INTERNATIONAL JOURNAL OF NUMERICAL MODELLING: ELECTRONIC NETWORKS, DEVICES AND FIELDS
Int. J. Numer. Model. 2006; **19**:47–67
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/jnm.598

Accelerating the transient simulation of semiconductor devices using filter-bank transforms

M. Movahhedi[1,‡], A. Abdipour[1,*,†] and M. Dehghan[2]

[1] *Department of Electrical Engineering, AmirKabir University of Technology (Tehran Polytechnic), 424 Hafez Ave., Tehran, Iran*
[2] *Department of Applied Mathematics, AmirKabir University of Technology (Tehran Polytechnic), 424 Hafez Ave., Tehran, Iran*

SUMMARY

Simulation of high frequency semiconductor devices, where non-local and hot carrier transport cannot be ignored, requires solution of Poisson's equation and at least the first three moments of the Boltzmann transport equation (hydrodynamic transport model). These equations form non-linear, coupled and time-dependent partial differential equations. One of the most efficient solvers of such system of equations is decoupled solver. In conventional decoupled methods, the fully implicit, semi-implicit and explicit methods are used to solve the equations. In fully or semi-implicit schemes, the method is unconditionally stable for any Δt or for very large Δt compared to explicit scheme. Thus, these schemes are very suitable and efficient for transient simulations. But, using these techniques leads to a large system of linear equations. Here for the first time, a filter bank-based preconditioning method is used to facilitate the iterative solution of this system. This method provides efficient preconditioners for matrices arising from discretizing of the PDEs, using finite difference techniques. Numerical results show that the condition number and iteration number are significantly reduced. The most important advantage of this preconditioner is its low computational complexity which can be reduced to $O(N)$. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: coupled PDEs; filter-bank transforms; finite difference method; preconditioning; semiconductor device simulation; Poisson's equation; wavelets

1. INTRODUCTION

The equations describing the behaviour of high-frequency semiconductor devices are generally highly non-linear. Except in a number of artificially simplified cases, the solution of these equations must be performed numerically. Many numerical and computational techniques have

*Correspondence to: A. Abdipour, Department of Electrical Engineering, AmirKabir University of Technology (Tehran Polytechnic), 424 Hafez Ave., Tehran, Iran.
†E-mail: abdipour@aut.ac.ir
‡E-mail: movahhedi@aut.ac.ir

Contract/grant sponsor: Iran Telecommunication Research Center (ITRC)

Received 6 April 2005
Accepted 16 August 2005

been developed for this purpose, with the aim of providing accurate results in relatively inexpensive computational terms.

In microwave and high-frequency semiconductor devices, submicrometer dimensions are used. As device sizes continue to decrease, non-local and hot carrier transport becomes dominant and can no longer be ignored. Therefore, it is necessary to solve the Boltzmann Transport Equation (BTE) and Poisson equation self-consistently to simulate these devices. Hydrodynamic transport equations which are obtained by taking the first three moments of the BTE in **k** space, can be used to study non-stationary transport effects in submicrometer devices [1]. The HydroDynamic Transport (HDT) model consists of the Poisson and the three balance equations. These equations form a set non-linear, coupled and time-dependent partial differential equations. The solution methods can be divided into two categories: coupled (Newton) or decoupled (Gummel) solvers [2]. Since the transport equations in a HDT model are coupled, coupled algorithms may be preferred. The major advantage of using a coupled scheme is that there is no limit on the maximum allowable time step, Δt. However, for large Δt, if the initial guess of the solution in general turns out to be very different from the true solution, hence the overall convergence rate slows down [3]. In addition, since the equations are solved simultaneously, CPU memory requirement is three times larger than that for a decoupled solver.

The decoupled method (Gummel algorithm), treats each of the differential equations separately by decoupling the equations and solving the system step by step. Then this sequence is iteratively repeated until self-consistent values for all unknown variables are obtained with the desired accuracy. In conventional decoupled methods, the fully implicit [2, 4–6], semi-implicit [7, 8] and explicit methods [9] are used to solve the equations. In semi-implicit schemes, while solving an equation, the principal variable is solved for using a fully implicit scheme with currently available values for the other variables [7]. In another semi-implicit scheme only some equations, for instance the continuity equation, are solved implicitly [8]. In some applications, the explicit scheme is used to solve the semiconductor equations. In full-wave analysis of active microwave/millimeter-wave transistors a three-dimensional time-domain solution of Maxwell's equations is coupled to the semiconductor equations [9]. Since in this simulation, the time step is determined according to the stability condition for the Maxwell's equations, no gain is made by using fully implicit or semi-implicit finite difference formulation. Although in explicit scheme, variables can be evaluated easily through simple algebraic computations, but the major disadvantage of this scheme is that this method is numerically unstable for Δt greater than the maximum allowed values. For most practical problems, the maximum Δt is very small and it makes the explicit methods to be very inefficient, especially for obtaining steady-state solutions. In fully or semi-implicit schemes, the method is unconditionally stable for any Δt or for very large Δt compared to explicit schemes. The only disadvantage of this scheme is its high computational cost. Even though this technique is computationally more expensive compared to explicit methods for obtaining variables in new time, it becomes more and more economical in problems where steady-state solutions or transient solutions for long times are desired, because of the possibility of using large Δt when employing the implicit schemes.

As it will be mentioned in the next section, using the fully and semi-implicit methods for parabolic or hyperbolic equations leads to a system of linear equations, $[\mathbf{A}]\mathbf{x} = \mathbf{b}$. In many problems of semiconductor device simulation, size of the matrix [**A**] is very large and has a large condition number. Solutions of such large systems, by a direct method is prohibitively

expensive; because, these methods work fine only for well-posed problems (problems in which the corresponding matrix [**A**] has a small condition number). In this case, an iterative technique is usually adopted and effective preconditioner of the matrix [**A**] is required in order to make it better conditioned than the original matrix. Generally, the better-conditioned system leads to an accelerated convergence in the iterative solution [10, 11]. Some well-documented preconditioning methods such as incomplete LU factorization (ILU) and polynomial preconditioning method can be effective. However, they usually require well-above $O(N)$ operations to implement. Recently, an interesting preconditioning method based on the filter bank transforms was proposed [12]. The most important advantage of the new preconditioner is its low computational cost, which can be reduced to $O(N)$ complexity.

In this paper we will use this preconditioner to accelerate the transient simulation of semiconductor devices. This method provides efficient preconditioners for matrices coming from finite difference of PDEs. Hence, this preconditioner can be applied to matrices arising from discretizing when using the fully and semi-implicit finite difference schemes for solving time-dependent hydrodynamic transport equations. Here, as the first step in the performance investigation of this preconditioner in the simulation of semiconductor devices, we will apply it to the discretized Poisson's equation (as an elliptic equation). Solving this equation always leads to a system of linear equations, even though we use explicit schemes for time-dependent equations.

This paper is organized as follows. Section 2 describes the hydrodynamic transport equations for semiconductor devices, implicit and explicit discretization forms of them and Laplacian operator matrices with different boundary conditions. The basic principles to construct a filter bank-based preconditioner and its complexity analysis are discussed in Section 3. Numerical results for a general problem with Dirichlet boundary conditions and a special problem based on a MESFET transistor structure are presented in Section 4. Finally, some conclusions are drawn is Section 5.

2. HYDRODYNAMIC TRANSPORT MODEL

A semiclassical model, based on the BTE, is used as a compromise between the classical model and the quantum transport theory to describe carrier transport in submicrometer devices. Taking the first three moments of BTE in momentum space yields carrier conservation equation, momentum conservation equation and energy conservation equation, shown below for unipolar devices [2]:

- Carrier conservation equation

$$\frac{\partial n}{\partial t} = -\nabla.(n\mathbf{v}) + \left(\frac{\partial n}{\partial t}\right)_c \tag{1}$$

- Momentum conservation equation

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}.\nabla)\mathbf{v} = \frac{q\mathbf{E}}{m\bigstar} - \frac{1}{m\bigstar n}\nabla(nkT) + \left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}\right)_c \tag{2}$$

- Energy conservation equation

$$\frac{\partial(ne)}{\partial t} = \mathbf{J}.\mathbf{E} - \left(\frac{5}{2} + \alpha\right)\nabla.(\mathbf{J}kT) - \frac{n(e-e_0)}{\tau_e(e)} - \nabla.\mathbf{Q} \tag{3}$$

where

$$\alpha = \frac{\frac{1}{2}m^{\bigstar}\upsilon^2}{\frac{3}{2}(kT)}$$

and $\mathbf{Q} = \kappa\nabla T = -\beta k^2 n\mu T\nabla T$ (β is of order 1) when defining carrier mobility as $\mu \equiv q\tau_p/m^{\bigstar}$.

In the carrier conservation equation (1), the collision term is zero for single-valley problems. The collision term in (2) is treated phenomenologically using the momentum relaxation time

$$\left(\frac{\mathrm{d}\upsilon}{\mathrm{d}t}\right)_c = \frac{\upsilon}{\tau_p(e)} \tag{4}$$

Equations (1)–(4) and Poisson's equation for the potential ψ

$$\nabla^2\psi = -\frac{q}{\varepsilon_0\varepsilon_r}(N_D^+ - n) \tag{5}$$

together with the electric field

$$\mathbf{E} = -\nabla\psi \tag{6}$$

form a system of partial differential equations that allows us to take non-stationary electron transport into account [13].

2.1. Implicit and explicit schemes

The set of transport equations (1)–(3), and the Poisson's equation can be solved using either explicit or implicit methods. Consider a time-dependent equation of the following form:

$$\frac{\partial x}{\partial t} = f(x) \tag{7}$$

The time derivative of x can be obtained from Taylor expansion. This gives

$$x^{t_1+1} = x^{t_1} + \Delta t.\left(\frac{\partial x}{\partial t}\right)_{t_1} + O\{(\Delta t^2)\} \tag{8}$$

where $O\{(\Delta t^2)\}$ represents second and higher order terms in Δt. Ignoring second and higher terms yields

$$\left(\frac{\partial x}{\partial t}\right)_{t=t_1} \cong \frac{x^{t_1+1} - x^{t_1}}{\Delta t} \tag{9}$$

Using (9) in (7) results in

$$x^{t_1+1} = \Delta t.f(x^{t_1}) + x^{t_1} \tag{10}$$

Since all equations on the right-hand side in (10) are known, x^{t_1+1} can be evaluated easily through simple algebraic computation, even if 'f' contains spatial operators. This makes the above method, known as the explicit method, to be very attractive. The major disadvantage of

this scheme is that it is numerically unstable for Δt greater than the maximum allowed values. The maximum value of Δt depends on the operator 'f'.

If a Taylor expansion of x is made around x at $t = t_1 + 1$, then we have

$$x^{t_1+1} = x^{t_1} + \Delta t.\left(\frac{\partial x}{\partial t}\right)_{t_1+1} + O(\Delta t^2) \tag{11}$$

Using (11) and (7) we obtain

$$\frac{x^{t_1+1} - x^{t_1}}{\Delta t} \cong f(x^{t_1+1}) \tag{12}$$

Application of the difference approximation to operator 'f', gives

$$(\mathbf{1} - \Delta t\mathbf{A})x^{t_1+1} = x^{t_1} \tag{13}$$

where $\mathbf{A}$ is the difference approximation of operator 'f'. It can be shown that in our case the above scheme, known as implicit method, is stable for very large Δt compared to the explicit method [2].

2.2. Discretized Poisson's equation

The two-dimensional Poisson's equation has the following form:

$$\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = f(x, y) \tag{14}$$

for (x, y) in a region Ω in the (x, y) plane. To make the solution well defined, we also need to specify boundary conditions, i.e. the value of $\psi(x, y)$ on the boundary of Ω. Usually, three types of boundary conditions are used; Dirichlet and Neumann boundary conditions and also mixed of them. Dirichlet boundary conditions specify the value of the function on the boundary, $\psi|_{\text{boundary}} = \text{Constant}$, when Neumann boundary conditions specify the normal derivative of the function on the boundary of Ω, $\frac{\partial\psi}{\partial n}|_{\text{boundary}} = \text{Constant}$.

Here, we focus on the finite difference discretization of Poisson's equation [14]. Equation (14) can be discritized on an equally spaced mesh as

$$\frac{\psi_{i-1,j} - 2\psi_{i,j} + \psi_{i+1,j}}{(\Delta x)^2} + \frac{\psi_{i,j-1} - 2\psi_{i,j} + \psi_{i,j+1}}{(\Delta y)^2} = f(x_i, y_j) \tag{15}$$

where the subscripts (i, j) denote the (i, j)th grid point on the x–y plane.

For a problem with Neumann boundary condition, Equation (15) can be written in a matrix form

$$[\mathbf{\Gamma}]_{(N.M)\times(N.M)}\mathbf{\Phi}_{(N.M)\times 1} = \mathbf{F}_{(N.M)\times 1} \tag{16}$$

where the vectors $\mathbf{\Phi}$ and $\mathbf{F}$ contain the variables $\psi_{i,j}$ and $f(x_i, y_j)$ which are lined up in a systematic order, also N and M are the number of nodes in x and y directions, respectively. The

matrix $[\Gamma]$ has the form

$$\begin{bmatrix} [\mathbf{A}] & [\mathbf{C}] & & & \mathbf{0} \\ [\mathbf{B}] & [\mathbf{A}] & [\mathbf{B}] & & \\ & \ddots & \ddots & \ddots & \\ & & [\mathbf{B}] & [\mathbf{A}] & [\mathbf{B}] \\ \mathbf{0} & & & [\mathbf{C}] & [\mathbf{A}] \end{bmatrix}_{(N.M)\times(N.M)} \tag{17}$$

where the matrix elements of this block matrix are equal to

$$[\mathbf{A}] = \begin{bmatrix} \zeta & -\frac{2}{(\Delta x)^2} & & & \mathbf{0} \\ -\frac{1}{(\Delta x)^2} & \zeta & -\frac{1}{(\Delta x)^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{(\Delta x)^2} & \zeta & -\frac{1}{(\Delta x)^2} \\ \mathbf{0} & & & -\frac{2}{(\Delta x)^2} & \zeta \end{bmatrix}_{N\times N}$$

$$[\mathbf{B}] = \begin{bmatrix} -\frac{1}{(\Delta y)^2} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & -\frac{1}{(\Delta y)^2} \end{bmatrix}_{N\times N}$$

and

$$[\mathbf{C}] = \begin{bmatrix} -\frac{2}{(\Delta y)^2} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & -\frac{2}{(\Delta y)^2} \end{bmatrix}_{N\times N}$$

Also, the potential vector is

$$\mathbf{\Phi}_{(N.M)\times 1} = [\psi_{1,1}, \psi_{2,1}, \ldots, \psi_{N,1}, \psi_{1,2}, \psi_{2,2}, \ldots, \psi_{N-1,M}, \psi_{N,M}]^{\mathrm{T}} \tag{18}$$

and $\zeta = 2.((1/\Delta x)^2 + (1/\Delta y)^2)$.

As it was mentioned, in this case, only Neumann boundary conditions are used. If only Dirichlet boundary conditions satisfy the problem, the matrix [**Γ**] will be changed to

$$\begin{bmatrix} [\acute{\mathbf{A}}] & [\mathbf{B}] & & & \mathbf{0} \\ [\mathbf{B}] & [\acute{\mathbf{A}}] & [\mathbf{B}] & & \\ & \ddots & \ddots & \ddots & \\ & & [\mathbf{B}] & [\acute{\mathbf{A}}] & [\mathbf{B}] \\ \mathbf{0} & & & [\mathbf{B}] & [\acute{\mathbf{A}}] \end{bmatrix} \tag{19}$$

and

$$[\acute{\mathbf{A}}] = \begin{bmatrix} \zeta & -\frac{1}{(\Delta x)^2} & & & \mathbf{0} \\ -\frac{1}{(\Delta x)^2} & \zeta & -\frac{1}{(\Delta x)^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{(\Delta x)^2} & \zeta & -\frac{1}{(\Delta x)^2} \\ \mathbf{0} & & & -\frac{1}{(\Delta x)^2} & \zeta \end{bmatrix}$$

The matrix (17) is strongly ill-conditioned and one cannot easily apply the preconditioning methods to it. But almost in all semiconductor device simulation problems, both Neumann and Dirichlet Boundary conditions exist. Thus, the condition number of the matrix of the resulted system becomes normal. In such problems, Dirichlet boundary conditions can be added easily by deleting the rows and columns of matrix (17), corresponding to the grid points where the potential is enforced.

Matrices of the one-dimensional Poisson's equation can be easily obtained by placing $M = 1$ in above relations.

3. FILTER BANK-BASED PRECONDITIONER

When solving elliptic PDEs, or when using implicit methods for solving time-dependent PDEs, large systems of linear equations need to be solved. The size of the problems are often too large for using a direct solver, and one has to relay on iterative methods. Such methods are dependent on the condition numbers of the operators in the sense that small condition numbers guarantee a fast convergence to the solution, whereas large condition numbers often result in slow convergence [12].

By the evolution of the wavelet techniques [15, 16], an alternative approach has become available. Results showing that general operators have a sparse representation in wavelet bases were derived in Reference [17], and in Reference [18] it was shown that the efficient scale

decomposition of the wavelet transform can be used to construct diagonal preconditioners when using a Galerkin method. In Reference [19], preconditioners for matrices arising from a finite difference discretization of a partial differential equation were constructed, using Daubechies wavelets. The discrete analogue of the biorthogonal wavelet transforms relies on the so-called perfect reconstruction filter bank transforms. In this paper we use biorthogonal filter banks to carry on the construction of preconditioner for Poisson's equation, which is shown to improve the performance of the method. The motivation for using orthogonal wavelet transforms is that the condition numbers for the transformed and untransformed operator are the same. However, the condition number actually decreases when employing the biorthogonal wavelet transforms.

3.1. Filter bank transforms

The filter bank transform formally does the same thing as the wavelet transform, but it works on discrete functions (grid functions) instead of on wavelet spaces [20]. A perfect reconstruction filter bank is a quadruple of filters, $(h, g, \tilde{h}, \tilde{g})$ satisfying [21]

$$\sum_k h_k = \sum_k \tilde{h}_k = \sqrt{2} \tag{20}$$

$$\sum_n h_n \tilde{h}_{n+2k} = \delta_k \tag{21}$$

$$g_k = (-1)^k \tilde{h}_{1-k}, \quad \tilde{g}_k = (-1)^k h_{1-k} \tag{22}$$

If we define the operators

$$[\tilde{\mathbf{H}}] = \frac{1}{\sqrt{2}} \downarrow \check{\tilde{h}}, \quad [\tilde{\mathbf{G}}] = \sqrt{2} \downarrow \check{\tilde{g}}, \quad [\mathbf{H}] = \sqrt{2} \downarrow \check{h}, \quad [\mathbf{G}] = \frac{1}{\sqrt{2}} \downarrow \check{g} \tag{23}$$

we get the perfect reconstruction identity (see Reference [21] for notation)

$$([\tilde{\mathbf{G}}] \oplus [\tilde{\mathbf{H}}])^{-1} = [\tilde{\mathbf{G}}]^{\mathrm{T}} \oplus [\tilde{\mathbf{H}}]^{\mathrm{T}} \tag{24}$$

and by repeated iteration with $[\tilde{\mathbf{H}}]$, the filter bank transform, $[\tilde{\mathbf{T}}]^{n-n_0}$, of a grid function $\mathbf{s}^n$ is defined as

$$[\tilde{\mathbf{T}}]^{n-n_0}\mathbf{s}^n = \mathbf{d}^{n-1} \oplus \mathbf{d}^{n-2} \oplus \cdots \oplus \mathbf{d}^0 \oplus \mathbf{s}^0 = \mathbf{t}^{n-1} \oplus \mathbf{t}^{n-2} \oplus \cdots \oplus \mathbf{t}^{n_0-1} = \boldsymbol{\tau}$$

$$\mathbf{d}^{n-j} = [\tilde{\mathbf{G}}][\tilde{\mathbf{H}}]^{j-1}\mathbf{s}^n, \quad \mathbf{s}^{n_0} = [\tilde{\mathbf{H}}]^{n-n_0}\mathbf{s}^n$$

i.e.

$$[\tilde{\mathbf{T}}]^{n-n_0} = \left(\bigoplus_{j=1}^{n-n_0} [\tilde{\mathbf{G}}][\tilde{\mathbf{H}}]^{j-1} \right) \oplus [\tilde{\mathbf{H}}]^{n-n_0} \tag{25}$$

With a perfect reconstruction filter bank we get the inversion formula

$$([\tilde{\mathbf{T}}]^{n-n_0})^{-1} = ([\mathbf{T}]^{\mathrm{T}})^{n-n_0} = \left(\bigoplus_{j=1}^{n-n_0} ([\mathbf{H}]^{\mathrm{T}})^{j-1}[\mathbf{G}]^{\mathrm{T}} \right) \oplus ([\mathbf{H}]^{\mathrm{T}})^{n-n_0} \tag{26}$$

We will leave out the number of steps in the transform, and write $[\mathbf{T}], [\tilde{\mathbf{T}}]$ instead of $[\mathbf{T}]^{n-n_0}$, $[\tilde{\mathbf{T}}]^{n-n_0}$. A number of filter banks (and their corresponding wavelets) have been used in different applications, e.g. the classical Haar filter bank (the H filter bank) and its generalizations by Daubechies to higher order filter banks [22] (the D filter banks). If we do not restrict ourselves to orthogonal filter banks, other important examples are the δ-filter banks introduced in Reference [23], and the spline (β) filter banks [21]. Some examples of perfect reconstruction filter banks are shown in Table I. This table contains Daubechies filters (D2 or Haar and D4) that lead to orthonormal wavelets (so $h = \tilde{h}$). Also, the table contains the filter coefficients of δ_1- and δ_3-filter banks.

The mentioned filter bank transforms are one-dimensional and are used for one-dimensional problems. For higher-dimensional problems, we can use the higher-dimensional filter bank transforms that are explained in detail in Reference [12]. In these problems, there are different approaches to how to construct the preconditioner, depending on the chosen filter bank decomposition of operator. The simplest approach is to view the operator as a matrix that acts on vectors, i.e. we map the higher-dimensional domain to one-dimensional and perform the one-dimensional transform on this vector [20]. Another approach is using the higher-dimensional or tensor product filter bank transforms. In the next section, we will compare the performance of these two approaches (called 1-D and 2-D transform), for preconditioning the 2-D Poisson equation.

3.2. Construction of preconditioner

If we consider a partial differential equation with special boundary conditions and assume that this problem is discretized with a finite difference method, a system of linear equations can be obtained as

$$[\mathbf{A}]\mathbf{x} = \mathbf{b} \tag{27}$$

We want to use the filter bank transform to precondition the operator. The idea is that the [**W**] parts of the transformed operator will be diagonally dominant, whereas the [**V**] is not, but this part is very small and can be directly inverted [20].

The filter bank transform is chosen as

$$[\tilde{\mathbf{A}}] = [\mathbf{T}][\mathbf{A}][\tilde{\mathbf{T}}]^{-1} \tag{28}$$

where $[\mathbf{T}]$, $[\tilde{\mathbf{T}}]$ perform $n - n_0$ steps in the filter bank transform and $[\tilde{\mathbf{T}}]^{-1} = [\mathbf{T}]^{\mathrm{T}}$. The matrix representation of the transformed operator will be the same as shown in Figure 1.

Table I. Coefficients of the δ-filter banks and the Daubechies (D) orthogonal filter banks (Daubechies wavelets).

Name	h	$\tilde{h}$
δ_1	$\frac{\sqrt{2}}{4}(1, \underline{2}, 1)$	$(\underline{\sqrt{2}})$
δ_3	$\frac{\sqrt{2}}{32}(-1, 0, 9, \underline{16}, 9, 0, -1)$	$(\underline{\sqrt{2}})$
D_2(Haar)	$(\underline{0.70710678118655}, 0.70710678118655)$	$\tilde{h} = h$
D_4	$(\underline{0.48296291314}, 0.836516303, 0.2241438680, -0.129409522)$	$\tilde{h} = h$

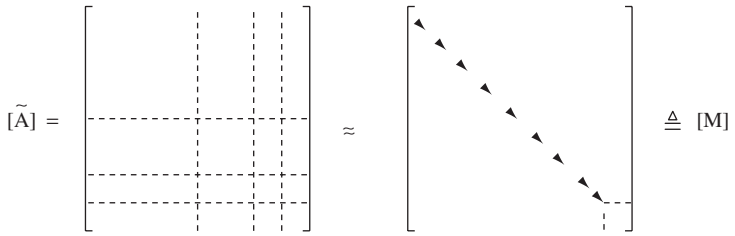


Figure 1. Structure of matrix representation of transformed operator.

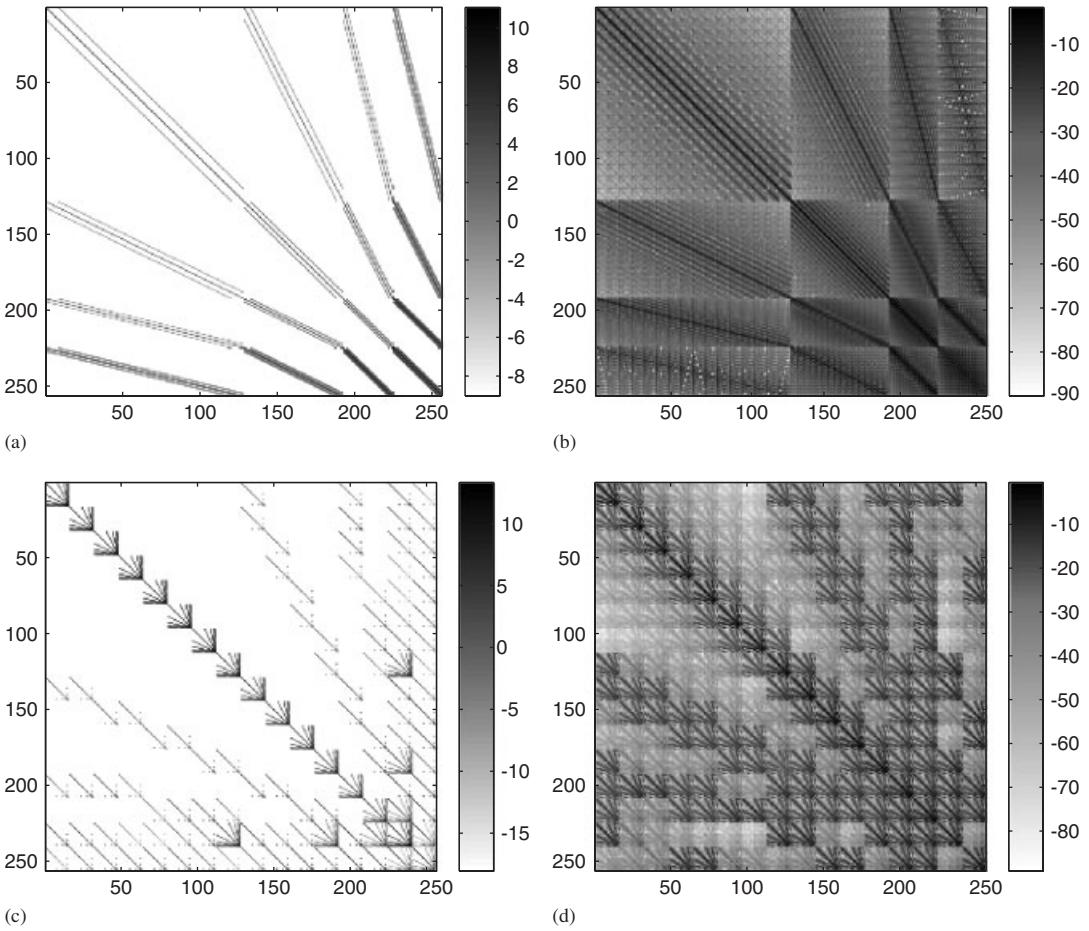


Figure 2. (a) The 1-D filter bank transform of [**A**]; (b) the inverse of the 1-D filter bank transform of [**A**]; (c) the 2-D filter bank transform of [**A**]; and (d) the inverse of the 2-D filter bank transform of [**A**], all drawn in decibel scale.

For example, we consider matrix [**A**] which is derived from discretizing the 2-D Laplacian operator on a 16×16 grid. Figure 2 illustrates the 1-D and 2-D filter bank transforms of the

matrix [**A**] and their inverses. As it is seen in this figure, both the initial matrix and its inverse are sparse with the same sparsity pattern. It makes the calculation of approximation inverse easier and more efficient computationally [11]. Here, [**M**] is the approximation of the transformed operator and we will construct the preconditioner from its inverse, that can be computed easily. Our approximation of $[\tilde{\mathbf{A}}]$ is chosen like in Figure 1 [20]

$$\mathbf{M}_{i,j} = \begin{cases} \tilde{\mathbf{A}}_{i,j}, & i = j, j \in W^k, n_0 \leqslant k < n \\ \tilde{\mathbf{A}}_{i,j}, & i, j \in V^{n_0} \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

Therefore, we can precondition $[\tilde{\mathbf{A}}]$ with $[\mathbf{M}]^{-1}$ as

$$[\tilde{\mathbf{A}}] \rightarrow [\tilde{\mathbf{P}}] \triangleq [\mathbf{M}]^{-1}[\tilde{\mathbf{A}}] \tag{30}$$

Now, the filter bank preconditioning algorithm can be described in following steps:

1. Take the filter bank transform to get $[\tilde{\mathbf{A}}] = [\mathbf{T}][\mathbf{A}][\mathbf{T}]^{\mathrm{T}}$.
2. Create the approximation of the transformed operator ([**M**]) by Equation (29).
3. Use $[\mathbf{M}]^{-1}$ as preconditioner to solve $[\tilde{\mathbf{A}}]\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, where $\tilde{\mathbf{b}} = [\mathbf{T}]\mathbf{b}$.
4. Apply backward filter bank transform to $\tilde{\mathbf{x}}$ to obtain $\mathbf{x} = [\mathbf{T}]^{\mathrm{T}}\tilde{\mathbf{x}}$.

3.3. Complexity analysis of the filter bank preconditioner

We now consider the computational cost of constructing the wavelet and filter bank preconditioner and for the preconditioning operation. This computational complexity which is shown in Table II, is a function of problem dimension, n, problem size, N (m points in each direction, $N = m^n$) and coarsest level (m_0 in each direction, $N_0 = m_0^n$).

For preconditioning, we altogether perform the transformation

$$[\mathbf{A}]\mathbf{x} = \mathbf{b} \rightarrow ([\mathbf{M}]^{-1}[\mathbf{T}][\mathbf{A}][\tilde{\mathbf{T}}]^{-1})([\tilde{\mathbf{T}}]\mathbf{x}) = ([\mathbf{M}]^{-1}[\mathbf{T}]\mathbf{b}) \tag{31}$$

The basic procedures to construct and implement the preconditioner are listed as follows:

1. Transform the matrix [**A**] into wavelet or filter bank domain to generate $[\tilde{\mathbf{A}}]$ and then construct [**M**]. We know that the matrix [**A**], which obtains from discretization of the operator with finite difference method, is sparse [17]. The total number of non-zero elements is $O(m^n)$. For problems with constant coefficients (uniform mesh), one can use that the

Table II. Arithmetic complexity of iterative solver and proposed filter bank preconditioning as a function of number of dimensions, n, problem size, N (m points in each dimension, $N = m^n$), and coarsest level (m_0 points in each dimension, $N_0 = m_0^n$).

Step	Constant coefficients	Variable coefficients
Construction of [**M**]	$O(\log(m))$	$O(m^n)$
Construction of $[\mathbf{M}]^{-1}$	$O(\log(m) + m_0^{3n})$	$O(m^n + m_0^{3n})$
Mult. $[\mathbf{M}]^{-1}[\mathbf{T}]$	$O(m^n + m_0^{2n})$	$O(m^n + m_0^{2n})$
Mult. $[\mathbf{M}]^{-1}[\mathbf{T}][\mathbf{A}][\tilde{\mathbf{T}}]^{-1}$	$O(m^n + m_0^{2n})$	$O(m^n + m_0^{2n})$
Total	$O(m^n + m_0^{3n})$	$O(m^n + m_0^{3n})$

interior of the preconditioner can be constructed on each level, by simply performing the transform on one row of the operator for each level, giving a construction of the operator with $O(\log(m))$ operations. For problems with variable coefficients (non-uniform mesh), the total complexity and memory requirement for the transform are both $O(m^n)$ [12].

2. Compute $[\mathbf{M}]^{-1}$ from $[\mathbf{M}]$. Because of special structure of matrix $[\mathbf{M}]$, the total complexity to compute its inverse is $O(m^n + m_0^{3n})$. We know that the size of non-diagonal part of the matrix is $N_0 = m_0^n$ [12].
3. Carry out the preconditioning operations on the both side of (31): During each iteration, the operation is performed from right to left both side of (31). Because of special structure of matrix $[\mathbf{M}]^{-1}$, its product with another matrix requires $O(m^n + m_0^{2n})$ operations.

To summarize, in every step of constructing the proposed preconditioner and performing all computations for one iteration, the computational complexity and memory requirements are limited to be at most $O(m^n + m_0^{3n})$. The estimates are the same for different wavelet and filter bank transforms. But, the constants are essentially proportional to the length of transform filters. It is interesting to note that if $m_0^{3n} \leqslant m^n$, the computational complexity can be reduced to $O(N)$. This situation will appear in large problems when we transform matrix to very low coarsest levels. Therefore, when the third power of the size of non-diagonal part of $[\mathbf{M}]$ is smaller than the size of $[\mathbf{A}]$, the computational cost for preconditioning will be $O(N)$.

4. NUMERICAL RESULTS

In this section, the effect of different wavelet and filter bank transforms on the condition number of the preconditioned matrix is investigated. In the first subsection, the initial matrix has been obtained from the discretized 1-D and 2-D Poisson's equation with Dirichlet boundary conditions. As mentioned in Section 2, the condition number of the Poisson equation with only Neumann boundary conditions is very huge and cannot use the preconditioning algorithm, directly. Fortunately, in practical problems, both the Neumann and Dirichlet boundary conditions satisfy the problem and it makes the condition number to be normal. Because in different practical problems, the boundary conditions and the combination format of them will be distinctive. Therefore we cannot consider a unique matrix and investigate the effects of preconditioning algorithms on it. So, in Section 4.2, a special problem (MESFET transistor) has been considered. Then, according to its forced boundary conditions, the Laplacian matrix can be obtained and the performance of the proposed preconditioning scheme can be investigated.

It is important to note that, in this paper as the first step in the performance investigation of the proposed method, the preconditioner is applied only to Poisson's equation. But, the method generally can accelerate the iterative solution of the system $[\mathbf{A}]\mathbf{x} = \mathbf{b}$, which arises from discretizing of the PDEs using finite difference techniques. So, the method can be applied, when the other semiconductor conservation equations are discretized using fully or semi-implicit schemes as explained in Sections 2, for accelerating the transient simulation of semiconductor devices. For example, in carrier conservation equation (1), when using implicit scheme we will have a system of linear equations, $[\mathbf{A}]\mathbf{x} = \mathbf{b}$, which $\mathbf{x}$ is the values of electron concentration, n, in the new time. In Reference [24], the performance of the proposed method, when applied to another equation called modified Poisson's equation, has been investigated.

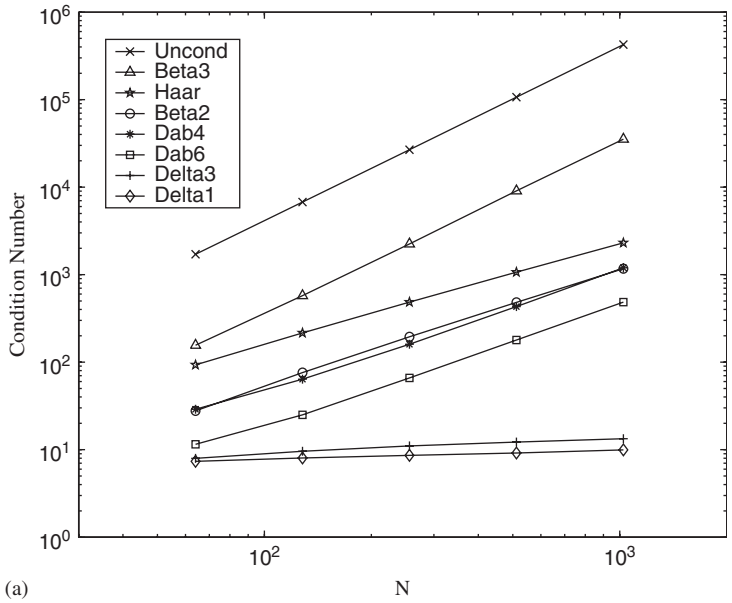
4.1. Dirichlet boundary conditions

We start with the 1-D Poisson's equation with Dirichlet boundary conditions. The 1-D Laplacian operator matrix arising from the finite difference algorithms. We transform it to two coarsest levels $n_0 = 3, 4$, which means that the non-diagonal part of [**M**], [$\acute{\mathbf{M}}$], will be (8×8) and (16×16) matrices. We look at the spline (β) and δ filter banks and Daubechies and Haar wavelets. The condition number of the preconditioned operator for size $N = 32$–1024 is plotted compared with the size of the problem in Figure 3, where we have also plotted the condition number of the unpreconditioned matrix (which grows like N^2). We see that the condition number is improved for all the methods, and that the δ filter banks are the ones that perform best. Especially, the δ_1 filter bank, which only has 2 vanishing moments and its filter length is very short (and requires about the same work as the Haar wavelet) is the one that performs best, with condition number very close to unity. It is interesting to note that by decreasing the coarsest level which leads to shorter non-diagonal part of [**M**], the performance of the preconditioning algorithm using δ filter banks will be improved and the computational cost will be decreased. We have also plotted the condition numbers of the transformed, but unpreconditioned operator, [$\tilde{\mathbf{A}}$], in Figure 4. As it is illustrated in this figure, the condition number of the transformed matrix by filter bank transforms is decreased. Whereas the orthogonal wavelet transforms do not change the condition number of the transformed matrix. Unfortunately, the cancellation properties of the filter bank transforms are not as good in two-dimensional as in one-dimensional. This comes from the fact that the non-diagonal blocks (the [**W**] $\times$ [**V**] blocks) of the discrete Laplacian operator are close to zero in one-dimension. Whereas in two-dimensions, there are many such mixed blocks, and the cancellation for some of them is not as good as in the one-dimensional case. However, the transform gives an operator quite close to diagonal even in higher dimensions.

Here we use (19) as the initial matrix and $\delta 1$ filter bank transform for performance investigation of the proposed preconditioner. As described in Section 3.1, in two dimensions we can construct the preconditioner with mapping either the one-dimensional (1-D transform), or the two-dimensional tensor product (2-D transform). The resulting condition numbers for the preconditioned Laplacian matrix with Dirichlet boundary conditions are shown in Table III. The size of the preconditioned matrix is specified by N and M, which are the number of nodes in x and y directions. [$\acute{\mathbf{M}}$] in this table is the non-diagonal part of [**M**] that its size has been determined. We see that the behaviour of 1-D transform is not stable and in most cases actually increases the condition numbers. But using the tensor product (2-D) transform for preconditioning performs very well. It is clear that in two-dimensional problems not only using 1-D transforms for preconditioning does not improve the condition number but also increases it in some cases.

4.2. Dirichlet and Neumann boundary conditions

To demonstrate the potential of the proposed approach in problems with both Dirichlet and Neumann boundary conditions, it is applied to an idealized MESFET structure (Figure 5) which is discretized by a uniform mesh of $65\Delta x \times 32\Delta y$. The structure parameters are given in Table IV. Dirichlet boundary conditions are used at the electrodes while Neumann boundary conditions are used at the other walls. The Laplacian operator matrix corresponding to this structure can be obtained by considering all boundary conditions as Neumann boundary condition, matrix (17), and then adding Dirichlet boundary conditions by deleting the rows and



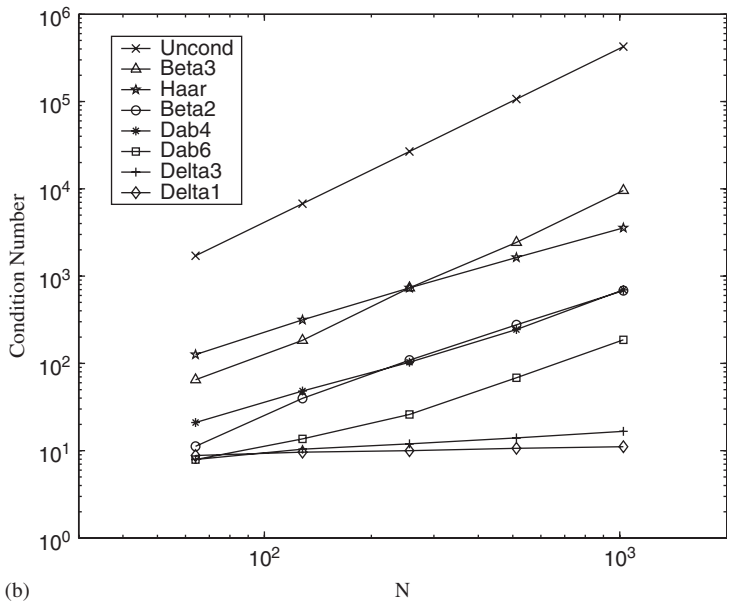


Figure 3. Condition number of preconditioned 1-D Laplacian operator matrix with Dirichlet b.c.s. for different filter banks (Haar, spline β, Daubechies, and δ) versus matrix dimension (N): (a) the coarsest level is $n_0 = 3$; and (b) the coarsest level is $n_0 = 4$.

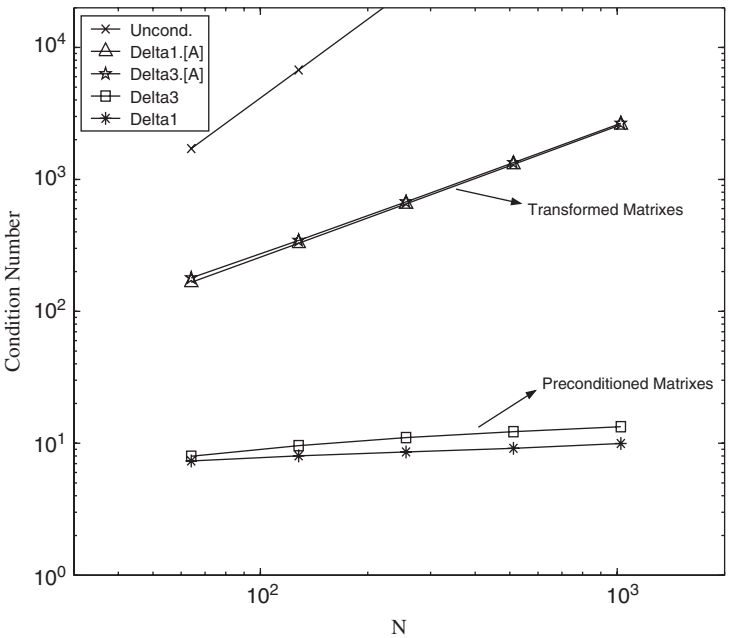


Figure 4. Condition number of preconditioned and transformed 1-D Laplacian operator matrix with Dirichlet b.c.s. for δ_1 and δ_3 filter banks.

Table III. Condition number of preconditioned two-dimensional Laplacian operator matrix with Dirichlet b.c.s. and $(N.M) \times (N.M)$ points, for 1-D and 2-D δ_1 filter bank transforms.

N, M	$[\acute{\mathbf{M}}]$	no precond.	1-D transform	2-D transform
$2^4, 2^4$	$(2^3.2^3) \times (2^3.2^3)$	116	40	22
$2^4, 2^4$	$(2^2.2^2) \times (2^2.2^2)$	116	190	37
$2^5, 2^4$	$(2^3.2^2) \times (2^3.2^2)$	184	283	36
$2^4, 2^5$	$(2^2.2^3) \times (2^2.2^3)$	184	237	36
$2^5, 2^5$	$(2^3.2^3) \times (2^3.2^3)$	440	327	40
$2^6, 2^4$	$(2^3.2^2) \times (2^3.2^2)$	218	245	76
$2^4, 2^6$	$(2^2.2^3) \times (2^2.2^3)$	218	1092	78
$2^6, 2^5$	$(2^4.2^3) \times (2^4.2^3)$	701	349	59
$2^5, 2^6$	$(2^3.2^4) \times (2^3.2^4)$	701	349	58
$2^6, 2^5$	$(2^3.2^3) \times (2^3.2^3)$	701	1527	78
$2^5, 2^6$	$(2^3.2^3) \times (2^3.2^3)$	701	1002	76

columns corresponding to the grid points where the potential is enforced, as explained in Section 2. The size of the matrix will be equal to 2048×2048. The condition number of this matrix is 4051 and we apply the proposed filter bank based and wavelet based preconditioners to it to reduce this condition number.

Table IV. Device parameters used in the simulation.

Drain and source contacts, L_d, L_s	0.2 μm
Gate-source separation, L_{gs}	0.3 μm
Gate-drain separation, L_{gd}	0.4 μm
Device thickness, L	0.3 μm
Gate length, L_g	0.3 μm
Active layer thickness, L_a	0.12 μm
Active layer doping, N_{D1}	2×10^{17} cm^{-3}
Semi-insulating layer doping, N_{D2}	1×10^{13} cm^{-3}
Schottky barrier height	0.7 V
Gate–source voltage	−0.5 V
Drain–source voltage	2 V

Table V. Condition number of the preconditioned two-dimensional Laplacian operator matrix with Dirichlet and Neumann b.c.s., for different 1-D and 2-D δ-filter bank transforms.

Type of filter bank	Name	$[\acute{\mathbf{M}}]$	1-D transform	2-D transform
δ_1	Delta1_{22}	$(2^4.2^3) \times (2^4.2^3)$	4780	592
δ_1	Delta1_{32}	$(2^3.2^3) \times (2^3.2^3)$	6982	698
δ_1	Delta1_{33}	$(2^3.2^2) \times (2^3.2^2)$	16 653	820
δ_1	Delta1_{43}	$(2^2.2^2) \times (2^2.2^2)$	15 368	984
δ_3	Delta3_{22}	$(2^4.2^3) \times (2^4.2^3)$	5678	774
δ_3	Delta3_{32}	$(2^3.2^3) \times (2^3.2^3)$	8274	959
δ_3	Delta3_{33}	$(2^3.2^2) \times (2^3.2^2)$	21 505	1159
δ_3	Delta3_{43}	$(2^2.2^2) \times (2^2.2^2)$	21 354	1300

Table VI. Condition number of the preconditioned two-dimensional Laplacian operator matrix with Dirichlet and Neumann b.c.s., for different 1-D and 2-D Daubechies wavelet transforms.

Type of wavelet	Name	$[\acute{\mathbf{M}}]$	1-D transform	2-D transform
D_4	Dab4_{22}	$(2^4.2^3) \times (2^4.2^3)$	4121	1140
D_4	Dab4_{32}	$(2^3.2^3) \times (2^3.2^3)$	5865	844
D_4	Dab4_{33}	$(2^3.2^2) \times (2^3.2^2)$	5000	810
D_4	Dab4_{43}	$(2^2.2^2) \times (2^2.2^2)$	3558	790
D_2(Haar)	Dab2_{22}	$(2^4.2^3) \times (2^4.2^3)$	7386	1940
D_2(Haar)	Dab2_{32}	$(2^3.2^3) \times (2^3.2^3)$	8848	1412
D_2(Harr)	Dab2_{33}	$(2^3.2^2) \times (2^3.2^2)$	8896	1381
D_2(Harr)	Dab2_{43}	$(2^2.2^2) \times (2^2.2^2)$	27 330	1250

In Tables V and VI, we present the condition number variation of the preconditioned matrix according to the type of filter bank and wavelet transforms and to the number of steps in the transform, which determines the size of non-diagonal part of matrix [**M**]. Also, we use, as before, the 1-D and tensor product (2-D) transforms for preconditioning. It is clearly shown

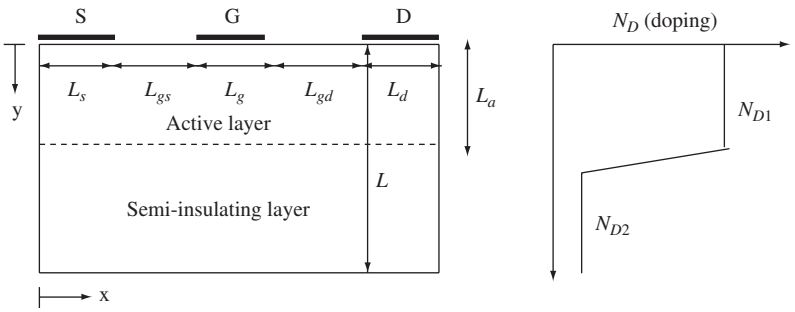


Figure 5. Cross section of the simulated MESFET transistor.

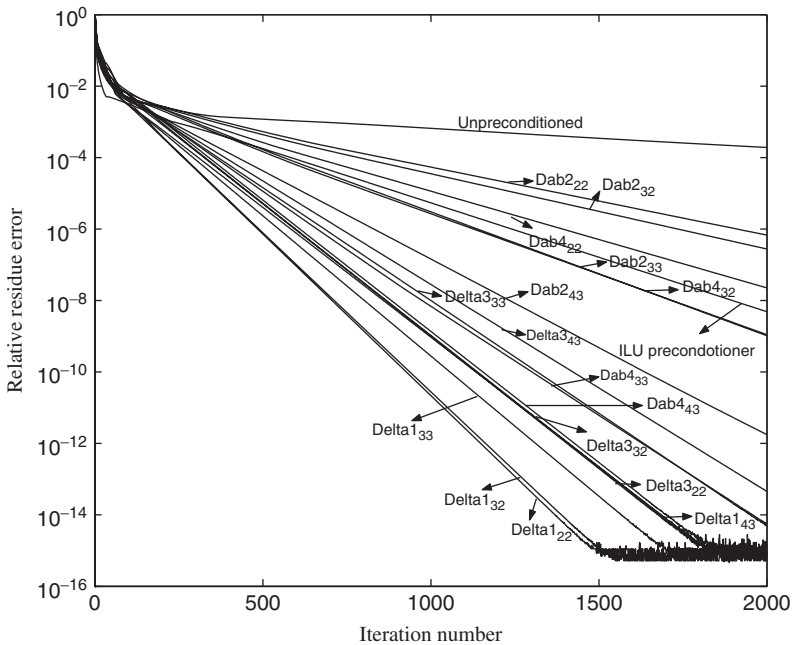


Figure 6. Convergence behaviour of the ILU preconditioner and the proposed preconditioner system by different wavelet and filter-bank transforms.

that for the 2-D δ filter bank transforms, the condition number decreases as the filter bank is of lower order and as the decomposition level decreases (Table V). But for the Daubechies wavelet transforms, the condition number decreases as the wavelet is of higher order and as the decomposition level increases (Table VI). Figure 6 shows the convergence behaviour of the proposed preconditioner for different filter bank and wavelet 2-D transforms. Convergence behaviour of the preconditioned system is similar to variation of its condition number. As it is seen, the convergence rate increases as the filter bank is of lower order and as the decomposition levels decreases. But for the Daubechies wavelet transforms the convergence rate increases as the

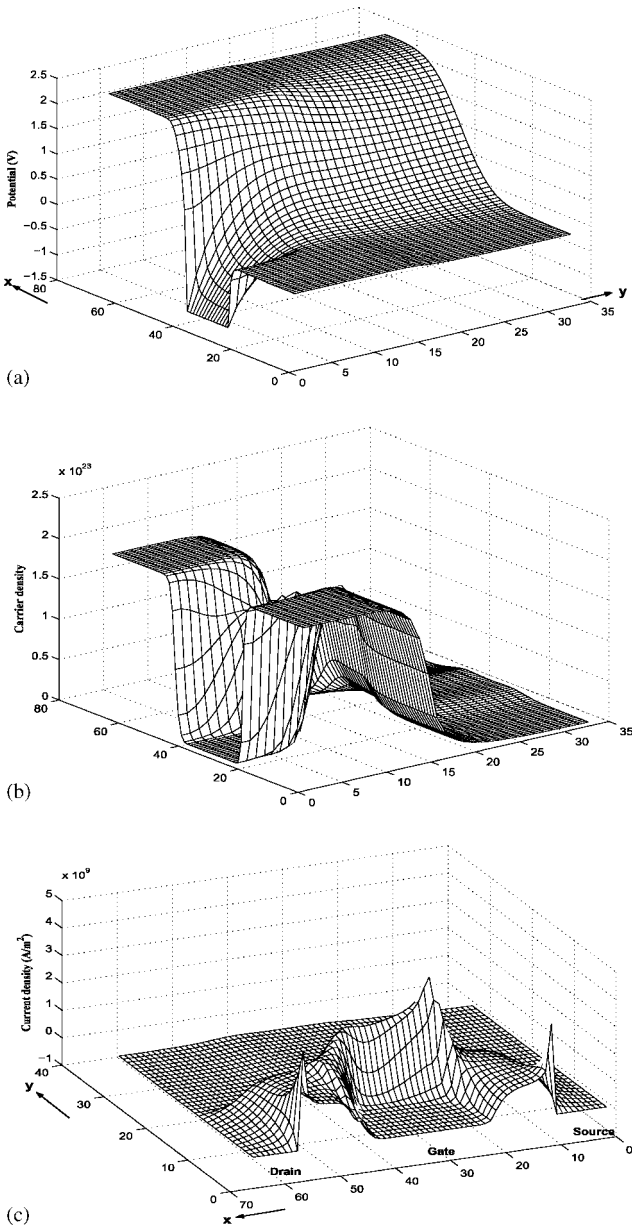


Figure 7. Sample DC results obtained using the proposed algorithm: (a) potential distribution; (b) carrier-density distribution; and (c) current density, J_x.

wavelet is as higher order and as the decomposition level increases. We find that preconditioning using δ_1-filter bank transform converges faster than the other filter bank and wavelet transforms. To compare the performance of the used preconditioner (filter bank-based

preconditioner) with the well-known preconditioning methods, the convergence rate of the incomplete LU factorization (ILU) preconditioner which applied to our problem, has been illustrated in Figure 6. As it is clearly seen, almost in all cases, for different filter-bank and wavelet transforms and decomposition levels, the convergence rate is faster than the ILU(0) preconditioner.

We know when the order of the wavelets and filter banks increases, the length of corresponding filters for transformation increases too (Table I). Also by increasing the number of steps in the transform, the size of non-diagonal part of [**M**] decreases. Therefore, the computational complexity of the preconditioning method, which is equal to $O(N + \acute{M}^3)$—N is the size of original matrix and $\acute{M}$ is the size of non-diagonal part of [**M**]—can be reduced by increasing the number of steps in the transform.

It is interesting that we can obtain both good conditioning and low computational cost by using δ_1-filter bank transform. For example, in $\text{Delta}1_{43}$ situation (Table V), the complexity is $O(2^{11} + (2^4)^3) - N = 2^{11}$ and $\acute{M} = 2^4$—or $O(N)$.

Here, we use the solution of Poisson's equation obtained using the proposed filter bank preconditioner for simulating of the considered MESFET transistor. The transistor model used in this study is a two-dimensional (2-D) simplified hydrodynamic transport model [8]. An explicit finite-difference time-domain (FDTD) scheme was used to discretize equations of the hydrodynamic transport model (Section 2). More details of the discretization method of each term in the conservation equations can be found in Reference [25]. Figure 7(a) shows the potential distribution obtained using the proposed algorithm, while, Figures 7(b) and (c) illustrate the carrier density and the current density distributions. It is significant to indicate that the proposed algorithm gives precisely the same results obtained when the used iterative method does not employ the proposed preconditioner method. The comparison results between the algorithms are not provided because their results coincide exactly on each other.

5. CONCLUSIONS

In this paper, we have proposed a filter bank-based preconditioner to facilitate and accelerate the iterative solution of the system arising from discretizing of fully or semi-implicit schemes for numerically solving the hydrodynamic transport equations. Here, as the first step to use this preconditioner for simulation of semiconductor devices, it is applied to Poisson's equation. The filter bank transforms provide efficient preconditioners for matrices appearing when using finite difference methods. The step of going from orthogonal (wavelet-based) to non-orthogonal (filter bank-based) transforms actually seems to improve the condition number of the unpreconditioned Laplacian operator. Especially, the δ_1 filter bank transform, which its filter length is very short, is well-suited for our problem.

The proposed method by considering the matrix which arising from the discretization of a PDE, can estimate a better approximation of the transformed matrix with lower computational cost to calculate the approximate inverse of the matrix. Also, we use biorthogonal filter bank transforms that have better performance than conventional orthogonal wavelet transforms, as was shown. Numerical results show that the performance of δ filter bank transform for one-dimensional problems is better than wavelet transforms and the condition number of the preconditioned matrix is very close to unity. For two-dimensional problems, we can obtain both

good conditioning and low computational cost by the proposed method, which uses the tensor product or 2-D transform. Results show that the convergence rate of the used preconditioning scheme, by δ_1 filter-bank transform, is faster than the well-known ILU method. Moreover, the total computational cost for the construction and application of the preconditioner can be reduced to $O(N)$ by increasing the number of steps in the transform.

ACKNOWLEDGEMENTS

This research was supported in part by Iran Telecommunication Research Center (ITRC).

REFERENCES

1. Grasser T, Tang TW, Kosina H, Selberherr S. A review of hydrodynamic and energy-transport models for semiconductor device simulation. *Proceedings of the IEEE* 2003; **51**(8):251–274.
2. Yoganathan S, Banerjee SK. A new decoupled algorithm for non-stationary, transient simulations of GaAs MESFET's. *IEEE Transactions on Electron Devices* 1992; **39**(7):1578–1587.
3. Selberherr S. *Analysis and Simulation of Semiconductor Devices*. Springer: New York, 1984.
4. Mock MS. A time-dependent numerical model of the insulated-gate filed-effect transistors. *Solid-State Electronics* 1981; **24**(10):959–966.
5. Nikoleava VA, Ryzhii VI, Chetverushkin BN. Numerical simulation of non-equilibrium processes in an electron-hole plasma of binary heterostructures. *Inzhenerno-Fizicheskii Zhurnal* 1986; **51**(3):230–241.
6. Kreskovsky JP, Grubin HL. Application of LBI techniques to the solution of the transient, multidimensional semiconductor equations. *Journal of Computational Physics* 1987; **68**:420–432.
7. El-Ghazaly S, Lefebvre M, Salmer G, Ibrahim M, El-Sayed OL. Two-dimensional FET simulation in non-stationary conditions. In *Proceedings of the 13th European Solid State Devices Research Conference*, Cantebury, U.K., 1983; 127–130.
8. Feng F, Hintz A. Simulation of submicrometer GaAs MESFET's using full dynamic transport model. *IEEE Transactions on Electron Devices* 1988; **35**(9):1419–1431.
9. Alsunaidi MA, Imtiaz SM, El-Ghazaly SM. Electromagnetic wave effects on microwave transistors using a full wave high-frequency time-domain model. *IEEE Transactions on Microwave Theory and Techniques* 1996; **44**(6):799–808.
10. Saad Y. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Co.: Boston, 1996.
11. Deng H, Ling H. An efficient wavelet preconditioner for iterative solution of three-dimensional electromagnetic integral equation. *IEEE Transactions on Antenna and propagation* 2003; **51**(3):654–660.
12. Walén J. Filter bank preconditioners for finite difference discretizations of PDEs. *Technical Report 198*, Department of Scientific Computing, Uppsala University, Sweden, 1997.
13. Tomizawa K. *Numerical Simulation of Submicron Semiconductor Devices* (1st edn). Artech House: Norwood, 1993.
14. Sadiku MN. *Numerical Techniques in Electromagnetics* (2nd edn). CRC press: Boca Raton, FL, 2000.
15. Chen K. Discrete wavelet transform accelerated sparse preconditioners for dense boundary element system. *Electrical Transactions on Numerical Analysis* 1999; **8**:138–153.
16. Ford JM. Wavelet-based preconditioning of dense linear systems. *Ph.D. Dissertation*, University of Liverpool, 2001.
17. Beylkin G, Coiman R, Rokhlin V. Fast wavelet transform and numerical algorithms I. *Communications on Pure and Applied Mathematics* 1991; **44**:141–148.
18. Jaffard S. Wavelet method for fast resolution of elliptic problems. *SIAM Journal of Numerical Analysis* 1992; **29**:965–986.
19. Dorobantu M. Wavelet-based algorithms for fast PDE solvers. *Ph.D. Dissertation*, Royal Institute of Technology, Sweden, 1995.
20. Walén J. A general adaptive solver for hyperbolic PDEs based on filter bank subdivisions. *Applied Numerical Mathematics* 2000; **33**:317–325.
21. Walén J. Filter bank methods for hyperbolic PDEs. *Technical Report 185*, Department of Scientific Computing, Uppsala University, Sweden, 1996.
22. Daubechies I. Orthonormal bases of compactly supported wavelets. *Pure and Applied Mathematics* 1988; **41**: 909–996.
23. Deslauriers G, Dubue S. Symmetric iterative interpolation processes. *Constructive Approximation* 1989; **5**(1):49–68.
24. Movahhedi M, Abdipour A. Accelerating the transient simulation of semiconductor devices using filter-bank transforms. In *Proceedings of the 13th European Gallium Arsenide and other Compound Semiconductors Application Symposium* (GAAS®2005), Paris, France, 2005; 477–480.
25. Aste A, Vahldiek R. Time domain simulation of the full hydrodynamic model. *International Journal of Numerical Modelling*: *Electronic Networks*, *Devices and Fields* 2003; **16**(2):161–174.

AUTHORS' BIOGRAPHIES

Masoud Movahhedi was born in Yazd, Iran in 1976. He received BSc in Electrical Engineering in 1998 from Sharif University of Technology, and MSc degree from AmirKabir University of Technology in 2000 in Electrical Engineering. He is working toward PhD degree in Electrical Engineering at the same university, since 2001. His research interests are in the areas of computer-aided design of microwave integrated circuits, computational electromagnetics and semiconductor high-frequency RF modelling.

Mr Movahhedi was the recipient of the GAAS-05 fellowship sponsored by the GAAS Association to young graduate researchers for his paper presented at the GAAS®2005.

Abdolali Abdipour was born in Iran in 1966. He obtained his BSc in Electrical Engineering from Tehran University, Tehran, Iran in 1989, and the MSc in Electronics from Limoges University, Limoges, France in 1992. Then he achieved his PhD degree in Electronic Engineering from Paris XI university, Paris, France in March 1996. His research areas include wireless communication systems (RF Technology and Transceivers), RF/Microwave/mm-wave circuit and system design, E&M modelling of active devices and circuits, high frequency electronics (signal and noise), non-linear modelling and analysis of microwave devices and circuits. He has published over 80 papers in the refereed journals and the local and international conferences. He authored *Noise in Electronic Communication: Modeling*, *Analysis and Measurement* (in Persian).

He is currently an associate professor of Electrical Engineering Department at Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran. He is also cooperating with Iran Telecommunication Research Center (ITRC) as a project manager.

Mehdi Dehghan is currently an associate professor in the Department of Applied Mathematics, AmirKabir University of Technology, Tehran, Iran. His research interests include Numerical Solution of Partial Differential (and Integral) Equations, Numerical Integration, Numerical Linear Algebra and Difference Equations. He has published about 40 papers in the refereed journals and he has also presented several papers in the local and international conferences.