Waveform Relaxation for Transient Simulation of Two-Dimensional MOS Devices

Mark Reichelt, Jacob White, Jonathan Allen

Research Laboratory of Electronics
Dept. of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

In this paper we present experimental results demonstrating the effectiveness of waveform relaxation (WR) for solving the large, sparsely-connected algebraic and differential system generated by standard spatial discretization of the 2-D time-dependent semiconductor device equations. The experiments demonstrate that WR converges in a uniform manner, and that there is typically some multirate behavior in a device that the WR algorithm can exploit. Speed and accuracy comparisons are made between standard direct methods, red/black Gauss-Seidel WR, and red/black overrelaxed WR. For our experiments, calculated terminal currents matched well between the methods, and overrelaxed WR was up to a factor of 3 faster than direct methods.

1 Introduction

The accuracy of a circuit simulator is limited by the inaccuracies of the device models it employs. For most applications, the analytic MOS models used in programs like SPICE [4] accurately reflect the behavior of terminal currents and charges, but in some cases, these models are inadequate. For example, charge distribution must be computed accurately when simulating MOS comparator circuits or switched-capacitor filters. In addition, distributed effects in power MOS devices cannot be ignored when considering their efficiency as transistor switches. A more accurate, but computationally expensive, way to simulate these difficult circuits is to use a mixed circuit/device simulator such as CODECS [3]. In a mixed circuit/device simulator, circuit behavior is computed by solving the Poisson equation and the drift-diffusion partial differential equations for each device, while simultaneously solving the equations governing circuit operation. This is like incorporating a semiconductor device simulator such as PISCES [5] or MINIMOS [8] into a circuit simulator.

The enormous computational expense and the growing importance of mixed circuit/device simulation, as well as the current trend towards parallel computation, suggest that specialized parallel algorithms be developed for transient simulation of MOS devices that need only be effective in the device's standard operating region. For this reason, we are investigating accelerating transient device simulation with a waveform relaxation (WR) algorithm. WR has several advantages that may make it more effective than standard direct methods on parallel processors: WR is an iterative method and therefore avoids solving large sparse matrices directly, different sections of a device can be simulated with different timesteps, and finally, it is well-known that the decomposed fashion in which the WR equations are solved is suitable for parallel implementations [10] [9].

We start, in the next section, by describing the drift-diffusion device simulation equations, and then show that standard spatial discretization techniques convert these equations into a large, sparsely-connected system of algebraic and differential equations. Section 3 is a description of how WR can be applied to the spatially-discretized device equations. In Section 4 we present experimental results from our 2-D MOS device simulation program and compare WR, overrelaxed WR, and direct solution methods. Finally, in Section 5 we present conclusions and acknowledgements.

2 Device Simulation

A device is assumed to be governed by the Poisson equation, and the electron and hole continuity equations:

$$\begin{aligned}
\frac{\epsilon k T}{q}\nabla^2 u + q\,(p - n + N_D - N_A) &= 0 \\
\nabla \cdot \mathbf{J}_n - q\left(\frac{\partial n}{\partial t} + R\right) &= 0 \\
\nabla \cdot \mathbf{J}_p + q\left(\frac{\partial p}{\partial t} + R\right) &= 0
\end{aligned}$$

where u is the normalized electrostatic potential, n and p are the electron and hole concentrations, $\mathbf{J}_n$ and $\mathbf{J}_p$ are the electron and hole current densities, N_D and N_A are the donor and acceptor concentrations, R is the net generation and recombination rate, q is the magnitude of electronic charge, and ϵ is the dielectric permittivity [1], [8].

The current densities $\mathbf{J}_n$ and $\mathbf{J}_p$ are given by the drift-diffusion approximations:

$$\begin{aligned}
\mathbf{J}_n &= -qD_n\,(n\,\nabla u - \nabla n) \\
\mathbf{J}_p &= -qD_p\,(p\,\nabla u + \nabla p)
\end{aligned}$$

where D_n and D_p are the diffusion coefficients. In these equations, the diffusion constants are assumed to be related to the electron and hole mobilities by the Einstein relations. $\mathbf{J}_n$ and $\mathbf{J}_p$ are typically eliminated from the continuity equations using the drift-diffusion approximations, leaving a differential-algebraic system of three equations in three unknowns, u, n, and p.

Given a rectangular mesh that covers a two-dimensional slice of a MOSFET, a common approach to spatially discretizing the device equations is to use a finite-difference formula to discretize the Poisson equation, and an exponentially-fit finite-difference formula to discretize the continuity equations (the Scharfetter-Gummel method) [1] [8]. The discretized Poisson equation at each mesh node i, $f_1(u_i, n_i, p_i, u_j) = 0$, is:

$$\frac{\epsilon kT}{q}\sum_j\left\{\frac{d_{ij}}{L_{ij}}(u_i-u_j)\right\} - qA_i\left(p_i - n_i + N_D - N_A\right) = 0$$

where the sum is taken over the four nodes adjacent to i (north, south, east, and west), d_{ij} is the distance from node i to node j, and L_{ij} is the length of the perpendicular bisector of the edge between nodes i and j. The discretized electron continuity equation with the drift-diffusion approximation, $f_2(dn_i/dt, u_i, n_i, u_j, n_j) = 0$, is:

$$qD_n\sum_j\left\{\frac{d_{ij}}{L_{ij}}\Big[n_jB(u_j-u_i) - n_iB(u_i-u_j)\Big]\right\} - \qquad qA_i\left(\frac{dn_i}{dt}+R_i\right) = 0$$

where $B(u) = u/(e^u - 1)$ is the Bernoulli function, used to exponentially fit potential variation to electron concentration variation. The discretized hole continuity equation, $f_3(dp_i/dt, u_i, p_i, u_j, p_j) = 0$, is similar.

If there are N mesh nodes, then the result of this spatial discretization is a sparse differential-algebraic system of $3N$ equations in as many unknowns. The Poisson equations generate N algebraic constraints on the $2N$ nonlinear ordinary differential equations formed from the electron and hole continuity equations. At least one thousand mesh nodes are typically needed to accurately represent a 2-D slice of a MOS transistor, so that simulating a circuit where even a few transistors are treated by numerically solving the device equations leads to an enormous coupled system of algebraic and differential equations.

3 WR for Device Simulation

The standard approach used to solve the differential-algebraic system generated by spatial discretization of the device equations is to discretize the d/dt terms with a low order integration method such as the second-order backward difference formula. The result is a sequence of nonlinear algebraic systems in $3N$ unknowns, each of which can be solved with some variant of Newton's method and/or relaxation [3]. Another approach is to apply relaxation directly to the differential-algebraic equation system with a WR algorithm [2], such as that in Figure 1.

```
guess $u^0, n^0, p^0$ waveforms at all nodes
for $k = 0, 1, 2, \ldots$ until converged {
  for each node $i$ {
    solve for $u_i^{k+1}, n_i^{k+1}, p_i^{k+1}$ waveforms:
        $f_1(u_i^{k+1}, n_i^{k+1}, p_i^{k+1}, u_j^k) = 0$
        $f_2(dn_i^{k+1}/dt, u_i^{k+1}, n_i^{k+1}, u_j^k, n_j^k) = 0$
        $f_3(dp_i^{k+1}/dt, u_i^{k+1}, p_i^{k+1}, u_j^k, p_j^k) = 0$
  }
}
```

Figure 1: The point Gauss Jacobi WR algorithm.

A WR algorithm reduces the problem of simultaneously solving the $2N$ differential equations and N algebraic equations to one of iteratively solving $3N$ independent equations. At each mesh node i, the equations governing the $u_i(t)$, $n_i(t)$, and $p_i(t)$ waveforms can be solved with a numerical integration method such as the second-order backward difference formula. The inherent advantage of the WR approach is that the differential equations are solved independently, and therefore different sets of timesteps can be used at different mesh nodes to calculate the time evolution of u, n, and p. Therefore, if the device exhibits multirate behavior, WR can be very efficient provided it converges rapidly enough.

4 2-D MOS Transistor Experiments

In this section we present results from experiments with our 2-D WR-based transient device simulation program. The program computes transient behavior using either block WR or direct methods. For either solution technique, the differential-algebraic equations are solved using the second-order backward difference formula with a standard local truncation error timestep control scheme, the implicit algebraic systems generated by the backward difference formula are solved with Newton's method [1], and the linear equation systems generated by Newton's method are solved with sparse Gaussian elimination. Convergence is determined by testing for convergence of the potential and electron concentrations, as well as the convergence of the terminal currents. The simulator is written in C, and uses the Berkeley Sparse 1.3 sparse matrix solver written by K. Kundert. All experiments were run on a Sun-4 260.

In the experiments below, the device simulated is an n channel MOSFET with a $2.2\mu m$ channel length, an oxide thickness of $50\,nm$, a drain and source p^+ doping of $N_a = 10^{20}\,cm^{-3}$, an abrupt junction depth of $0.2\,\mu m$, a substrate doping of $N_a = 2.5 \times 10^{16}\,cm^{-3}$, and a channel implant of $N_a = 10^{16}\,cm^{-3}$ that extends to a depth of $50\,nm$. Dirichlet boundary conditions were imposed by a gate contact and by ohmic contacts at the drain, the source, and along the bottom of the substrate. Neumann reflecting boundary conditions were imposed along the left and right edges of the region.

In order to test both the low and high current case, the gate contact was held at $5\,v$, the source at $0\,v$, the substrate at $0\,v$, and the drain voltage was raised linearly from $0\,v$ to $5\,v$ over $3\,ns$, and then fixed at $5\,v$. The entire simulation interval was $30\,ns$. The experimental setup is illustrated in Figure 2.

The MOS device was spatially discretized on three different tensor product meshes (19x31, 23x31, and 23x33). In all three meshes, the grid lines were placed closer together at points where u, n, and p were expected to exhibit rapid spatial variation. In the largest problem, the 23x33 mesh (23 rows and 33 columns), there were 586 silicon nodes and 107 oxide nodes, so that this problem contained 1865 sparsely coupled algebraic

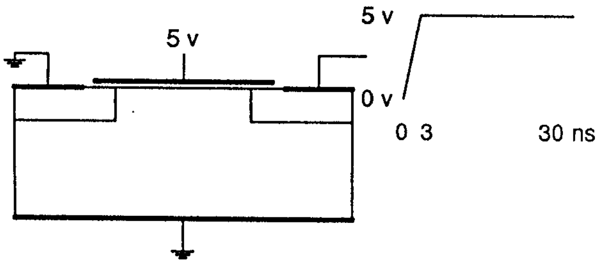

Figure 2: The transient simulation setup.

and differential equations in as many unknowns.

To solve with block WR, the mesh was broken into blocks defined by its vertical lines. In order to obtain accurate drain current calculations, the two vertical lines next to the drain were always blocked together. The equations governing nodes in the same block were solved simultaneously using Newton's method and sparse Gaussian elimination. The blocks were processed in red/black order as this maximizes parallelizability.

4.1 Uniformity of Convergence

Previous theoretical results guarantee the convergence of the WR algorithm for the device simulation problem, and they suggest that the WR algorithm will converge in such a way that on each WR iteration, all the timepoints in a waveform will move closer to the correct solution [2], [6]. This uniformity of convergence is essential to WR efficiency. In order to demonstrate that it occurs in practice (or at least in our examples), we compare the number of WR iterations required to solve the 19x31 mesh over the 30 *ns* simulation interval (106 iterations) to the number of iterations required to solve the same mesh over just the first 1 *ns* of the 30 *ns* interval (92 iterations). The number of iterations required is about the same, indicating that WR is converging uniformly over the entire 30 *ns* simulation interval.

That the WR algorithm converges in a uniform manner suggests that it will be possible to accelerate WR convergence with overrelaxation. Like successive overrelaxation (SOR) for algebraic problems, waveform SOR (WSOR) involves modifying the waveform iterates by pushing them further (or not as far) in the iteration direction by some parameter $\omega \in [0,2]$. The impact of SOR on WR efficiency is discussed in a later section.

4.2 Multirate Behavior

Figure 3 illustrates the number of timepoints required per block to solve the 23x33 mesh with WR. Different blocks required different numbers of timesteps, indicating that in practice some multirate behavior can be exploited by WR. It is interesting to note that different nodes changed at different rates, not so much because the electron and hole concentrations changed at different times, but because they changed by different orders of magnitude (*multimagnitude* behavior).

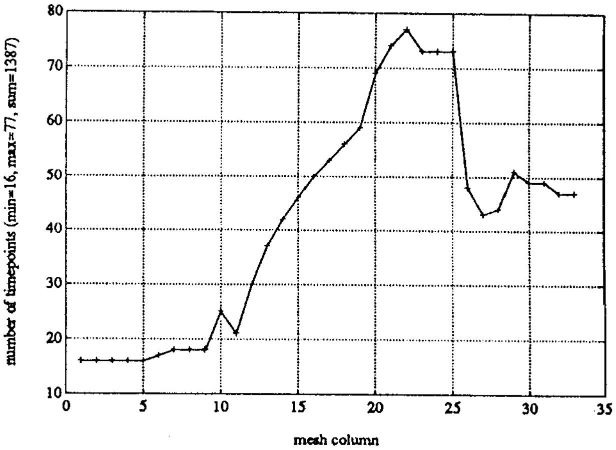

Figure 3: Multirate behavior in the k23x33 mesh.

4.3 Comparisons to Direct Solution

To solve with WSOR, we proceed as with WR and after twenty WR iterations over all blocks, we accelerate convergence by setting the overrelaxation factor to some fixed value $1 < \omega < 2$. In the examples shown here, $\omega \approx 1.5$.

device	number of unknowns	CPU sec direct	CPU sec WR (GS)	CPU sec WSOR
k19x31	1379	3324.31	5042.42	2791.50
k23x31	1751	6449.68	6218.78	3335.57
k23x33	1865	8726.60	6712.23	3767.99

Table 1: Comparison of CPU times for WR, WSOR and direct solution.

The results show that WR is competitive with direct solution, and that WSOR is faster. We found in general that the red/black ordering of the blocks didn't worsen the WR or WSOR convergence rate compared to natural ordering. Because we used red/black ordering, and there were more than 30 blocks, a further factor of 15 speedup could be obtained by implementing the algorithm on a parallel machine.

As Figure 4 illustrates for the drain current, there was no significant difference in terminal current calculations between WR and direct solution over the full range from zero current to the maximum operating current. The accuracy can be improved further by tightening the WR convergence tolerances. As mentioned above, to obtain this accuracy, it was necessary to block the pair of vertical lines next to the drain together. This insured that the potentials at both ends of the edges of maximal current flow were solved at the same timepoints, eliminating any interpolation error.

Figure 5 shows the electron concentrations calculated by WR, at the silicon-oxide interface in the channel of the 19x31 mesh.

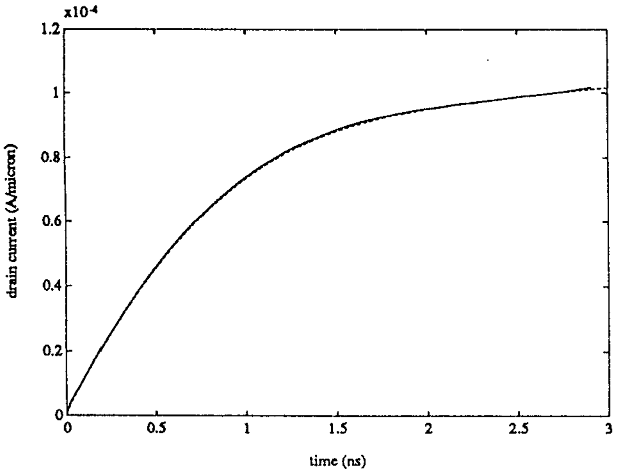

Figure 4: Drain current in the 19x31 mesh for direct and WR.

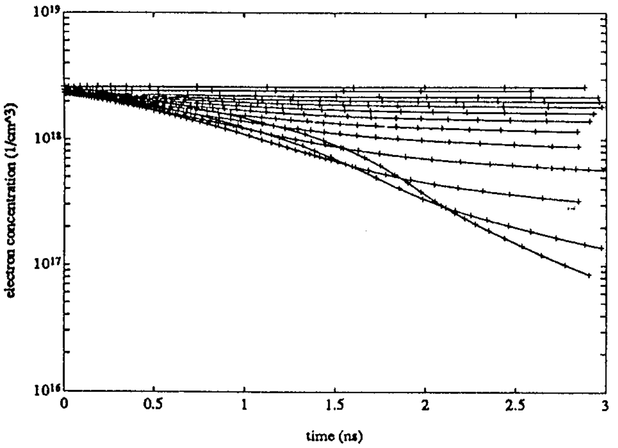

Figure 5: Electron concentrations in the channel of the 19x31 mesh calculated by WR.

5 Conclusions and Acknowledgements

In this paper, applying WR to 2-D MOS device transient simulation is investigated. These preliminary experimental results show that WR converges in a fairly uniform manner, that some multirate behavior can be exploited, and that an appropriately blocked and overrelaxed WR method can be faster by almost a factor of 3 than direct methods. In addition, since red/black Gauss-Seidel was used, the algorithm has substantial easily-exploited parallelism. Also, the results presented are for relatively small problems, with less than 2000 unknowns, and we expect that larger problems, such as more accurate 2-D simulation or 3-D simulation, will make the WR algorithms even more effective.

There are a variety of techniques that can futher improve WR performance. We are currently working on refining the timesteps with iterations, and using a single waveform-Newton iteration to solve the nonlinear WR equations. With these additional techniques, WR promises to be a fast and easily parallelizable technique for transient device simulation, a good platform upon which to build a mixed circuit/device simulator.

The authors would like to thank Bob Armstrong and the other members of the custom integrated circuits group at MIT for their assistance, and F. Odeh, L. Reyna and A. Ruehli of the IBM T. J. Watson research center for many valuable discussions. This work was supported by a grant from IBM, the Defense Advanced Research Projects Agency contract N00014-87-K-825, and the National Science Foundation.

References

[1] R.E. Bank, W.C. Coughran Jr. W. Fichtner, E.H. Grosse, D.J. Rose, and R.K. Smith, "Transient Simulation of Silicon Devices and Circuits", IEEE Transactions on Computer-Aided Design, Vol. CAD-4, No. 4, October 1985, pp. 436-451.

[2] E. Lelarasmee, A. Ruehli, A. Sangiovanni-Vincentelli, "The Waveform Relaxation Method for the Time Domain Analysis of Large Scale Integrated Circuits." *IEEE Trans. on CAD*, Vol. 1, No. 3, July 1982.

[3] K. Mayaram, D.O. Pederson, "CODECS: A Mixed-Level Device and Circuit Simulator," *Proc. Int. Conf. on Computer-Aided Design*, Santa Clara, California, October 1988.

[4] L.W. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits", Electronic Research Laboratory *Report No. ERL-M520*, University of California, Berkeley, May 1975.

[5] C.S. Rafferty, M.R. Pinto, and R.W. Dutton, "Iterative Methods in Semiconductor Device Simulation", IEEE Transactions on Computer-Aided Design, Vol. CAD-4, No. 4, October 1985, pp. 462-471.

[6] M. Reichelt, J. White, J. Allen and F. Odeh, "Waveform Relaxation Applied to Transient Device Simulation," *1988 Int'l. Symp. on Circuits and Systems*, Espoo, Finland.

[7] M. Reichelt and J. White, "Techniques for Switching Power Converter Simulation", NASECODE VI, Dublin, Ireland, July 1989.

[8] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*, Springer-Verlag, New York, 1984.

[9] S. Vandewalle and D. Roose, "The Parallel Waveform Relaxation Multigrid Method", *Parallel Processing for Scientific Computing*, SIAM, Philadelphia, 1989, pp. 152-156.

[10] J. White, A. Sangiovanni-Vincentelli, *Relaxation Techniques for the Simulation of VLSI Circuits* Kluwer Academic Publishers, Norwell, Massachusetts, 1986.