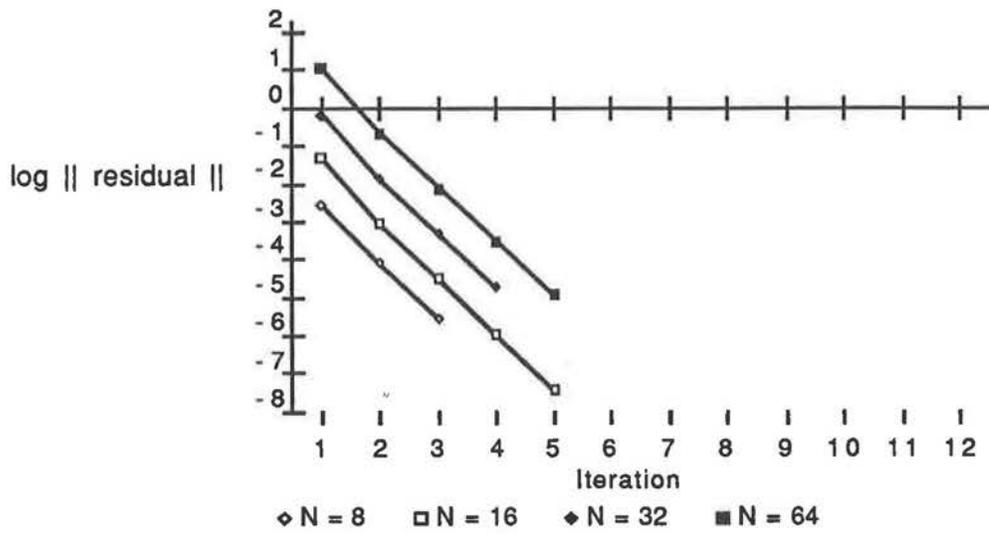


Figure 15: Convergence of SC-2, $V_0 = 200$, $N_f = 8, 16, 32, 64$ (Test 2)

strates a steady convergence rate for all finest grid sizes (Figure 7). SC-1 exhibits the same steady convergence rate for $V_0 = 400$ as well, but SC-2 does not converge in this case at all.

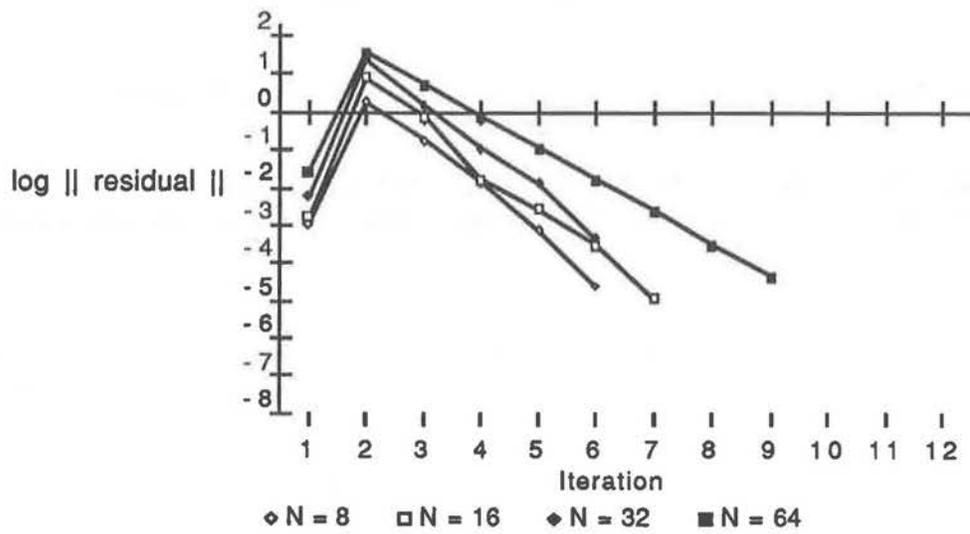


Figure 16: Convergence of SC-2, $V_0 = -200$, $N_f = 8, 16, 32, 64$ (Test 2)

References

- [1] S.E. Adams, "Semiconductor Device Modelling using the Multigrid Method", Master's Thesis, University of British Columbia, 1988.
- [2] D.E. Akcasu, "Convergence properties of Newton's method for the solution of the semiconductor transport equations and hybrid solution techniques for multi-dimensional simulation of VLSI devices", *Solid State Electron.* **27** (1984), 319 – 328.
- [3] R.E. Alcouffe, A. Brandt, J.E. Dendy and J.W. Painter, "The multigrid method for the Diffusion Equations with strongly discontinuous coefficients", *SIAM J. Sci. Stat. Comput.* **2** (1981), 430 – 454.
- [4] U.Ascher, P.A Markowich, C. Schmeiser, H. Steinruck, and R. Weiss "Conditioning of the Steady State Semiconductor Device Problem", *SIAM J. Appl. Math* (in press).
- [5] U.Ascher, R.M.M. Mattheij and R.D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Prentice Hall, 1988.
- [6] R. E. Bank and D. J. Rose, "Parameter selection for Newton like methods applicable to Nonlinear Partial Differential Equations", *SIAM J. Num. Anal.* **17** (1980), 806 – 822.
- [7] R. E. Bank, D. J. Rose and W. Fichtner, "Numerical methods for semiconductor device simulation", *SIAM J. Sci. Stat. Computation* **4** (1981), 416 – 435.
- [8] R. E. Bank, J.W. Jerome and D. J. Rose, "Analytical and numerical aspects of semiconductor device modelling", in *Computing Methods in Applied Sciences and Engineering*, R. Glowinski and J.L. Lions editors, North-Holland (1982).
- [9] A. Brandt, "Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems", in *Proceedings Third International Conference on Numerical Methods in Fluid Mechanics, Paris 1972*, H. Cabannes and R. Teman editors, Springer-Verlag, Berlin, (1973).
- [10] A. Brandt, "Guide to Multigrid Development", in *Multigrid Methods*, W. Hackbusch and U. Trottenberg editors, Springer-Verlag, Berlin, 1982.
- [11] W. Briggs and S. McCormick, "Introduction", in *Multigrid Methods*, S. McCormick, editor, SIAM, 1987.
- [12] E. M. Buturla, P. E. Cottrell, B.M. Grossman, K.A. Salsburg, M.B. Lawlor, C.T. McMullen, "Three Dimensional Finite Element Simulation of Semiconductor Devices", *Proc. Int. Solid-State Circuits Conf.* (1980), 76 – 77.

- [13] A. De Mari, "An Accurate Numerical Steady-State One Dimensional Solution of the PN-Junction", *Solid-State Electron.* **11** (1968), 33 – 58.
- [14] J.E. Dendy, "A priori Local Grid Refinement in the multigrid method", in *Elliptic Problem Solvers II*, G. Birkhoff and A. Schoenstadt, editors, Academic Press, 1984.
- [15] W. Fichtner and D. J. Rose, "On the numerical solution of nonlinear elliptic PDE's arising from semiconductor device modelling", in *Elliptic Problem Solvers*, Academic Press, 1981.
- [16] S.P. Gaur and A. Brandt, "Numerical Solution of semiconductor transport equations in two dimensions by the multigrid method", in *Advances in Computer Methods for Partial Differential Equations II* R. Vichnevetsky editor, 1977.
- [17] G. Golub and C.F. van Loan, *Matrix Computations*, Johns Hopkins University Press, 1983.
- [18] H. K. Gummel, "A Self-consistent iterative scheme for one- dimensional steady state transistor calculations ", *IEEE Trans. Elect. Dev.* **ED-11** (1964), 455 – 465.
- [19] W. Hackbusch, *Multigrid Methods and Applications*, Springer-Verlag, 1985.
- [20] P.W. Hemker "A Nonlinear Multigrid Method for one dimensional semiconductor device simulation: The Diode", manuscript, 1987.
- [21] K. Hwang, D. H. Navon, T. W. Tang and M A. Osman, "Improved convergence of numerical device simulation iterative algorithms", *IEEE Trans. Elect. Dev.* **32** (1985), 1143 – 1145.
- [22] D. Kamowitz, "Multigrid Applied to Singular Perturbation Problems", *ICASE report 87-1* (1987).
- [23] D. Kamowitz and S.V. Parter, "A Study of Some Multigrid Ideas", *Applied Math. and Comp.* **17** (1985) 153 – 184.
- [24] F. K. Manasse, *Semiconductor Electronics Design* , Prentice- Hall, Englewood Cliffs, 1977.
- [25] P.A. Markowich, *The Stationary Semiconductor Equations*, Springer, Wien New York, 1986.
- [26] S.F. McCormick, "An Algebraic Interpretation of Multigrid Methods", *SIAM J. Numer. Anal.* **19** (1982), 548 – 560.
- [27] M.S. Mock, "A Time-Dependent Numerical Model of the Insulated-Gate Field-Effect Transistor", *Solid State Electron* **24** (1981), 959 – 966.

- [28] M.S. Mock, "An example of nonuniqueness of stationary solutions in semiconductor models", *COMPEL* 2 (1983), 165 - 174.
- [29] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, 1970.
- [30] A.R. Mitchell and D.F. Griffiths, *The Finite Difference Method in Partial Differential Equations*, Wiley, 1980.
- [31] S. J. Polak, C. Den Heijer, W. H. A. Schilders and P. A. Markowich, "Semiconductor Modelling from the Numerical Point of View", *Int. J. Num. Meth. Eng.* 24 (1987), 763 - 838.
- [32] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*, Springer, Wien New York, 1984.
- [33] D.L. Scharfetter and H.K. Gummel, "Large Scale Signal Analysis of a Silicon Read Diode Oscillator", *IEEE Trans. Electron Devices* ED-16 (1969), 64 - 77.
- [34] T. I. Seidman, "Steady-state solutions of diffusion reaction systems with electrostatic convection", *Nonlinear Analysis* 4 (1980), 623 - 637.
- [35] R.A. Smith, *Semiconductors, Second Edition*, Cambridge University Press, London, 1978.
- [36] J.W. Slotboom, "Iterative Scheme for 1- and 2-Dimensional D.C.-Transistor Simulation", *Electron. Lett.* 5 (1969), 677 - 678.
- [37] K. Stuben and U. Trottenburg, "Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications", in *Multigrid Methods*, W. Hackbusch and U. Trottenburg editors, Springer-Verlag, 1982.
- [38] K. Stuben, "Algebraic Multigrid (AMG): experiences and comparisons", *Appl. Math. Comput.* 13 (1983), 419 - 451.
- [39] S.M. Sze, *Physics of Semiconductor Devices*, Wiley, 1981.
- [40] W. V. van Roosbroeck, "Theory of flow of electrons and holes in Germanium and other semiconductors", *Bell Syst. Techn. J.* 29 (1950), 560 - 607.
- [41] K. Yamaguchi, "A Time-Dependent and Two-Dimensional Numerical Model for MOSFET Device Operation", *Solid State Electron.* 26 (1983), 907 - 916.