JOURNAL OF COMPUTATIONAL PHYSICS 53, 472–483 (1984)

Numerical Solution of a Coupled Pair of Elliptic Equations from Solid State Electronics

TIMOTHY N. PHILLIPS

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center, Hampton, Virginia 23665

Received January 25, 1983; revised September 9, 1983

Iterative methods are considered for the solution of a coupled pair of second order elliptic partial differential equations which arise in the field of solid state electronics. A finite difference scheme is used which retains the conservative form of the differential equations. Numerical solutions are obtained in two ways—by multigrid and decoupled dynamic alternating direction implicit methods. Numerical results are presented which show the multigrid method to be an efficient way of solving this problem.

1. INTRODUCTION

In solid state electronics the designers of PIN diodes are interested in the effect on the performance of the diode due to changes in various design parameters. An advantage of producing a good mathematical model is that the testing, when performed experimentally, could be a lengthy and expensive process. A description of the model and the derivation of the equations can be found in Aitchison and Berz [2]. The model gives rise to a coupled system of elliptic partial differential equations.

In this paper we consider numerical techniques for the solution of this pair of equations. We consider a two-dimensional diode which is defined in Cartesian coordinates and where it is assumed that the diode is very long in the third dimension. After deriving the finite difference equations we describe applications of a multigrid method and the dynamic ADI (DADI) method to obtain numerical solutions.

2. THE DIFFERENTIAL EQUATIONS

The problem is formulated in terms of the carrier density $c(x, y)$ and a stream function $u(x, y)$. The behavior of diodes which are effectively two dimensional and of rectangular cross section can be described by the equations

This work was supported in part by the Science and Engineering Research Council, United Kingdom, while the author was in residence at Oxford University Computing Laboratory and Merton College, Oxford and in part by the National Aeronautics and Space Administration under Contract NAS1-17070 while he was in residence at ICASE, NASA Langley Research Center, Hampton, Va. 23665.

0021-9991/84 $3.00

$$\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} = c, \tag{1}$$

$$\frac{\partial}{\partial x}\left(\frac{1}{c}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{1}{c}\frac{\partial u}{\partial y}\right) = 0, \tag{2}$$

in the region $R = \{(x, y): 0 \leqslant x \leqslant A, 0 \leqslant y \leqslant B\}$.

The boundary conditions are

$$\frac{\partial c}{\partial x} = \frac{b}{(1+b)}\frac{\partial u}{\partial y}, \qquad \frac{\partial u}{\partial x} = (1+b)\frac{\partial c}{\partial y} \qquad \text{on } x = 0, \tag{3), (4}$$

$$\frac{\partial c}{\partial x} = \frac{-1}{(1+b)}\frac{\partial u}{\partial y}, \qquad \frac{\partial u}{\partial x} = \frac{-(1+b)}{b}\frac{\partial c}{\partial y} \qquad \text{on } x = A, \tag{5), (6}$$

$$\frac{\partial c}{\partial y} = 0, \qquad u = 0 \qquad \text{on } y = 0, \tag{7), (8}$$

$$\frac{\partial c}{\partial y} = -sc, \qquad u = -1, \qquad \text{on } y = B, \tag{9), (10}$$

where s and b are positive constants.

There are two quantities in which the designers of diodes are particularly interested. The first is the equipotential check, $K(y)$, given by

$$K(y) = \frac{-2b}{(1+b)^2}\int_0^A \frac{1}{c}\frac{\partial u}{\partial y}\,dx + \left(\frac{b-1}{b+1}\right)\log\left(\frac{c(0, y)}{c(A, y)}\right) + \log(c(0, y)\cdot c(A, y)). \tag{11}$$

Aitchison [1] showed that this quantity is constant. The second quantity of interest is the total charge, Q, defined by

$$Q = \iint_R c\,dx\,dy. \tag{12}$$

We can easily verify that Q can be expressed in the form

$$Q = 1 - s\int_0^A c(x, B)\,dx. \tag{13}$$

3. Finite Difference Approximation

The finite difference equations are constructed using the integration method of Varga [8]. This method was also used by Aitchison [1] who solved the problem using Newton's method and a sparse matrix routine. The technique is used because the conservative form of Eq. (2) is retained in the finite difference scheme.

We consider a rectangle R in which $A = 2$ and $B = 4$. The region R is covered with a square grid of step size h in both the x and y directions where $Nh = 2$. Let $c_{i,j}$ and $u_{i,j}$ be the values of $c(x, y)$ and $u(x, y)$ at the grid point (x_i, y_j), where $x_i = ih$ and $y_j = jh$. Let the region $r_{i,j}$ be defined as lying within R and being bounded by the lines $x = x_i - \frac{1}{2}h$, $x = x_i + \frac{1}{2}h$, $y = y_j - \frac{1}{2}h$, and $y = y_j + \frac{1}{2}h$. In our application of the technique the regions $r_{i,j}$ are either square or rectangular. Various regions $r_{i,j}$ are shown in Fig. 1. Let $s_{i,j}$ be the boundary of the region $r_{i,j}$.

We first consider Eq. (1). Integrating Eq. (1) over the region $r_{i,j}$ gives

$$\iint_{r_{i,j}} \left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} - c \right) dx\, dy = 0. \tag{14}$$

We apply Green's theorem to the first two terms in this integral to obtain

$$\int_{s_{i,j}} \frac{\partial c}{\partial n}\, ds - \iint_{r_{i,j}} c\, dx\, dy = 0, \tag{15}$$

where n is the unit outward drawn normal.

The finite difference approximation at internal points is therefore given by

$$\left(\frac{c_{i+1,j} - c_{i,j}}{h} \right) h + \left(\frac{c_{i,j+1} - c_{i,j}}{h} \right) h + \left(\frac{c_{i-1,j} - c_{i,j}}{h} \right) h + \left(\frac{c_{i,j-1} - c_{i,j}}{h} \right) h - h^2 c_{i,j} = 0, \tag{16}$$

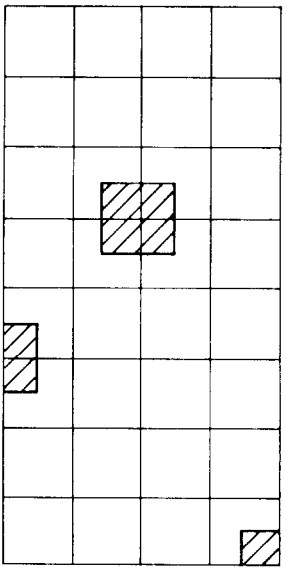

FIGURE 1

where $0 < i < N$, $0 < j < 2N$. This equation can be simplified to give

$$c_{i+1,j} + c_{i-1,j} + c_{i,j+1} + c_{i,j-1} - 4c_{i,j} = h^2 c_{i,j}, \tag{17}$$

which is the same as that obtained by using the standard five-point finite difference approximation to Eq. (1).

To construct the finite difference approximation along $x = 0$ we need to consider the following integral

$$\begin{aligned}\int_{y_{j-1/2}}^{y_{j+1/2}} \frac{\partial c}{\partial x}\, dy &= \frac{b}{(1+b)} \int_{y_{j-1/2}}^{y_{j+1/2}} \frac{\partial u}{\partial y}\, dy \\ &= \frac{b}{(1+b)} \{u(0, (j+\tfrac{1}{2})h) - u(0, (j-\tfrac{1}{2})h)\} \\ &\simeq \frac{b}{(1+b)} \left\{\frac{u_{0,j+1} - u_{0,j-1}}{2}\right\}, \end{aligned} \tag{18}$$

where $0 < j < 2N$. In deriving Eq. (18) we have used boundary condition (3) and the approximation $u(0, (j+\frac{1}{2})h) \simeq (u_{0,j} + u_{0,j+1})/2$. Using Eq. (18) we obtain the following finite difference approximation to Eq. (1) along $x = 0$:

$$2c_{1,j} + c_{0,j+1} + c_{0,j-1} - 4c_{0,j} - \frac{b}{(1+b)} (u_{0,j+1} - u_{0,j-1}) = h^2 c_{0,j},$$

where $0 < j < 2N$. In a similar fashion we can obtain finite difference approximations to Eq. (1) along other parts of the boundary of R.

We now consider the discretization of Eq. (2). Integrating Eq. (2) over the region $r_{i,j}$ yields

$$\iint_{r_{i,j}} \left\{ \frac{\partial}{\partial x}\left(\frac{1}{c}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{1}{c}\frac{\partial u}{\partial y}\right) \right\} dx\, dy = 0. \tag{19}$$

Applying Green's theorem to this equation we obtain

$$\int_{s_{i,j}} \frac{1}{c}\frac{\partial u}{\partial n}\, ds = 0. \tag{20}$$

The finite difference approximation to Eq. (2) at internal points is therefore

$$\frac{u_{i+1,j} - u_{i,j}}{c_{i+1,j} + c_{i,j}} + \frac{u_{i,j+1} - u_{i,j}}{c_{i,j+1} + c_{i,j}} + \frac{u_{i-1,j} - u_{i,j}}{c_{i-1,j} + c_{i,j}} + \frac{u_{i,j-1} - u_{i,j}}{c_{i,j-1} + c_{i,j}} = 0, \tag{21}$$

where $0 < i < N$, $0 < j < 2N$.

To construct the finite difference approximation along $x = 0$ we consider the integral

$$\int_{y_{j-1/2}}^{y_{j+1/2}} \frac{1}{c(0, y)} \frac{\partial u}{\partial x}(0, y)\, dy = (1 + b) \int_{y_{j-1/2}}^{y_{j+1/2}} \frac{1}{c(0, y)} \frac{\partial c}{\partial y}(0, y)\, dy$$

$$= (1 + b)\{\log(c(0, (j + \tfrac{1}{2})h)) - \log(c(0, (j - \tfrac{1}{2})h))\}$$

$$\simeq (1 + b) \log \left(\frac{c_{0,j+1} + c_{0,j}}{c_{0,j-1} + c_{0,j}} \right). \tag{22}$$

In this calculation we have made use of boundary condition (4). So the finite difference approximation to Eq. (20) along $x = 0$ is given by

$$\frac{(u_{1,j} - u_{0,j})}{\frac{1}{2}(c_{1,j} + c_{0,j})} + \frac{(u_{0,j+1} - u_{0,j})}{\frac{1}{2}(c_{0,j+1} + c_{0,j})} \cdot \frac{1}{2} + \frac{(u_{0,j-1} - u_{0,j})}{\frac{1}{2}(c_{0,j-1} + c_{0,j})} \cdot \frac{1}{2}$$

$$- (1 + b) \log \left(\frac{c_{0,j+1} + c_{0,j}}{c_{0,j-1} + c_{0,j}} \right) = 0,$$

where $0 < j < 2N$.

Similarly we obtain finite difference approximations to Eq. (2) along $x = 2$ using the boundary condition given by Eq. (6).

Along $y = 0$ and $y = 4$ we have

$$u_{i,0} = 0 \qquad \text{and} \qquad u_{i,2N} = -1,$$

respectively, where $0 \leqslant i \leqslant N$.

Let K_j be the discrete form of $K((j + \frac{1}{2})h)$ for $j = 0, 1, \ldots, 2N - 1$; $K((j + \frac{1}{2})h)$ is discretized using the trapezoidal rule. The resulting discretization is given by

$$K_j = \frac{-2b}{(1 + b)^2} \sum_{i=0}^{N}{}'' \left(\frac{u_{i,j+1} - u_{i,j}}{\frac{1}{2}(c_{i,j+1} + c_{i,j})} \right) + \left(\frac{b - 1}{b + 1} \right) \log \left(\frac{c_{0,j+1} + c_{0,j}}{c_{N,j+1} + c_{N,j}} \right)$$

$$+ \log\{(c_{0,j+1} + c_{0,j})(c_{N,j+1} + c_{N,j})\}, \tag{23}$$

where the summation notation is defined by

$$\sum_{i=0}^{N}{}'' a_i = \tfrac{1}{2}a_0 + a_1 + \cdots + a_{N-1} + \tfrac{1}{2}a_N.$$

Aitchison [1] shows that K_j is a constant independent of j and so the above difference scheme exactly conserves this constant. To calculate the total charge Q we discretize Eq. (13), again using the trapezoidal rule. Let $\hat{Q}$ be the discrete form of Q, then $\hat{Q}$ is given by

$$\hat{Q} = 1 - sh \sum_{i=0}^{N}{}'' c_{i,2N}. \tag{24}$$

4. A Multigrid Algorithm

We consider a multigrid method of solution to this coupled system of equations using a natural extension of the accommodative full approximation storage (FAS) cycling algorithm of Brandt [3]. The multigrid method is a numerical strategy to solve partial differential equations by switching between finer and coarser levels of discretization. The characteristic feature of the method is the combination of a smoothing step and a coarse grid correction. During the smoothing step the residuals are not necessarily decreased but smoothed. In the following correction step the discrete solution is improved by means of an auxiliary equation on a coarser grid. This results in an iterative method that is usually very fast and effective. A detailed description of the multigrid method can be found in Brandt [3] and Hackbusch [7].

Let $G_1,\ldots, G_M$ be a sequence of grids approximating the region $R = \{(x, y): 0 \leqslant x \leqslant 2, 0 \leqslant y \leqslant 4\}$ with corresponding mesh sizes $h_1,\ldots, h_M$. Let $h_k = 2h_{k+1}$ for $k = 1,\ldots, M-1$. The problem is discretized on each grid G_k using the technique described in the previous section. Let the discrete operators L_1^k and L_2^k define the resulting discretizations of Eqs. (1) and (2), respectively, on G_k, where L_2^k depends on c^k. For $k < M$ we solve an auxiliary equation on G_k (cf. Algorithm 1). The steps of the algorithm are

Algorithm 1

(a) Set $k = M$, the initial working level and choose ε_k to be a suitable tolerance. Choose initial approximations c^k and u^k to c and u, respectively. Set f_1^k and f_2^k equal to zero since $L_1^k c^k = 0$ and $L_2^k u^k = 0$. On coarser grids f_1^k and f_2^k will denote the modified right-hand sides (see Eqs. (25) and (26)).

(b) Set $\bar{e}_k = 10^{30}$.

(c) Perform one relaxation sweep over all the equations. Compute the l_2-norm of the residuals e_k, where

$$e_k = h^{-1}\sqrt{[\|L_1^k c^k - f_1^k\|_2^2 + \|L_2^k u^k - f_2^k\|_2^2]}.$$

(d) If $e_k < \varepsilon_k$, i.e., relaxation has sufficiently converged on the current level, go to step (f). If not, and if convergence is still fast, i.e., $e_k \leqslant \eta \bar{e}_k$, where η is fixed and chosen later, set $\bar{e}_k = e_k$ and go to step (c). The parameter η is known as the switching parameter. If convergence is slow, i.e., $e_k > \eta \bar{e}_k$ and we are not on the coarsest grid go to step (e). If convergence is slow and we are on the coarsest grid go to step (c) to perform another relaxation sweep.

(e) Decrease k by 1. Transfer the current approximations on level $k+1$ to the new level k as

$$c^k = I_{k+1}^k c^{k+1}, \qquad u^k = I_{k+1}^k u^{k+1},$$

where I_{k+1}^k denotes some transfer of values from the fine grid. The right-hand sides for the new level are defined by

$$f_1^k = L_1^k c^k + 4I_{k+1}^k (f_1^{k+1} - L_1^{k+1} c^{k+1}), \tag{25}$$

$$f_2^k = L_2^k u^k + 4I_{k+1}^k (f_2^{k+1} - L_2^{k+1} u^{k+1}). \tag{26}$$

The factor 4 appearing in the above equations is a scaling factor which is introduced because we multiplied through by h^2 before defining the difference operators. Set $\varepsilon_k = \delta e_{k+1}$ to be the tolerance for the problem on the new level where δ is some parameter. Go to step (b).

(f) If $k = M$, the algorithm is terminated since the problem has been solved to the required tolerance. If $k < M$ we correct the approximation on the next finer grid G_{k+1}. Put

$$c^{k+1} = c^{k+1} + I_k^{k+1}(c^k - I_{k+1}^k c^{k+1}),$$
$$u^{k+1} = u^{k+1} + I_k^{k+1}(u^k - I_{k+1}^k u^{k+1}),$$

where the c^{k+1}'s and u^{k+1}'s on the right-hand sides are the previous approximations on the level $k+1$ and I_k^{k+1} is some interpolation of values. Increase k by 1 and go to step (c).

Multigrid Components

We use "nonstandard" multigrid techniques introduced by Foerster, Stuben, and Trottenberg [6] and developed by Foerster and Witsch [5].

(i) *Relaxation.*

Pointwise Gauss–Seidel relaxation is used with the points ordered in the checkerboard (even–odd) manner. The relaxation of the equations is performed in the following order:

(1) relax the equation $L_2^k u^k = f_2^k$ at the white (even) points, i.e., those points (x_i, y_j) for which $i + j$ is even;
(2) relax the equation $L_2^k u^k = f_2^k$ at the black (odd) points, i.e., those points (x_i, y_j) for which $i + j$ is odd;
(3) relax the equation $L_1^k c^k = f_1^k$ at the white points;
(4) relax the equation $L_1^k c^k = f_1^k$ at the black points.

This is just one of a number of ways of performing checkerboard Gauss–Seidel relaxation on these equations. We experimented using several alternatives and found that the above order of relaxation was slightly more efficient than the others. At the end of the relaxation sweep the residuals of the equation $L_1^k c^k = f_1^k$ are zero at the black points since at this stage all the black point equations are simultaneously satisfied.

(ii) *Fine-to-coarse transfer.*

Since checkerboard Gauss–Seidel relaxation produces highly oscillating residuals it is not advisable to simply transfer the residuals by injection to a coarser grid. Instead we transfer the residuals by full-weighting to the coarse grid because the coefficients of Eq. (2) are variable. For this transfer the operator I_k^{k-1} is defined by

$$I_k^{k-1}v_{i,j}^k = \tfrac{1}{4}v_{2i,2j}^k + \tfrac{1}{8}(v_{2i+1,2j}^k + v_{2i-1,2j}^k + v_{2i,2j+1}^k + v_{2i,2j+1}^k) + \tfrac{1}{16}(v_{2i+1,2j-1}^k + v_{2i-1,2j-1}^k + v_{2i+1,2j+1}^k + v_{2i-1,2j+1}^k).$$

(iii) *Coarse-to-fine transfer.*

Bilinear interpolation is used to transfer the correction to the fine grid to provide a new approximation there.

5. A Dynamic ADI Algorithm

We now show how the dynamic ADI (DADI) method of Doss and Miller [4] can be used to obtain a numerical solution to this problem. The ADI approach first converts Eqs. (1) and (2) to the parabolic equations

$$\frac{\partial c}{\partial t} = \frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} - c, \tag{27}$$

$$\frac{\partial u}{\partial t} = \lambda \left\{ \frac{\partial}{\partial x}\left(\frac{1}{c}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{1}{c}\frac{\partial u}{\partial y}\right) \right\}. \tag{28}$$

We assume that as $t \to \infty$ the solution of these time-dependent equations tends to the steady state solution, if one exists. The parameter λ appearing in Eq. (28) is used to control the interaction between the equations. When these equations are discretized in time it means that, effectively, we use different time steps for the two equations. The right-hand sides of Eqs. (27) and (28) are discretized using the technique described earlier. We note that the systems of equations which we solve in this ADI process are tridiagonal and diagonally dominant. This means that the necessary matrix inverses exist and the solution of the systems by Gaussian elimination is stable without the need for interchanges. Detailed discussions of the ADI method and its implementation can be found in Varga [8] and Young [9].

Each step of the DADI method comprises two double sweeps of the ADI iteration with time step Δt together with a bookkeeping double sweep of the ADI iteration with time step $2\Delta t$. At the end of the step we use a computerized strategy to determine how to change Δt for the next step. Here we define a double sweep of the ADI iteration to be a double sweep of ADI performed on Eq. (28) followed by a double sweep of ADI performed on Eq. (27). The method we have described in this section is a form of decoupled DADI method where, in a given iteration, we have preferred to iterate first

on u and then on c. A true DADI method for this system would solve simultaneously for both unknowns. In this case we would solve systems which are block tridiagonal with 2×2 blocks. The procedure we use is described briefly in Algorithm 2.

Algorithm 2

(a) Choose an initial time step $\Delta t = \Delta t_0$. The acceleration parameters are then $h^2/\Delta t$ for Eq. (27) and $h^2/(\lambda \Delta t)$ for Eq. (28). Set $k = 0$, where k is the total number of time steps we have advanced. Let ε be the required tolerance. Choose initial approximations $c^{(k)}$ and $u^{(k)}$.

(b) Start a step of the DADI process with current approximations $c^{(k)}$ and $u^{(k)}$.

(c) Perform two double sweeps of the ADI iteration with time step Δt. Let $c^{(k+4)}$ and $u^{(k+4)}$ be the new approximations obtained. Compute e, the l_2-norm of the residuals of the steady state equations.

(d) If $e < \varepsilon$, then the residuals are sufficiently small and the algorithm is terminated. If $e \geqslant \varepsilon$, set $\Delta t^* = 2\Delta t$ and determine the corresponding acceleration parameters.

(e) Perform a double sweep of the ADI iteration with time step Δt^* starting with approximations $c^{(k)}$ and $u^{(k)}$ to obtain $\tilde{c}^{(k+4)}$ and $\tilde{u}^{(k+4)}$, respectively. This is the bookkeeping part of the DADI process.

(f) Compute the test parameter TP given by

$$\mathrm{TP} = \sqrt{[\mathrm{SUM}/\mathrm{ASUM}]},$$

where

$$\mathrm{SUM} = \|c^{(k+4)} - \tilde{c}^{(k+4)}\|_2^2 + \|u^{(k+4)} - \tilde{u}^{(k+4)}\|_2^2$$

and

$$\mathrm{ASUM} = \|c^{(k+4)} - c^{(k)}\|_2^2 + \|u^{(k+4)} - u^{(k)}\|_2^2.$$

(g) It TP > 0.6, then we reject the present DADI step, replace Δt by $\frac{1}{16}\Delta t$ and go to step (b). If TP $\leqslant 0.6$, then we accept the present DADI step and change Δt by the factor of 4, 2, $\sqrt{3}$, $\frac{1}{2}$, $\frac{1}{4}$ for the next step if TP falls in the intervals $(-\infty, 0.05]$, $(0.05, 0.1]$, $(0.1, 0.3]$, $(0.3, 0.4]$, $(0.4, 0.6]$, respectively. Increase k by 4 and go to step (b). This is the computerized strategy for changing Δt.

6. Numerical Results

When $s = 0$, Eqs. (1) and (2) together with the boundary conditions (3) to (10) possess an analytic solution which is given by

$$c(x, y) = \frac{\{b \cosh(2 - x) + \cosh(x)\}}{4(1 + b)\sinh(2)}, \qquad u(x, y) = -\frac{1}{4}y,$$

TABLE I

Details of the Solution for Different Values of h

	h		
	0.5	0.25	0.125
$c(1, 2)$	0.0890	0.0906	0.0910
$u(1. 2)$	−0.5542	−0.5550	−0.5520
$c(0, 0)$	0.2018	0.2071	0.2086
$c(2, 0)$	0.1203	0.1228	0.1253
$c(0, 4)$	0.0591	0.0682	0.0749
$c(2, 4)$	0.0196	0.0203	0.0209
Q	0.7831	0.7871	0.7884
K	−1.6415	−1.6074	−1.5984

when $A = 2$ and $B = 4$. These functions are used as our initial approximation to the solution of the problem for $s \neq 0$. Numerical results are presented for $s = 5$. The constant b was given the fixed value 2.7.

In the multigrid method we define a work unit to be the computational work in one relaxation sweep over the finest grid. The step size on the coarsest grid is $h = 1$. The values of the parameters η and δ in Algorithm 1 were chosen to be 0.5 and 0.3, respectively. It was found that the choice of η and δ was not critical in the sense that values of these parameters in the neighborhood of the chosen values produced similar efficiency of the algorithm in terms of the number of work units.

The algorithms were terminated when the l_2-norm of the residuals was less than 10^{-6}. The results in Table I indicate the variation with h of the values of c and u at the center of the diode, the values of c at the corners, and the values of the constants Q and K. In Table II we give details of the multigrid method of solution. Details of the DADI method of solution are given in Table III for $\lambda = 1$ and $\lambda = 0.05$. The results were obtained on the Oxford University ICL 2980 computer.

Various values of λ were tried in the DADI method and it was found, by experiment, that the value $\lambda = 0.05$ produced the fastest convergence. It can be seen from the computational details in Table III that this value of λ is considerably better

TABLE II

Details of Multigrid Method

h	Number of Work Units	Time (sec)
0.5	46	1.0
0.25	56	1.9
0.125	65	5.7

TABLE III

Details of DADI Method

	Number of DADI Steps		Time (sec)	
h	$\lambda = 1$	$\lambda = 0.05$	$\lambda = 1$	$\lambda = 0.05$
0.5	84	64	2.3	1.9
0.25	140	98	14.0	9.7
0.125	216	122	155.7	88.6

than $\lambda = 1$ for the smallest mesh size $h = 0.125$. However, even with this value of λ, the multigrid method performs much better than the DADI method on this problem.

The values of the asymptotic convergence factor are a little higher than typical for multigrid methods. Experiments were performed fixing each of the variables in turn to determine how the convergence rate behaves for a single equation. The results of this investigation are given in Table IV. In this table we give the asymptotic convergence factors per work unit for different values of h. We see that for the single equations the typical multigrid rates are realized. Closer examination revealed that the higher convergence factors of the system were due to the coupling through the boundary conditions. A more effective treatment of the boundary conditions in the multigrid context will be the subject of future work.

The methods described here are not restricted to use on square grids. These grids were chosen since there were no advantages in using non-uniform grids for this problem. A qualitative discussion of what one learns about the diode from this solution appears in Aitchison [1].

TABLE IV

Asymptotic Convergence Factors

	h		
	0.5	0.25	0.125
c equation	0.42	0.45	0.46
u equation	0.45	0.35	0.35
system	0.72	0.75	0.77

ACKNOWLEDGMENTS

I am grateful to Joyce Aitchison and David Mayers for introducing me to this problem and for many helpful discussions.

REFERENCES

1. J. M. AITCHISON, *J. Inst. Math. Appl.* (1981), in press.
2. J. M. AITCHISON AND F. BERZ, Solid-St. Electron. **24** (1981), 795.
3. A. BRANDT, *Math. Comput.* **31** (1977), 330.
4. S. DOSS AND K. MILLER, *SIAM J. Numer. Anal.* **16** (1979), 837.
5. H. FOERSTER AND K. WITSCH, On efficient multigrid software for elliptic problems on rectangular domains, University of Bonn, preprint, No. 458, 1981.
6. H. FOERSTER, K. STUBEN, AND U. TROTTENBERG, *in* "Elliptic Problem Solvers" (M. Schultz, Ed.), Academic Press, New York, 1981.
7. W. HACKBUSCH, *Computing* **20** (1978), 291.
8. R. S. VARGA, "Matrix Iterative Analysis," Prentice–Hall, Englewood Cliffs, N.J., 1962.
9. D. M. YOUNG, "Iterative Solution of Large Linear Systems," Academic Press, New York, 1971.