SEMICONDUCTOR DEVICE SIMULATION BY A NEW METHOD OF SOLVING POISSON, LAPLACE, AND SCHRODINGER EQUATIONS

M. J. Sharifi and A. Adibi

Department of Electrical Engineering
Amirkabir University of Technology
Tehran, Iran

(Received: April 14, 1997 - Accepted in Revised form : July 9, 1998)

Abstract In this paper, we have extended and completed our previous work, that was introducing a new method for finite differentiation. We show the applicability of the method for solving a wide variety of equations such as poisson, Laplace and Schrodinger. These equations are fundamental to the most semiconductor device simulators. In a section, we solve the Schrodinger equation by this method in several cases including the problem of finding electron concentration profile in the channel of a HEMT. In another section, we solve the Poisson equation by this method, choosing the problem of SHD as an example. Finally we solve the Laplace equation in two dimensions and as an example, we focus on the VED. In this paper, we have shown that, the method can get stable and precise results in solving all of these problems. Also the programs which have been written based on this method become considerably faster, more clear, and more abstract.

Key Words Finite Difference Method, Laplace Equation, Poisson Equation, Schrodinger Equation, Shotcky Barrier Diode, Vacuum Electronic Devices

INTRODUCTION

The study of complicated systems using computer simulation has become a growing task since the development of computers. In the electronic device domain, in most cases, computer simulation means numerical solutions of partial differential equations with boundary value conditions. There are two usual numerical methods for solving these boundary value problems, namely, FD method and FE method [1]. Although both methods have been used for electronic device simulation, but the FD method is more conventional (See References 2 and 3 for a general review, References 7 to 11 for FD method and Ref-
Many authors have tried to modify these formulas. An interesting work is introduced in Reference 12.

We have used the formulas that come from fitting a polynomial to objective function. An $N$ point formula uses an $N-1$ degrees polynomial. For example, for a 5 point formula, we have:

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$$

For the moment, we assume that we want the derivatives in the point $x_0 = 0$. Also we take all $\Delta x_i$ to be the same (of course the results, are general and not restricted to these limitations). Several 5 point formulas can be obtained based on various possible arrangements of the points in the environment of the objective point. If, for example, we want one point to be on the left and three points to be on the right of the objective point, we have:

$$\ldots x_9 \ldots x_0 \ldots \Rightarrow$$

$$\begin{bmatrix}
1 & -1 & 1 & -1 & 1 \\
1 & \theta & \theta & \theta & \theta \\
1 & 1 & 1 & 1 & 1 \\
1 & 2 & 4 & 8 & 16 \\
1 & 3 & 9 & 27 & 81
\end{bmatrix}
\begin{bmatrix}
a_0 \\
a_1 \\
a_2 \\
a_3 \\
a_4
\end{bmatrix}
= \begin{bmatrix}
y(-1) \\
y(1) \\
y(2) \\
y(3)
\end{bmatrix}$$

(5)

After solving these equations, the coefficients $a_0$ to $a_4$ are obtained and we get the following formulas for finite differentiation:

$$y'(x_0) = a_1 = \frac{-1}{12 \Delta x} \left[ -3 y(x_0 - \Delta x) - 10 y(x_0) + 18 y(x_0 + \Delta x) - 6 y(x_0 + 2\Delta x) + y(x_0 + 3\Delta x) \right]$$

(6)

$$y'(x_0) = 2a_2 = \frac{1}{12 \Delta x^3} \left[ 111 y(x_0 - 2\Delta x) - 20 y(x_0) + 6 y(x_0 + 2\Delta x) + 4 y(x_0 + 2\Delta x) - y(x_0 + 3\Delta x) \right]$$

(7)

The accuracy of conventional formulas is very poor not only relatively (with the same $\Delta x$) but also abso-

1. NEW FORMULAS FOR FINITE DIFFERENTIATION

The FD method traditionally uses these well known formulas for finite-difference approximation of derivatives [1].

$$\ldots x_9 \ldots \Rightarrow$$

$$\begin{align*}
\frac{dy}{dx} &= \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \\
\frac{d^2 y}{dx^2} &= \frac{f(x_0 + 2\Delta x) - 2f(x_0) + f(x_0 - \Delta x)}{\Delta x^2}
\end{align*}$$

(1)

(2)

(3)

(4)
ultely (with any $\Delta x$) when they are compared with the above formulas.

By the various formulas discussed above at hand, we can construct a number of matrix operators for solving differential equations in the finite difference approximation [4]. In the finite difference approximation of differential equations, we discretize the equation by dividing its domain into $N$ (equal or not equal space) pieces. For example if our main equation be:

$$y'' + a(x)y = \beta(x)$$ \hspace{1cm} (8)

Then after discretization, we have a set of $N$ coupled equations as:

$$y''(x_i + i\Delta x) + a(x_i + i\Delta x)y(x_i + i\Delta x) = \beta(x_i + i\Delta x)$$

$$i = 1, 2, \ldots, N \hspace{1cm} (9)$$

Now we wish to have a matrix operator for the second order differentiation. These operators have the following general form:

$$y'' \rightarrow Ay + An \hspace{1cm} (10)$$

"$A$" is the matrix operator for the second order differentiation and "$An$" is a vector which contains initial conditions. Suppose, for example, we know $y_0$ and $y_{N+1}$ (Dirichlet Boundary conditions) and we want to use 5 point formulas. For the first equation, ($i = 1$) we must use a formula which contains the points numbered 0,1,2,3 and 4. This means that it is a formula with one point on the left and 3 points on the right of the objective $i = 1$ point. It is the formula numbered (7). For the second equation, ($i = 2$) we must use a formula which contains the points 0,1,2,3 and 4. This means that it is a formula with 2 points on the left and 2 points on the right of the objective $i = 2$ point. These two equations have a contribution to initial value vector "$An$". Other equations from $i = 3$

to $i = N-2$ use the formulas with 2 points on the left and 2 points on the right of the objective point but have no contribution to the vector "$An$". Equations $i = N-1$ and $i = N$ are symmetric form of the equations $i = 2$ and $i = 1$ which were discussed above and also have a contribution to "$An$" corresponding to the boundary value in $y_{N+1}$. If we take all $\Delta x$ equal, the matrix operator "$A$" and initial value vector "$An$" have the following form:

$$A = \frac{1}{12 \Delta x^2} \begin{bmatrix}
20 & 6 & 4 & -1 & 0 & 0 \\
16 & -20 & 16 & -1 & 0 & 0 \\
-1 & 16 & -30 & 16 & -1 & 0 \\
0 & -0 & -1 & 16 & -30 & 16 \\
0 & 0 & -1 & 16 & -30 & 16 \\
0 & 0 & 0 & -1 & 4 & 6 & -20
\end{bmatrix}$$

$$An = \frac{1}{12 \Delta x^2} \begin{bmatrix}
y(0) \\
\vdots \\
y(N+1)
\end{bmatrix}$$

(11)

In case of Neumann boundary conditions [13], we can find, from the first derivative formulas, the necessary relation at the boundaries between the function’s value and its derivatives. All other things are the same as Dirichlet boundary conditions.

2. NUMERICAL SOLUTION OF THE SCHRODINGER EQUATION

The Schrödinger equations is the most important equation for modern nanoscale semiconductor devices.

$$\frac{\hbar^2}{2m} \Delta \varphi + V \varphi = E \varphi$$
At the first step, we solve this equation for the most simple case, \(V=0\) in vanishing Dirichlet boundary conditions (correspond to infinite quantum well). In this case, the equation has exact analytical solutions which we can compare our results with them. In Figure 1 we show the calculated eigenvalue energies (normalized with \(\hbar^2/2m\)) based on our method using the 5-point and 7-point formulas in comparison with the conventional formulas and also the exact solutions. Clearly the accuracy of our formulas is better than the conventional formulas, with the exact solutions as reference.

After the above demonstration of accuracy, as a more realistic problem, we consider electron confinement in the channel of a HEMT [6]. In this structure, a triangular shape potential well, which is formed at a GaAs–AlGaAs interface, confines electrons. Figure 2 shows the potential profile used in our calculation (same as Reference 15). This potential profile must be added to the diagonal terms of the matrix operator for the second order differentiation.

Figure 3 shows the calculated results for the ground state (consistent with Reference 15(Figure2)) and two exited states of this system.

3. NUMERICAL SOLUTION OF THE POISSON EQUATION

Every device simulator needs to solve the Poisson equations.

\[
\nabla^2 V = - \frac{q}{\epsilon} (N^\rho - N_n + p-n)
\]

This equation was solved using several methods (for example with FD in Reference 11 and with FE in Reference 14 and in several devices before. In this section we want to solve it using our method in one dimension for a SBD [6] in the equilibrium state. In this case, due to dependency of electron concentration to the potential profile, we encounter a nonlinear second degree differential equation with Dirichlet conditions.

![Figure 1. Accuracy of various formulas.](image1)

![Figure 2. The potential profile.](image2)

![Figure 3. The calculated results for the ground state.](image3)
boundary conditions:

\[ \frac{\partial V^2}{\partial x^2} = -\frac{q}{e}(N_i - N_e e^{(\phi_f + \psi)/kT}) \]

The following small and very interesting program can do this task. It is very clear and abstract due to the usage of full matrix calculation and high precision formulas. These features eliminate the need to any mesh space adjustment and normalization procedure which are necessary in other routines [2,3,11].

% This Program Solves Equilibrium State of Shoc tky Barrier Diode
for i = 1:50
n = n0*exp(t*ef + v-eg/kt); % Electron profile
f = q/(epsr*eps0*epsr)*9*ndi-n)+a*bound;
% N-R Objective Function
fnorm = f^2; % Norm of Error
pause ('Press Ctrl + Break if the error is sufficiently small')
df = a*(q/(eps0*epsr)^2/20,20)-diag(n,0))/v; % Derivative of f
v = v-inv(df^2); % New Value for Potential Profile
end

Physical, material and device constants like \( \epsilon_{0} \), \( \epsilon_{psr} \), \( \psi \), \( \epsilon_{r} \), \( \epsilon_{q} \), \( \text{nd} \) [3] and also matrix operator \( 'a' \) and boundary condition vector \( 'vbound' \) in addition to an initial estimation of potential profile must be previously defined for the program. Usually less than 10 iterations are needed for the program to be converged. Figure 4 shows the calculated potential profile as the result.

### 4. NUMERICAL SOLUTION OF THE LAPLACE EQUATION

In this section we solve the Laplace equation in two dimensions. Boundary conditions are so designed that the device be a VED. The device's shape is shown in

Figure 4. Equilibrium potential profile in SBD.

Figure 5. Here, we can see another important feature of the method. Conventional algorithms based on the FD method, often use uniform or nonuniform rectangular meshes [3]. This type of discretization generates many unnecessary points (variables) and therefore terminating in huge matrices. In our method, having the various formulas for finite differentiation at hand, we can eliminate, as can be seen from the figure, unnecessary points. This elimination reduces the required memory space and increases the speed, although retains the matrix sparisty. We use only 35 points for this simulation. The obtained results are shown in Figure 6.

**CONCLUSION**

In this paper we review our previously published
paper about introducing new formulas for finite differentiation and then, apply them to the most important equations in the area of electronic device simulation. These are schrodinger, Poisson, and Laplace equations. We choose as illustrative examples, quantum well structures, SBD, and VED. Each one of these three examples shows one of the three main features of the method. The higher accuracy is demonstrated in the first example while the ability of making programs more abstract and clear is demonstrated in the second example. In the third example, we exhibit the speeding up and the required memory reducing features of the method. In each case this method has obvious superiority over conventional algorithms without almost any additional cost.

**ABBREVIATIONS**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE</td>
<td>Finite Element</td>
</tr>
<tr>
<td>FD</td>
<td>Finite Difference</td>
</tr>
<tr>
<td>HEMT</td>
<td>High Electron Mobility Transistor</td>
</tr>
<tr>
<td>SBD</td>
<td>Shottky Barrier Diode</td>
</tr>
<tr>
<td>VED</td>
<td>Vacuum Electron Device</td>
</tr>
</tbody>
</table>

**REFERENCES**

5. MATLAB is a general purpose software for matrix calculation.