JOURNAL OF COMPUTATIONAL PHYSICS 99, 79–83 (1992)

A Direct Solution of Poisson's Equation in a Three-Dimensional Field-Effect Transistor Structure

DONALD REID, ANDREW CHAN, AND MUSTAFA AL-MUDARES

Department of Electrical and Electronic Engineering, The University, Glasgow G12 8QQ, Scotland

Received September 19, 1990

To solve Poisson's equation in a three-dimensional field-effect transistor structure we use the "finite strip" method. This allows us to treat the problem as a set of two-dimensional planes which can be analysed in parallel. We describe a method to treat each plane using a direct algorithm and report the comparative timing of two-dimensional and three-dimensional models.

1. INTRODUCTION

In many branches of science and engineering which model particles interacting with a field, it may be necessary to numerically solve elliptical partial differential equations (PDEs) such as Poisson's equation in a three-dimensional (3D) structure [1]. A common example is found in semiconductor device simulations where the electrostatic potential must be repeatedly calculated for varied charge distributions in a given physical structure until a steady state solution is found.

In programs such as WATMOS [2] or CADDETH [3] which model metal-oxide-semiconductor transistors, iterative matrix methods such as successive over relaxation (SOR) or incomplete cholesky–conjugate gradient (ICCG) are used [4, 5]. For simple planar transistor structures (Fig. 1) such as metal-semicondoctor field-effect transistors (MESFETs) we can take advantage of the boundary conditions and use direct (i.e., non-iterative) methods.

In this paper we describe a technique analogous to the "finite strip" method [6], commonly used in civil engineering, which transforms the 3D structure into a set of two-dimensional (2D) planes. We then describe how these planes can be treated in parallel and the quasi-2D solutions reconstituted into a 3D solution. We use a model problem of a three-electrode FET with simple boundary conditions. The techniques described, however, are more generally applicable.

2. 3D ALGORITHM

Poisson's equation is defined in 3D as

$$\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial y^2}+\frac{\partial^2\phi}{\partial z^2}=\rho, \tag{1}$$

where, in the case of semiconductor devices, ϕ is the electrostatic potential and ρ is the charge density (defined in suitable units).

For our model problem the region's shape is uniform in the z-dimension and so the electrostatic potential can be described by

$$\phi=\sum_{m=1}^{r} f_m(x, y)\, \Pi_m(z), \tag{2}$$

where $\Pi_m(z)$ are a set of orthogonal functions. In our case we will take advantage of the uniform discretization and simple boundary conditions and use a real fast Fourier transform.

2.1. *Boundary Conditions for FET Structures: Fig. 1*

Our aim is to solve Poisson's equation for the finite block of semiconductors shown in Fig. 1. At the three electrodes the electrostatic potential is fixed. In the interelectrode regions, the change in relative permittivity at the semiconductor air interface causes a first-order approximation, the electric field (i.e., the gradient of the electrostatic potential) normal to the surface to be zero. A similar argument implies that the electric fields at $z=0$ and $z=c$ are also zero. The planes $x=0$, $x=a$, and $y=0$ are assumed to be far enough from the changing electrode potentials for the electric field to have fallen to zero. Dimensions a and b can always be adjusted until this is the case. Thus in all planes other than the top plane we use Neuman boundary

0021-9991/92 $3.00

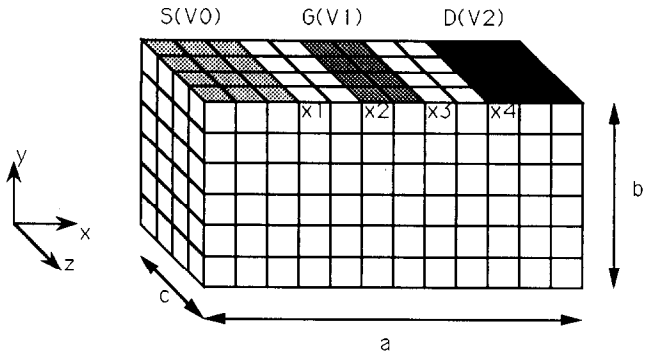

FIG. 1. 3D FET Structure.

conditions. In the top plane a mix of Neuman and Dirichlet boundary conditions are used. The boundary equations may be written:

$$\frac{\partial \phi}{\partial x} = 0 \qquad \text{at} \quad x = 0 \text{ and } x = a, \tag{3a}$$

$$\frac{\partial \phi}{\partial z} = 0 \qquad \text{at} \quad z = 0 \text{ and } z = c, \tag{3b}$$

$$\frac{\partial \phi}{\partial y} = 0 \qquad \text{at} \quad y = 0 \tag{3c}$$

$$\phi = V_0 \qquad \text{at} \quad y = b \text{ and } x \in (0, x_1) \tag{3d}$$

$$\phi = V_1 \qquad \text{at} \quad y = b \text{ and } x \in (x_2, x_3) \tag{3e}$$

$$\phi = V_2 \qquad \text{at} \quad y = b \text{ and } x \in (x_4, a) \tag{3f}$$

$$\frac{\partial \phi}{\partial y} = 0 \qquad \text{at} \quad y = b \text{ and } x \in (x_1, x_2) \text{ or } x \in (x_3, x_4). \tag{3g}$$

2.2. *Decomposing the 3D Structure into a Set of 2D Planes*

In order to solve Poisson's equation numerically we first discretise it. To make the resulting calculation easier we use a mesh which ranges from 0–2^{ξ}, where ξ is an integer in the x-dimension, with analogous ranges in the y- and z-dimensions. The edges of the grid are chosen to be coincidental with the edges of the structure by judicious choice of the mesh spacing.

Using i, j, and k as the counting integers and Δ_x, Δ_y, and Δz as the mesh spacing in the x, y, and z dimensions, respectively, Poisson's equation becomes

$$\frac{\phi_{i+1,j,k} - 2\phi_{i,j,k} + \phi_{i-1,j,k}}{\Delta_x^2} + \text{similar terms in } j \text{ and } k = \rho_{i,j,k}. \tag{4}$$

We use a real fast Fourier transform in the z-dimension to uncouple (4). (The z-dimension is chosen because the geometry of the region is uniform in this dimension.)

Then

$$\bar{\phi}_{i,j,\kappa} = \frac{2}{N_z} \sum_{k=0}^{N_z} E(k)\, \phi_{i,j,k} \cos\left(\frac{\pi k \kappa}{N_z}\right), \qquad E(k) = \begin{cases} \frac{1}{2} & (k = 0, N_z) \\ 0 & \text{otherwise,} \end{cases} \tag{5a}$$

where

$$\phi_{i,j,k} = \sum_{\kappa=0}^{N_z} E(\kappa)\, \bar{\phi}_{i,j,\kappa} \cos\left(\frac{\pi k \kappa}{N_z}\right) \tag{5b}$$

and N_z is the number of mesh spacings in the z-dimension. $\rho_{i,j,k}$ is transformed to $\bar{\rho}_{i,j,\kappa}$ in the same way.

2.3. *Poisson's Equation in a 2D Transform Plane*

By substituting (b) into (4) we obtain an equation which is invariant in the third dimension. Thus each equation is effectively 2D for a given "plane number" κ:

$$\frac{\bar{\phi}_{i+1,j,\kappa} - 2\bar{\phi}_{i,j,\kappa} + \bar{\phi}_{i-1,j,\kappa}}{\Delta_x^2} + \text{term in } j - \bar{\phi}_{i,j,\kappa} \frac{\sin^2(\kappa\pi/2N_z)}{\Delta_z^2} = \rho_{i,j,\kappa}. \tag{6}$$

(For further details see, for example, Ref. [4].)

In vector notation this can be written as

$$\underline{\phi}_{j+1} + \bar{A}\underline{\phi}_j + \underline{\phi}_{j-1} = \Delta_y^2 \underline{\rho}_j = \underline{q}_j, \tag{7a}$$

where $\underline{q}_j$ is the modified charge distribution; $\bar{A}$ is a tridiagonal matrix given by

$$\bar{A} = \text{tri}[\alpha^{-1}, -2(1 + \alpha^{-1} + \alpha^{-1}\beta), \alpha^{-1}]; \tag{7b}$$

and the components α and β are given by

$$\alpha = \left(\frac{\Delta_x}{\Delta_y}\right)^2, \qquad \beta = \frac{1}{2}\left(\frac{\Delta_x \sin(\kappa\pi/2N_z)}{\Delta_z}\right)^2. \tag{7c}$$

2.4. *Boundary Conditions of 2D Plane*

We can look at the effect of the Fourier transform (FT) on the equations governing the boundary conditions. At the boundaries where the electric field is zero,

$$\frac{\partial \phi}{\partial n} = 0, \tag{8a}$$

where n is normal to the plane. By using cosines in the FT

we automatically satisfy this condition at $z=0$ and $z=c$. On substituting (5) into (8a) we obtain

$$\frac{\partial \bar{\phi}_\kappa}{\partial n}=0 \qquad \text{for each } \kappa. \tag{8b}$$

By substitution we can also find the equation at the contacts:

$$\bar{\phi}_\kappa=\frac{2}{N_z}\sum_{k=0}^{N_z} V\cos\left(\frac{k\kappa\pi}{N_z}\right)=2V \qquad (\kappa=0)$$
$$=0 \qquad (\kappa\neq 0). \tag{8c}$$

As the transformed equation and the boundary conditions are uncoupled for a given value of κ we can consider the problem as a set of 2D planes which can be solved *in parallel.*

3. SOLUTION TO THE QUASI-2D POISSON EQUATION

In order to solve these problems as quickly as possible we have modified the well-known FACR algorithm. The essence of this program is a combination of two direct methods: Fourier analysis (in our case the FFT) and cyclic reduction. By including an initial odd/even reduction we optimise the algorithm for maximum speed; and by including the capacity matrix technique we can deal with the mixed boundary conditions found when incorporating electrodes.

In the following section we give a brief account of the FACR algorithm and summarise the changes required to implement the quasi-2D Poisson equation. Essentially these are changing the constants (in the 2D algorithm) which depend on κ. (See Fig. 2 for a comparison between 2D and 3D. For further details of the FACR algorithm see, for example, Ref. [1, 4, 7].)

3.1. *Initial Odd/Even Reduction*

We start by reducing the full set of equations to the even lines only by combining three adjacent equations:

$$\underline{\phi}_{j-2}+\bar{A}\underline{\phi}_{j-1}+\underline{\phi}_j=\underline{q}_{j-1} \tag{9a}$$
$$\underline{\phi}_{j-1}+\bar{A}\underline{\phi}_j+\underline{\phi}_{j+1}=\underline{q}_j \tag{9b}$$
$$\underline{\phi}_j+\bar{A}\underline{\phi}_{j+1}+\underline{\phi}_{j+2}=\underline{q}_{j+1}, \tag{9c}$$

so that

$$\underline{\phi}_{j-2}+(2\bar{I}-\bar{A}^2)\,\underline{\phi}_j+\underline{\phi}_{j+2}=\underline{q}_j^* \tag{9d}$$

and

$$\underline{q}_j^*=\underline{q}_{j-1}+\underline{q}_{j+1}-\bar{A}\underline{q}_j. \tag{9e}$$

(Here $\bar{I}$ is the unit matrix.)

In the program, the subroutine which calculates (9e) will be altered using this procedure, since $\bar{A}(\kappa)$ is different for each plane.

3.2. *Fourier Analysis*

After Fourier analysis (9d) is given by

$$\bar{\phi}_{j-2}^{l}+\lambda_l\bar{\phi}_j^{l}+\bar{\phi}_{j+2}^{l}=\bar{q}_j^{l}, \tag{10}$$

where $\lambda_l(\kappa)$ is the eigenvalue of the matrix $(2\bar{I}-\bar{A}^2)$.

Note that the eigenvalues of a simple tridiagonal matrix tri$[1, 0, 1]$ are $2\cos\theta_l$, where $\theta_l=\pi l/N_x$, $0\leqslant l\leqslant N_x$. This is because

$$[1, 0, 1]\begin{Bmatrix}\sin(i\theta_l)\\ \cos(i\theta_l)\end{Bmatrix}=2\cos\theta_l\begin{Bmatrix}\sin(i\theta_l)\\ \cos(i\theta_l)\end{Bmatrix},$$
$$0\leqslant i\leqslant N_x. \tag{11a}$$

Therefore the eigenvalues of matrix $\bar{A}(\kappa)$ are

$$\mu_l(\kappa)=2\alpha^{-1}\cos\theta_l-2(1+\alpha^{-1}+\alpha^{-1}\beta); \tag{11b}$$

and the eigenvalues of matrix $(2\bar{I}-\bar{A}(\kappa)^2)$ are

$$\lambda_l(\kappa)=2-\mu_l(\kappa)^2. \tag{11c}$$

The sets of eigenvalues $\lambda_l(k)$ are calculated in an initial setup subroutine.

3.3. *Recursive Cyclic Reduction*

From (10) we can reduce the set of equations to the tth level, such that

$$\phi_{j-2^t}+\lambda^{(t)}\phi_j+\phi_{j+2^t}=q_j^{(t)},$$
$$j=2^t \text{ step } 2^t \text{ until } N_y-2^t, \tag{12a}$$

where

$$\lambda^{(t+1)}=2-(\lambda^{(t)})^2 \tag{12b}$$

and

$$q_j^{(t+1)}=q_{j-2^t}^{(t)}+q_{j+2^t}^{(t)}-\lambda^{(t)}q_j^{(t)}. \tag{12c}$$

This finally results in a pair of simultaneous equations which give the top and bottom lines:

$$2\phi_0 + \lambda^{(T)}\phi_{N_y} = q^{(T)}_{N_y} \tag{13a}$$

$$2\phi_{N_y} + \lambda^{(T)}\phi_0 = q^{(T)}_0. \tag{13b}$$

Hence we can find ϕ_0 and ϕ_{N_y}.

The intermediate values can be deduced from a backwards recursion:

$$\phi_j = (q^{(t)}_j - \phi_{j-2^t} + \phi_{j+2^t})/\lambda^{(t)}, \quad j = 2^t \text{ step } 2^t \text{ until } N_y - 2^t. \tag{14}$$

3.4. *Fourier Synthesis*

Solutions of potentials on the even lines can be found by forming the Fourier transform of the harmonic amplitudes.

3.5. *A Solution on the Odd Lines*

By combining the original equation with the solution already found on the even lines, a solution can be deduced for the odd lines

$$[1, -2(1+\alpha+\beta), 1]\,\underline{\phi}_j = \underline{q}_j - \alpha\underline{\phi}_{j-1} - \alpha\underline{\phi}_{j+1}. \tag{15}$$

The routine which calculates the left-hand side of (15) has to be altered to take account of the extra term β.

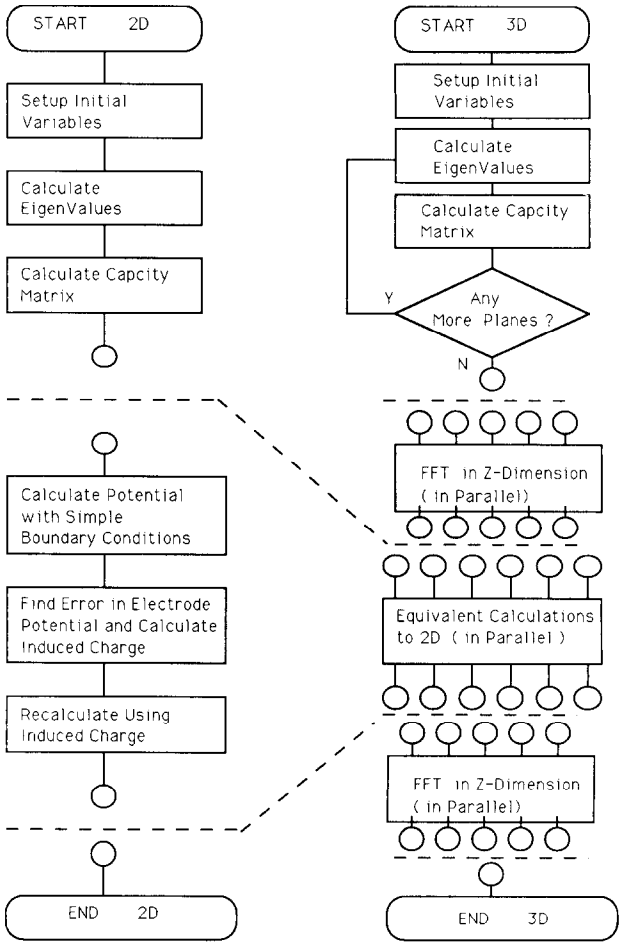

FIG. 2. 2D and 2D flowcharts.

3.6. *Including Electrodes*

In each quasi-2D plane (other than $\kappa = 0$ plane) the value on the electrodes is 0. These fixed potentials can be calculated by inducing a charge on the electrodes using Hockney's "capacity matrix" technique, where the matrices for each plane are calculated in a setup subroutine. (See Ref. [1] for further details.)

The capacity matrices and eigenvalues are calculated in a routine prior to running the model. Hence the modified routine should run in approximately the same time as the original.

4. TEST DATA AND EMPIRICAL TIMING OF 2D AND 3D ROUTINES

(All calculations were performed on an INMOS T800 transputer.)

A mesh size (0–128)x, (0–8)y, (0–8)z was used giving a quasi-2D mesh of ((0–128)x, (0–8)y). The modified charge distribution was varied from 1×10^{-4}V to 2×10^{-4}V by steps of 0.125×10^{-4}V, with a uniform distribution in each plane.

The time to process each quasi-2D plane varied between 0.45 and 0.48 s. This compares to a 2D mesh with identical dimensions and a charge distribution of 1×10^{-4}V which took 0.32 s. The additional 150 ms is accounted for by the time taken to move charge data from the 3D array to a 2D array and to move potential data from the 2D to the 3D array. The time to perform an initial and final Fourier transform was 2 s, so that the total time to process all the planes in serial was approximately 6.5 s.

Since the individual planes and the initial and final FFTs can be processed in parallel (see Fig. 2), even with message passing overheads, a reasonable reduction in time should be achievable.

CONCLUSIONS

We have a described a method which decouples the 3D Poisson equation into a set of 2D elliptical PDEs, for a given set of structures with simple boundary conditions and

uniform geometry in one dimension. We used a model problem to indicate how the decoupled "quasi-2D" equations could be resolved in parallel using direct methods. The initial timings performed in serial have indicated an acceptable speed of calculation per "plane."

REFERENCES

1. R. W. Hockney and J. W. Eastwood, *Computer Simulations Using Particles* (Adam Hilger, Bristol, 1988), p. 166.
2. A. Hussain and G. Chamberlain, *J. Solid State Phys.* **17**, 261 (1982).
3. T. Toyabe, H. Musada, Y. Aoki, H. Shukuri, and H. Hagiwari, *IEEE Trans. Electron Devices* **ED-17**, 2038, (1985).
4. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, (Cambridge Univ. Press, Cambridge, 1989), p. 615.
5. D. S. Kershaw, *J. Comput. Phys.* **26**, 43 (1978).
6. Y. K. Cheung, *The Finite Strip Method In Structural Analysis*, (Rainbow-Bridge, Hong Kong, 1976), p. 2.
7. R. W. Hockney, *Methods Comput. Phys.* **9**, 135 (1970).