PERGAMON

Solid-State Electronics 44 (2000) 1069–1075

SOLID-STATE ELECTRONICS

Levelized incomplete LU method and its application to semiconductor device simulation

Yao-Tsung Tsai [*], Chun-Yi Lee, Mu-Kai Tsai

Department of Electrical Engineering, National Central University, Chung-Li, Taiwan, ROC

Received 22 April 1998; received in revised form 9 November 1999

Abstract

In circuit simulation, the CPU time is spent in two parts: one is to transfer the circuit equation into the corresponding linear equation $Ax = B$, the other is to solve this matrix equation. In order to improve the simulation speed and apply our method to mixed-level device and circuit simulation, we propose to simplify the creation of the matrix equation by equivalent subcircuit, and speed up the simulation by levelized incomplete LU factorization [Karl-Michael E, Walter LE. IEEE Trans Comp Aided Des Integ Circ and Sys 1995;14:720.]. The levelized incomplete LU is used to solve $Ax = B$ because it offers, the good convergence of the direct method and the high speed, small memory space of the iteration method. The $Ax = B$ is obtained by transferring the Poisson equation and continuity equation into their equivalent circuits [Leblebici Y, Unlu MS, Morkoc H, Kang SM. IEEE Int Symp Circ and Sys 1992;2:895; 1995;13:396.] to simplify the mixed-level simulation. Finally, we will apply the above methods to the simulation of PN diodes and verify their performance on the simulation and design of semiconductor devices. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Levelized; Incomplete LU; Semiconductor device; Simulation

1. Introduction

A matrix solver always plays a major role in circuit simulation. In a VLSI circuit simulation, the consumption of CPU time and memory in solving the linear equations readily increases with the circuit scale. From above, developing a matrix solver which is fast and uses fewer system resource is needed for the circuit designer. There are two ways to solve matrices: direct method and iteration method. A direct method is easy to converge, but the solving time will increase rapidly with matrix size. Take the LU factorization method for example, a lot of "fill-in" will be generated when we use this method on a sparse matrix. The term fill-in is associated with direct methods. During the processes of LU factorization, some zero entries might become nonzero due to the operations. The new nonzero terms are called fill-ins. One fill-in tends to cause another. This phenomenon increases both the complexity and cost of computation. On the contrary, the iteration method requires shorter time and lesser memory, but it is not as stable as the direct method, because of diverging or slow converging if the matrix is not diagonally dominant. This limits its usage. In order to fully utilize the advantage of these methods and avoid the disadvantage, we plan to use the levelized incomplete LU method to develop a new matrix solver. Simply speaking, the levelized incomplete LU method is used to purge the high level fill-ins which are produced from LU decomposition. In this way, we can economize on memory space and calculation time, and then, search the right answer by iteration calculation. When alpha (Truncation parameter) is equal to zero, this is called the Gauss–Jacobi iteration method. When alpha is infinite, its a complete LU factorization. Thus, by different choices of alpha, we can dynamically change the weighting of direct and iteration method to gain the advantages of swiftness, robustness and small memory space. If we can minimize memory space and

[*] Corresponding author.

PII: S0038-1101(00)00003-4

shorten the simulation time, many circuit and device simulation can be directly conducted on a PC instead of a workstation.

On the contrary, in order to provide the IC designer an effective development circumstance, we try to integrate the circuit and device simulation into a complete mixed-level circumstance. In the process of VLSI design, we may need to understand the characteristics of each device of electric circuit, but the simulation software such as SPICE, we use now cannot do the detailed numerical simulation of a semiconductor device. The result of a device simulation software, such as PISCES, cannot be used in circuit simulation too. So, we used the equivalent circuit method to overcome the problem. The Poisson and continuity equations which control the behavior of charge carriers inside a semiconductor device are formulated into equivalent circuits, so that circuit elements such as voltage sources, resistors and capacitors can be used to implement the device equations [1,2]. This method not only provides the availability to simulate a semiconductor device numerically by using a circuit simulator, but also reduces the complexity of a numerical simulation. In addition, this method makes the mixed-level simulation between the semiconductor device and external circuit easier.

Finally, the mixed-level simulation of the diode switching circuit was executed to test the levelized incomplete LU matrix solver and equivalent circuit. The results show the method that we used is helpful in reducing simulation time and easier for simulating a mixed-level circuit.

2. Levelized incomplete LU method

In circuit simulation, we often use some nonlinear algebraic and differential equations to describe the behavior of each node. This set of equations is transformed into a sequence of systems of nonlinear algebraic equations $F(x) = 0$. Newton's method is always used in the traditional circuit simulation. For each iteration, k, Newton's method requires the computation of the inhomogeneity F and the Jacobian $F' = (\mathrm{d}/\mathrm{d}X)F(X)$ as well as the solution of the linear system

$$F'(X^{(k)})\delta^{(k)} = -F(X^{(k)}), \quad k = 0, 1, 2\ldots \tag{1}$$

for the correction $\delta^{(k)} = x^{(k+1)} - x^{(k)}$.

We can transfer Eq. (1) to the style of the linear equation

$$As = B, \tag{2}$$

where $A = F'(X^{(k)})$, $B = -F(X^{(k)})$, and $s = \delta^{(k)}$. Assume a sequence of approximate solution s_0, s_1, s_2,..., can be computed with the following relationship:

$$s_{i+1} = s_i + \varepsilon_i, \quad i = 0, 1, 2\ldots \tag{3}$$

We use the Newton-iterative method to derive the solution of $As = B$ quickly by decomposing matrix A into the nonsingular part Q and remainder R.

$$A = Q - R. \tag{4}$$

Therefore, Eq. (2) can be rewritten as

$$(Q - R)s = B, \tag{5}$$

then

$$Qs_{i+1} = B + Rs_i = B + (Q - A)s_i$$

or

$$s_{i+1} = (I - Q^{-1}A)s_i + Q^{-1}B. \tag{6}$$

Hence, the linear iteration scheme (6) converges for any choice of the starting value s_0 if and only if the spectral radius $\rho(I - Q^{-1}A) < 1$. Owing to the large calculating processes of Eq. (6), we can rewrite it as

$$Q(s_i + \varepsilon_i) = B + (Q - A)s_i \tag{7}$$

or

$$Q\varepsilon_i = B - As_i. \tag{8}$$

By continuous iteration of Eqs. (8) and (3), ε_i will approach zero, and s_i will close to the solution $\delta^{(k)}$.

We can find that the selection of matrix Q is very important to the convergent situation for the processes of the Newton-iterative method [3]. So that making the choice of a suitable Q to increase the convergence speed and to reduce the memory usage is one of the most important subject. The method we proposed is using the levelized incomplete LU method to solve the dilemma. The following items are the principles of distinguishing the level of fill-ins which come from the LU factorization of matrix A in the levelized incomplete LU method:

1. In matrix A, the main diagonal entries (i,i) are assigned to be in level 0.
2. In matrix A, off-diagonal nonzero entries (i,k), $i \neq k$, are assigned to be in level 1.
3. After LU factorization, the fill-in at position (i,j) which is generated from two off-diagonal entries (i,k) and (k,j). $L1$ and $L2$ are the levels of two entries, the fill-ins level L is $L1 + L2$.
4. If the fill-in at position (i,j) is generated in more than one way,such as (i,k),(k,j) and (i,m),(m,j). $L1$,$L2$,$L3$,$L4$ are their levels, and $(L1 + L2) < (L3 + L4)$, then the fill-ins level L is the minimum level $(L1 + L2)$.

The user could choose the suitable level (we can call it as truncation parameter $\alpha \in \{0, 1, 2\ldots\}$) by the property of matrix A. When matrix A is diagonally dominant, we can choose a smaller α to help converging. On the contrary, we have to choose the larger one. If the level of fill-in is over α, we delete this fill-in, that is, keeping the fill-ins whose level is under α. After this procedure, we have

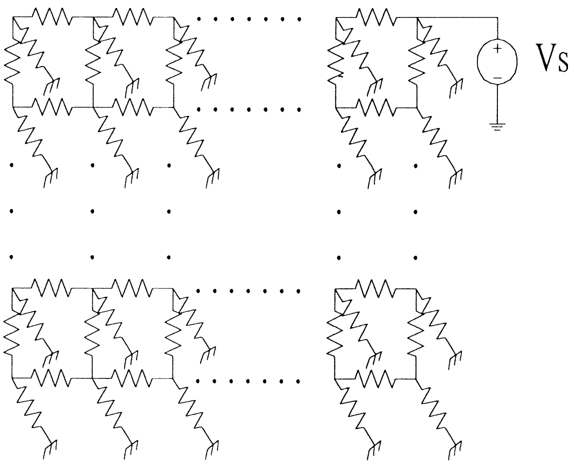

Fig. 1. A resistive grid illustrating the dependence of the matrix solution method on circuit size. The value of every resistor connected to the ground is 100 Ω; the others are 1 Ω; the value of the voltage source is 5 V.

$$A = \mathrm{LU} = Q - R = \tilde{\mathrm{L}}\tilde{\mathrm{U}} - R, \tag{9}$$

where $\tilde{\mathrm{L}}\tilde{\mathrm{U}}$ means an incomplete LU. It is clear here to see why the method is called incomplete LU because of keeping only the fill-ins whose levels are less than α.

When we observe the fill-ins produced by LU factorization, we can find that the higher the level, the lesser the entries affect the fill-ins. That is the reason why, we have the above derived principle. We can minimize the large calculating processes by removing the higher level fill-ins. Then, we can quickly solve Eq. (2) by Eqs. (8) and (9).

By choosing α, we can select the solving methods between the direct and iteration methods as we want. When α is null, we only take the value at the diagonal into iteration. This is the so-called Gauss–Jacobi method. When α is unity, the nonzero entries of matrix A will be maintained. This is the Stone method. When $\alpha = \infty$, all of the fill-ins will be maintained. This is the classic LU method.

To prove the performance of Levelized Incomplete LU, we design the resistance matrix as shown in Fig. 1. In Fig. 2, we can find the most efficient way is obtained by choosing $\alpha = 4$. In Fig. 3, we can see that the levelized incomplete LU takes the least time compared to the other methods although the node number of the resistance matrix increases.

3. Mixed level simulation

The carrier transport in semiconductors is described by three coupled partial differential equations, namely, the Poisson equation, the electron continuity equation and the hole continuity equation as follows:

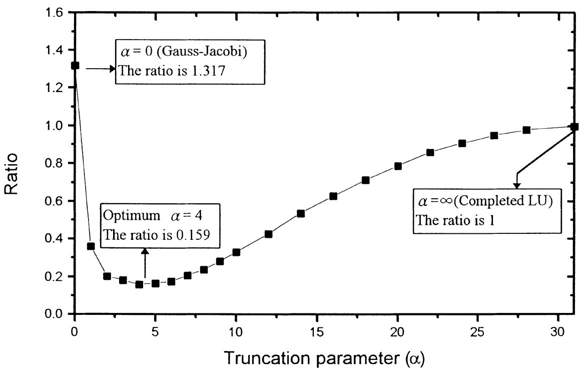

Fig. 2. Dependence of the computation effort of the levelized incompleted LU method on the different truncation parameter, α, for the resistive grid circuit simulation.The number of nodes are 1000. The CPU times are normalized with respect to the time for the complete LU method.

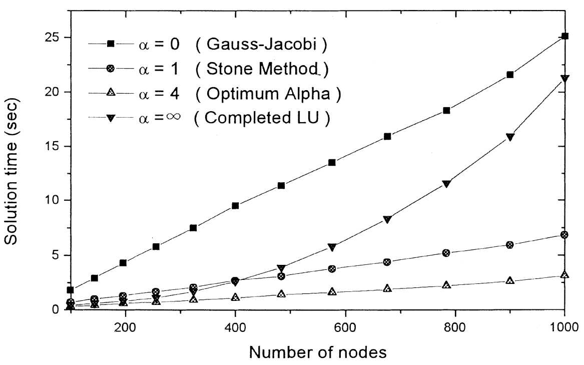

Fig. 3. Dependence of levelized incomplete LU method on the circuit size in comparison to that of the other different methods.

$$\nabla^2 \varphi = -\frac{q}{\varepsilon}\left[p - n + N_{\mathrm{D}}^{+} - N_{\mathrm{A}}^{-}\right], \tag{10}$$

$$\frac{\partial n}{\partial t} = \frac{1}{q}\nabla \vec{J}_n - R, \tag{11}$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q}\nabla \vec{J}_p - R, \tag{12}$$

where ε is the dielectric constant, q is the magnitude of the fundamental charge, N_{D}^{+} is the density of activated (ionized) donors, N_{A}^{-} is the density of activated acceptors, p is the density of holes, and n is the density of electrons. We assume that all of the dopants are activated everywhere, so that N_{D}^{+} and N_{A}^{-} are functions only of position. R is the recombination–generation term.

J_n and J_p are written as functions of (φ,n,p), consisting of drift and diffusion components. Besides, we neglect the effects of band gap narrowing and assume Boltzmann carrier statistics, that is, $\vec{E}_n = \vec{E}_p = \vec{E} = -\nabla\varphi$. For the range of operation of most semiconductor devices, the Fermi–Dirac distributions can be simplified as [4]

$$n \approx N_C \exp(\eta_n) = n_i \exp[(\varphi - \phi_n)/\phi_T], \tag{13}$$

$$p \approx N_V \exp(\eta_p) = n_i \exp[(\phi_p - \varphi)/\phi_T], \tag{14}$$

where the φ, ϕ_n and ϕ_p are the electrostatic potential, electron quasi-Fermi potential, and the hole quasi-Fermi potential, respectively.

For one-dimensional device simulation, the device can be partitioned into $n-1$ elements and n nodes as shown in Fig. 4. The number of unknown variables will be $3n$ by using a circuit simulator. The equivalent circuit model for one dimensional mixed-level simulation is shown in Fig. 4. It can be seen that each grid point is represented by three circuit nodes, where the node potentials correspond to the electrostatic potential and the two quasi-Fermi potentials associated with the grid point.

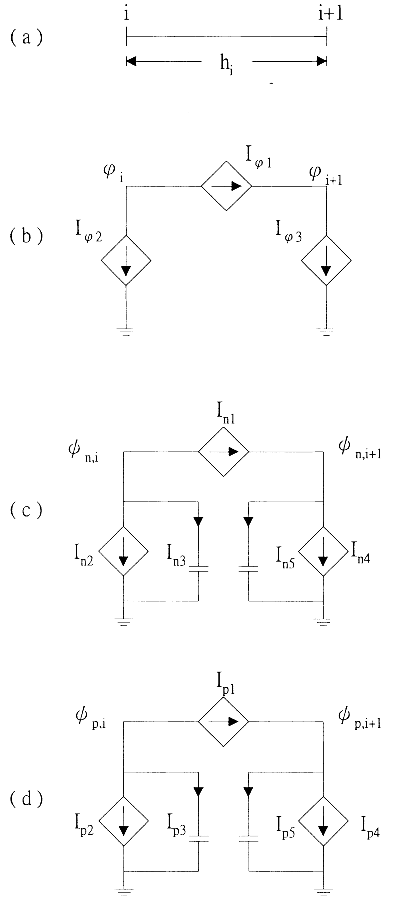

Fig. 4. One-dimensional device's equivalent circuit model: (a) *i*th element in one-dimensional simulation; (b) equivalent circuit for Poisson's equation; (c) circuit for electron continuity equation; (d) circuit for hole continuity equation.

First of all, the box-integration method [5] based on element by element is used to discrete the Poisson equation. For the *i*th element in Fig. 4(a), Poisson's equation can be rewritten as

$$\varepsilon A\left(\frac{\varphi_i - \varphi_{i+1}}{h_i}\right) + q(n - p - N_\mathrm{D}^+ + N_\mathrm{A}^-)_i \frac{h_i}{2} A = 0, \tag{15}$$

where A is the cross area.

Eq. (15) can be simplified as

$$I_{\varphi 1} + I_{\varphi 2} = 0, \tag{16}$$

where

$$I_{\varphi 1} = \frac{\varepsilon A}{h_i}(\varphi_i - \varphi_{i+1}), \tag{17}$$

$$I_{\varphi 2} = q(n - p - N_\mathrm{D}^+ + N_\mathrm{A}^-)_i \frac{h_i}{2} A. \tag{18}$$

Similarly, at node $i+1$,

$$-\varepsilon A\left(\frac{\varphi_i - \varphi_{i+1}}{h_i}\right) + q(n - p - N_\mathrm{D}^+ + N_\mathrm{A}^-)_{i+1} \frac{h_i}{2} A = 0. \tag{19}$$

Eq. (19) can be further simplified as

$$-I_{\varphi 1} + I_{\varphi 3} = 0, \tag{20}$$

where

$$I_{\varphi 3} = q(n - p - N_\mathrm{D}^+ + N_\mathrm{A}^-)_{i+1} \frac{h_i}{2} A. \tag{21}$$

The equivalent circuit is shown in Fig. 4(b).

Next, the equivalent circuit for electron continuity equation can be written as follows:

At node i:

$$I_{n1} + I_{n2} + I_{n3} = 0, \tag{22}$$

where

$$I_{n1} = -qD_n \frac{B(X_b)n_i - B(-X_b)n_{i+1}}{h_i} A, \tag{23}$$

$$I_{n2} = -qR_i \frac{h_i}{2} A, \tag{24}$$

$$\begin{aligned} I_{n3} &= -q\frac{\partial n_i}{\partial t}\frac{h_i}{2}A \\ &= (N_\mathrm{D}^+ - N_\mathrm{A}^-)(\varphi_i - \phi_{n,i})\frac{\partial(\phi_{n,i} - \varphi i)}{\partial t}. \end{aligned} \tag{25}$$

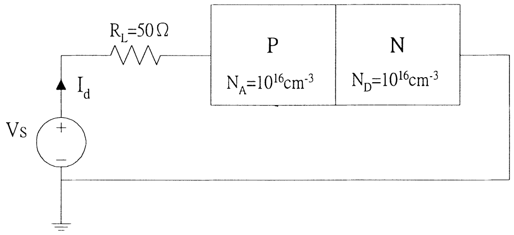

Fig. 5. A diode switching circuit for 1D mixed-level simulation.

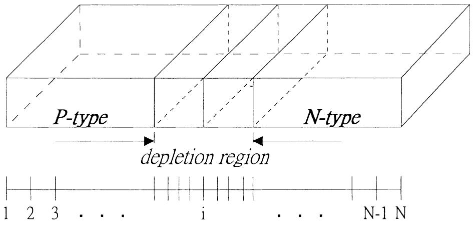

Fig. 6. Partitioning of a PN diode on a one-dimensional non-uniform grid for device simulation.

Note that X_b in I_{n1} is $(\varphi_i - \varphi_{i+1})/\phi_T$. $B(X)$ is the Bernoulli function which is defined as

$$B(X_b) = \frac{X_b}{\exp(X_b) - 1}. \qquad (26)$$

Recombination rate R_i is described by the well-known Shockley–Read–Hall model as

$$R_i = \left[\frac{pn - n_i^2}{\tau_p(n + n_i) + \tau_n(p + n_i)}\right]_i. \qquad (27)$$

In Eq. (27), we assume that the trap center is located at the center of the forbidden gap. I_{n2} responds to a recombination current, and I_{n3} to a nonlinear voltage dependent current, and can be implemented as a nonlinear capacitor. R_i and n_i are the recombination rate and the electron concentration at node i, and they may include the three variables φ, ϕ_n, and ϕ_p.

At node $i + 1$*:*

$$-I_{n1} + I_{n4} + I_{n5} = 0, \qquad (28)$$

where

$$I_{n4} = -qR_{i+1}\frac{h_i}{2}A, \qquad (29)$$

$$I_{n5} = -q\frac{\partial n_{i+1}}{\partial t}\frac{h_i}{2}A. \qquad (30)$$

The equivalent circuit for electron continuity equation is shown in Fig. 4(c). Similarly, the equivalent

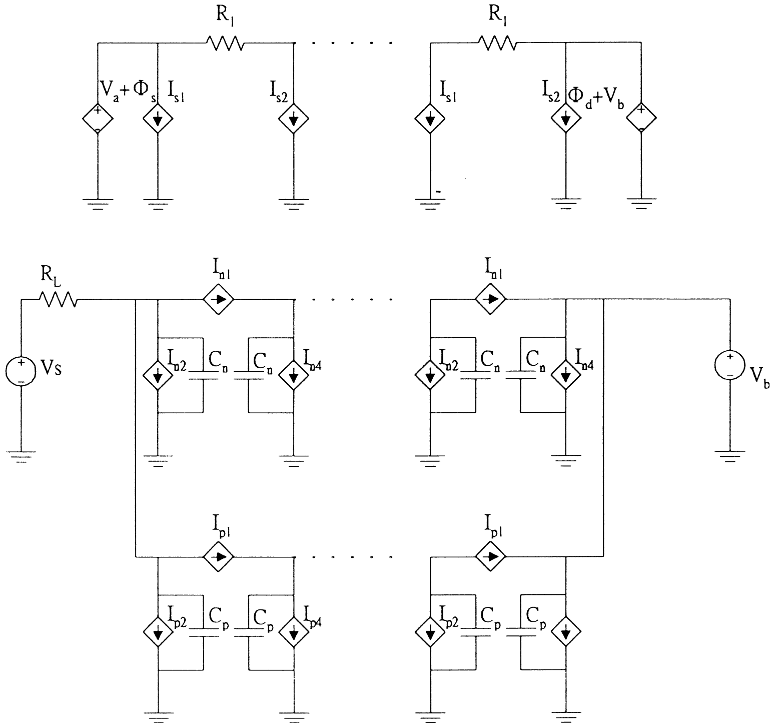

Fig. 7. Equivalent circuit of the diode switching circuit.

circuit model for the hole continuity equation can be obtained as shown in Fig. 4(d).

4. Simulation example

This section is devoted to discuss the implementation, using the levelized incomplete LU method and model proposed in Section 3. A diode switching circuit simulation is taken as an example for mixed-level simulation.

A simple diode switching circuit shown in Fig. 5 is first taken as an example for mixed-level simulation. The PN diode is partitioned into 251 elements. Its one-dimensional nonuniform grid partition is shown in Fig. 6. The voltage source V_s increases from 0 to 2 V and decreases to −1 V. The electron mobility $\mu_n = 1350$ cm^2/V s, and hole mobility $\mu_p = 400$ cm^2/V s. The carrier lifetimes τ_n and τ_p are equal to 1×10^{-6} s. The diode has the dopings and geometry as shown in Fig. 5. Using the model proposed in Section 3, the diode switching circuit has been transferred to the equivalent circuit as shown in Fig. 7. Before the process of simulation, we are interested in the influence of permutation in matrix A. Therefore, we compared two different sequences:

1. $[\varphi_1\varphi_2\ldots\varphi_N\phi_{n1}\phi_{n2}\ldots\phi_{nN}\phi_{p1}\phi_{p2}\ldots\phi_{pN}]^{\mathrm{T}}$ named as the sequential method.
2. $[\varphi_1\phi_{n1}\phi_{p1}\varphi_2\phi_{n2}\phi_{p2}\ldots\varphi_N\phi_{nN}\phi_{pN}]^{\mathrm{T}}$ named as the interleaving method.

The circuit simulation results are shown in Figs. 8–12. Fig. 8 is the waveform of V_s and V_d in the mixed-level device and circuit transient simulation. Fig. 9 is the waveform of the diode current. The result of simulation is reasonable and acceptable. Fig. 10 shows the simulation time with the different truncation parameter by using the interleaving method. In Fig. 10, when $\alpha = 2$, we have the shortest time with the factor of 0.619. In order to compare with the sequential method, we obtained Figs. 11 and 12. Note that in Fig. 11, α varies from 1 to 10, because the CPU time of the sequential

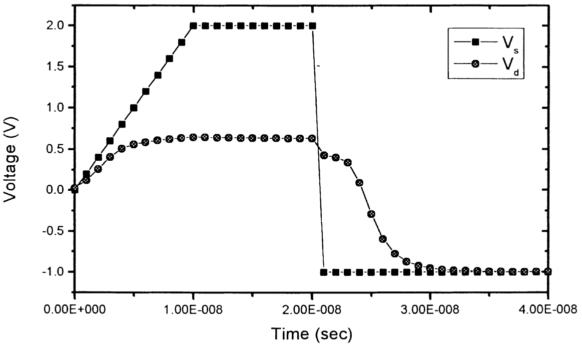

Fig. 8. The driving voltage waveform and the voltage waveform across the diode.

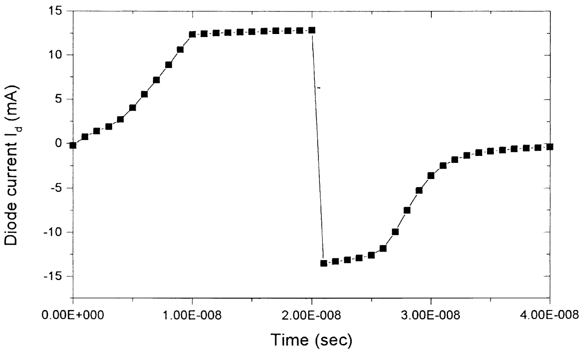

Fig. 9. The waveform of diode current I_d.

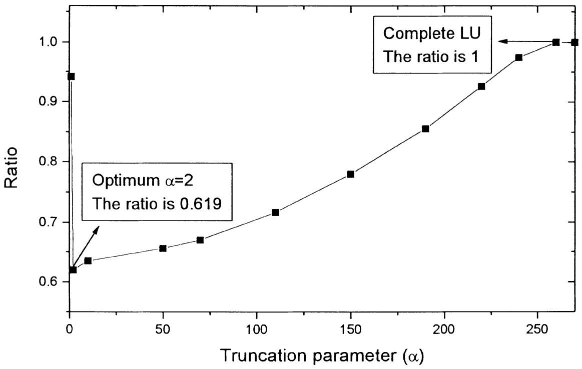

Fig. 10. Dependence of the computation effort of the levelized incomplete LU method with the different truncation parameter, α, for the diode switching circuit simulation by using the interleaving method. The CPU times are normalized with respect to the time for the completed LU method.

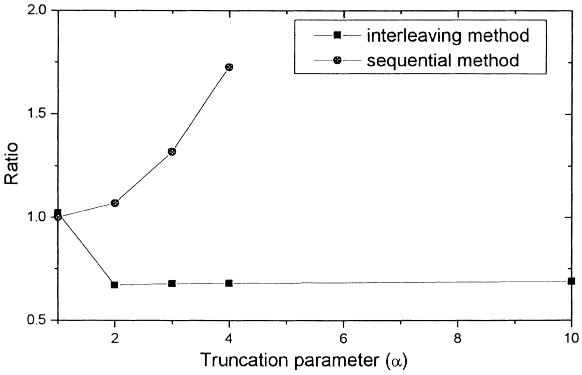

Fig. 11. The CPU time ratio normalized with respect to the time for the sequential method at $\alpha = 1$ versus the different truncation parameter of the interleaving method and the sequential method.

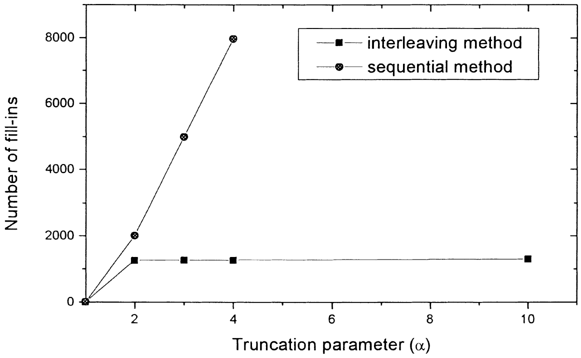

Fig. 12. The number of fill-ins versus the different truncation parameter of the interleaving method and the sequential method.

method increases too rapidly with α. In Fig. 11, the Y axis is the ratio of CPU time normalized with the time of the sequential method at $\alpha = 1$. For most α, the interleaving method is much faster than the sequential method except at $\alpha = 1$. Hence, the interleaving method is very useful in mixed-level simulation. When $\alpha = 0$, both methods fail to converge. Fig. 12 shows the fill-in comparison between the sequential method and interleaving method. It can be found that the fill-in number in the sequential method increases faster than that in the interleaving method. This is why the interleaving method is faster than the sequential method as shown in Fig. 11.

5. Conclusions

We have presented a technique for matrix solution that has distinct advantages over direct solution and traditional iterative solution. By adjusting the truncation parameter, we can mix and dynamically change the weighting between the direct and iteration method. This property gets rid of the disadvantages of the direct and iterative methods, while it inherits the advantages. It has been applied to a resistive grid and diode switching circuit. The results show that levelized incomplete LU method is a robust method.

On the contrary, an environment for circuit and semiconductor device mixed-level simulation has been built up. For node numbering in device equations, the interleaving method is found to be better than the sequential method. A pn diode switching circuit is taken as an example to verify the improvement.

References

[1] Leblebici Y, Unlu MS, Morkoc H, Kang SM. IEEE J Light Wave Technol 1995;13:396.

[2] Sherman AH. SIAM J Numer Anal 1978;15:755.

[3] Young DM. Iteration solution of large linear systems. New York: Academic Press, 1971.

[4] Massobriio G, Antognetti P. Semiconductor device modeling with SPICE. New York: McGraw-Hill, 1993.

[5] Selberherr S. Analysis and simulation of semiconductor devices. Berlin: Springer, 1984.