Conjugate Direction Waveform Methods for Transient Two-Dimensional Simulation of MOS Devices

Andrew Lumsdaine Mark Reichelt Jacob White

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

Abstract

In this paper, a conjugate-direction based acceleration to the waveform relaxation algorithm is derived and then applied to solving the differential-algebraic system generated by spatial discretization of the time-dependent semiconductor device equations. In the experiments included, the conjugate-direction waveform methods are up to 15 times faster than ordinary WR.

1 Introduction

The enormous computational expense and the growing importance of mixed circuit/device simulation, as well as the increasing availability of parallel computers, suggest that specialized, easily parallelized, algorithms be developed for transient simulation of MOS devices [1]. Recently, the easily parallelized waveform relaxation (WR) algorithm was shown to be a computationally efficient approach to device transient simulation [2], even though the WR algorithm typically requires hundreds of iterations to achieve an accurate solution. In this paper we derive a conjugate-direction based acceleration for the WR algorithm and present experimental results in which this acceleration reduces the computation time by up to a factor of 15.

In the next section, we start by briefly describing the standard equations that are solved to perform device transient simulation. The WR algorithm for solving these equations is given in Section 3 and a particular waveform conjugate direction method—waveform GCR—is derived. We present experimental results from our 2-D MOS device transient simulation program in Section 4 and compare two conjugate-direction waveform methods, waveform GMRES and waveform GCR, with standard WR and WRN. Finally, in Section 5 we present our conclusions and suggestions for future work.

2 Device Simulation

Device transient simulation is usually performed by numerically solving the coupled Poisson and time-dependent electron and hole current-continuity equations. To solve this coupled system, the equations are first discretized in space using exponentially-fit finite-difference or finite-volume methods [3]. On an N-node rectangular mesh, the spatial discretization yields a differential-algebraic system of $3N$ equations in $3N$ unknowns denoted by

$$\begin{aligned} \boldsymbol{f}_1(\boldsymbol{u}(t),\boldsymbol{n}(t),\boldsymbol{p}(t)) &= 0 && (1)\\ \boldsymbol{f}_2(\boldsymbol{u}(t),\boldsymbol{n}(t),\boldsymbol{p}(t)) &= \tfrac{d}{dt}\boldsymbol{n}(t) && (2)\\ \boldsymbol{f}_3(\boldsymbol{u}(t),\boldsymbol{n}(t),\boldsymbol{p}(t)) &= \tfrac{d}{dt}\boldsymbol{p}(t) && (3) \end{aligned}$$

where $t \in [0,T]$, and $\boldsymbol{u}(t),\boldsymbol{n}(t),\boldsymbol{p}(t) \in \mathbf{R}^N$ are vectors of normalized potential, electron concentration, and hole concentration, respectively. Here, $\boldsymbol{f}_1,\boldsymbol{f}_2,\boldsymbol{f}_3 : \mathbf{R}^{3N} \rightarrow \mathbf{R}^N$ are specified component-wise as

$$f_{1_i}(u_i,n_i,p_i,u_j) = \frac{\epsilon kT}{q}\sum_j \frac{d_{ij}}{L_{ij}}(u_i-u_j) - qA_i\,(p_i - n_i + N_D - N_A)$$

$$f_{2_i}(u_i,n_i,u_j,n_j) = \frac{D_n}{A_i}\sum_j \frac{d_{ij}}{L_{ij}}\Big[n_j B(u_j-u_i) - n_i B(u_i-u_j)\Big] - R_i$$

$$f_{3_i}(u_i,p_i,u_j,p_j) = \frac{D_p}{A_i}\sum_j \frac{d_{ij}}{L_{ij}}\Big[p_j B(u_i-u_j) - p_i B(u_j-u_i)\Big] - R_i.$$

The sums above are taken over the four nodes adjacent to node i (north, south, east, and west), L_{ij} is the

ALGORITHM 1 (WR for Device Simulation).

guess u^0, n^0, p^0 waveforms at all nodes
for $k = 0, 1, \ldots$ until converged
  for each node i
    solve for $u_i^{k+1}, n_i^{k+1}, p_i^{k+1}$ waveforms:

$$\begin{aligned} f_{1_i}(u_i^{k+1}, n_i^{k+1}, p_i^{k+1}, u_j^k) &= 0 \\ f_{2_i}(u_i^{k+1}, n_i^{k+1}, u_j^k, n_j^k) &= \tfrac{d}{dt} n_i^{k+1} \\ f_{3_i}(u_i^{k+1}, p_i^{k+1}, u_j^k, p_j^k) &= \tfrac{d}{dt} p_i^{k+1} \end{aligned}$$

distance from node i to node j, d_{ij} is the length of the side of the Voronoi box that encloses node i and bisects the edge between nodes i and j, and $B(v) = v/(e^v - 1)$ is the Bernoulli function.

3 Waveform GCR

Typically, (1)–(3) are solved using a low-order implicit time-integration scheme, combined with a Newton or relaxation method to solve the generated sequence of nonlinear algebraic equations [4, 5]. Another approach is to apply relaxation directly to the differential-algebraic equation system as in the waveform relaxation (WR) algorithm [2, 6], given in Algorithm 1. The WR algorithm has several advantages: it is an iterative method and therefore avoids factoring large sparse matrices; WR can exploit multi-rate behavior as different solution components can use different timesteps; and finally, WR is well suitabed to parallel computation. However, when applied to solving (1)–(3), the WR algorithm converges slowly. In this section, we derive a waveform generalization of the conjugate-direction method and show how to use this method to accelerate the convergence of WR. As the derivation will make clear, waveform conjugate-direction methods retain most of the advantages of waveform relaxation.

We begin the derivation by first considering the linear time-varying inital-value problem (IVP),

$$\begin{aligned} (\tfrac{d}{dt} + \boldsymbol{A}(t))\boldsymbol{x}(t) &= \boldsymbol{b}(t) \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0, \end{aligned} \qquad (4)$$

where $\boldsymbol{A}(t) \in \mathbf{R}^{N \times N}$, $\boldsymbol{b}(t) \in \mathbf{R}^N$ is a given right-hand side, and $\boldsymbol{x}(t) \in \mathbf{R}^N$ is the unknown vector to be computed over the simulation interval $t \in [0, T]$. Since the $\frac{d}{dt}$ operator is not self-adjoint, and since, in general, $\boldsymbol{A}(t)$ may be non-symmetric, only waveform extensions of conjugate-direction methods suitable for non-symmetric problems need be considered

ALGORITHM 2 (GCR).

Set $\boldsymbol{p}^0 = \boldsymbol{r}^0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^0$
For $k = 0, 1, \ldots$ until $\langle \boldsymbol{r}^k, \boldsymbol{r}^k \rangle < \epsilon$
  Minimize $\langle \boldsymbol{r}^{k+1}, \boldsymbol{r}^{k+1} \rangle$ in direction of $\boldsymbol{p}^k$
  $\alpha = \frac{\langle \boldsymbol{A}\boldsymbol{p}^k, \boldsymbol{r}^k \rangle}{\langle \boldsymbol{A}\boldsymbol{p}^k, \boldsymbol{A}\boldsymbol{p}^k \rangle}$
  $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha \boldsymbol{p}^k$
  $\boldsymbol{r}^{k+1} = \boldsymbol{r}^k - \alpha \boldsymbol{A}\boldsymbol{p}^k$
  $\boldsymbol{p}^{k+1} = \boldsymbol{r}^{k+1} + \sum_{j=0}^{k} \beta_j^{(k)} \boldsymbol{p}^j$
    where $\{\beta_j^{(k)}\}$ are chosen so that
    $\langle \boldsymbol{A}\boldsymbol{p}^{k+1}, \boldsymbol{A}\boldsymbol{p}^j \rangle = 0$ for $0 \le j \le k$

for solving (4). In particular, we extend the generalized conjugate residual algorithm (GCR), given in Algorithm 2, since GCR is the prototypical conjugate-direction method for solving non-symmetric linear algebraic systems [7].

The difference between algebraic GCR and waveform GCR (WGCR) is that the inner products and the matrix-vector products in Algorithm 2 must instead refer to waveform inner products and operator-waveform products, respectively. For the waveform inner product, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$, we use the familiar $\mathbf{L}_2$ inner product given by

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{N} \int_0^T x_i(t) y_i(t) dt. \qquad (5)$$

For this version of WGCR, which can be interpreted as using a Gauss-Jacobi dynamic preconditioner (Gauss-Seidel preconditioning is used in the actual implementation), computing the operator-waveform product involves solving a differential equation. Specifically, if

$$\boldsymbol{w} \equiv \boldsymbol{A}\boldsymbol{p}, \qquad (6)$$

then $\boldsymbol{w}$ is computed by first solving

$$\begin{aligned} \left(\tfrac{d}{dt} + a_{ii}(t)\right) y_i(t) + \sum_{j \neq i} a_{ij}(t) p_j(t) &= 0 \\ y_i(0) &= 0, \end{aligned} \qquad (7)$$

for $\boldsymbol{y}$, and then setting $\boldsymbol{w} = \boldsymbol{p} - \boldsymbol{y}$.

The initial residual, $\boldsymbol{r}^0$ in Algorithm 2, can be computed by first solving

$$\begin{aligned} \left(\tfrac{d}{dt} + a_{ii}(t)\right) y_i(t) + \sum_{j \neq i} a_{ij}(t) x_j^0(t) - b_i(t) &= 0 \\ y_i(0) &= 0, \end{aligned} \qquad (8)$$

ALGORITHM 3 (Nonlinear Waveform GCR).

Pick x^0, ϵ^0, $\nu < 1$
For $m = 0, 1, \ldots$ until $\langle r^m, r^m \rangle < \phi$,
  Linearize (1)–(3) to form (9)
  Solve (9) with Algorithm 2 using ϵ^m
  Update x^{m+1} and r^{m+1}
  Set $\epsilon^{m+1} = \epsilon^m \cdot \nu$

for y, and then setting $r^0 = y - x^0$. Subsequent updates to the residual are computed using the formula given in Algorithm 2.

Note that one iteration of linear Gauss-Jacobi waveform relaxation is embedded in both (7) and (8).

For nonlinear problems, the WGCR algorithm can be combined with a waveform-Newton algorithm [8, 9, 10]. Applying the waveform-Newton algorithm to solving (1)–(3), we obtain the following iteration:

$$\begin{bmatrix} 0 \\ \frac{d}{dt}n^{m+1} \\ \frac{d}{dt}p^{m+1} \end{bmatrix} + \begin{bmatrix} J_{f_{11}} & J_{f_{12}} & J_{f_{13}} \\ J_{f_{21}} & J_{f_{22}} & J_{f_{23}} \\ J_{f_{31}} & J_{f_{32}} & J_{f_{33}} \end{bmatrix} \begin{bmatrix} u^{m+1} \\ n^{m+1} \\ p^{m+1} \end{bmatrix} = \quad (9)$$
$$\begin{bmatrix} J_{f_{11}} & J_{f_{12}} & J_{f_{13}} \\ J_{f_{21}} & J_{f_{22}} & J_{f_{23}} \\ J_{f_{31}} & J_{f_{32}} & J_{f_{33}} \end{bmatrix} \begin{bmatrix} u^m \\ n^m \\ p^m \end{bmatrix} - \begin{bmatrix} f_1(u^m, n^m, p^m) \\ f_2(u^m, n^m, p^m) \\ f_3(u^m, n^m, p^m) \end{bmatrix}$$

where m is the Newton iteration index. Each step of (9) requires the solution of a linear time-varying differential-algebraic system, which can be accomplished with WGCR. This WN/WGCR algorithm, referred to as nonlinear WGCR, is shown in Algorithm 3, and we note the method is in the class of hybrid Krylov methods [11].

4 Experimental Results

Two waveform conjugate-direction methods were implemented in the WR-based device transient simulation program WORDS [2]: the nonlinear WGCR algorithm described in Section 3 as well as a nonlinear waveform GMRES (WGMRES) algorithm [11, 12]. The WORDS program uses red/black block Gauss-Seidel WR and WRN [8], where the blocks correspond to vertical mesh lines; the corresponding Gauss-Seidel preconditioner is used for the WGCR and WGMRES implementations. For all methods, the equations governing nodes in the same block are solved simultaneously using the second-order backward-difference formula. The implicit algebraic systems generated by the backward difference formula are solved with Newton's method and the linear equation systems generated by Newton's method are solved with sparse Gaussian elimination. Note that for the nonlinear WGCR and WGMRES algorithms, the linear algebraic equations at each timestep require only one algebraic Newton iteration.

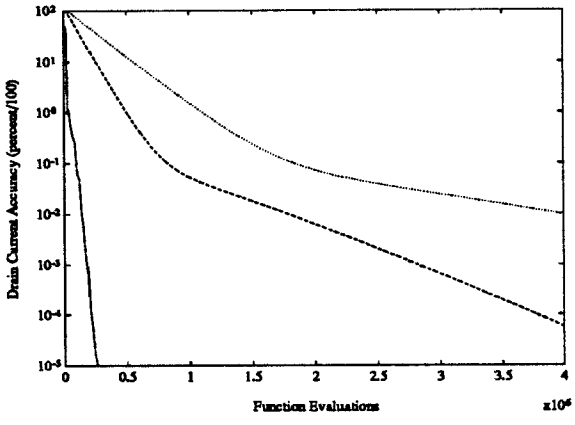

Figure 1: Convergence comparison between WR (dotted), WRN (dashed), and WGCR/WMRES (solid) for **jD** example. The max-norm of the relative drain terminal current error is plotted against the number of function evaluations.

Four N-channel MOSFET examples were used to compare the performance of the WR, WRN, nonlinear WGCR, and nonlinear WGMRES algorithms:

kG: 2.2 μm channel-length; 50 psec, 0-5V ramp on the gate with the drain held at 5V.

kD: 2.2μm channel-length; 50 psec, 0-5V ramp on the drain with the gate held at 5V.

jG: 0.17μm channel-length; 5 psec, 0-1V ramp on the gate with the drain held at 1V.

jD: 0.17μm channel-length; 5 psec, 0-1V ramp on the drain with the gate held at 1V.

The parameters used with the conjugate-direction methods were: $\epsilon^0 = 0.1$, $\nu = \sqrt{0.1}$, and $\phi = 1 \times 10^{-18}$. To simplify comparisons, 32 equally-spaced timesteps were used in all experiments.

Table 1 shows the number of function evaluations and the CPU time required for each of the waveform methods to reduce the max-norm of the drain terminal current error below 0.01% of the max-norm of the drain terminal current. As Table 1 indicates, conjugate-direction methods significantly reduced the number of function evaluations and CPU time over WR and WRN. In fact, in the **jD** example, WGMRES was 15 times faster than ordinary WR. As is

Example	Method	FEvals	CPU sec
jD	WR	8.43×10^6	14469
	WRN	3.77×10^6	7088
	WGCR	2.00×10^5	1050
	WGMRES	2.04×10^5	915
jG	WR	7.48×10^6	12615
	WRN	3.41×10^6	6214
	WGCR	1.97×10^5	1011
	WGMRES	1.97×10^5	877
kD	WR	1.22×10^6	1526
	WRN	3.94×10^5	559
	WGCR	9.03×10^4	315
	WGMRES	9.03×10^4	280
kG	WR	1.43×10^6	1756
	WRN	4.09×10^5	578
	WGCR	1.03×10^5	353
	WGMRES	1.03×10^5	316

Table 1: Comparison of WGCR, WGMRES, and WR. CPU times shown are for an IBM RS/6000 model 540.

common in the algebraic case, WGMRES and WGCR perform similarly, but WGMRES is computationally more efficient because it avoids several waveform inner products on each iteration. The graph in Figure 1 compares the convergence of WR, WRN, WGCR, and WGMRES for the **jD** example. In the graph, the terminal current error versus number of function evaluations is plotted and clearly demonstrates the rapid convergence of the conjugate-direction methods.

5 Conclusion

In this paper we derived waveform conjugate-direction methods for accelerating waveform relaxation. Experimental results demonstrated the effectiveness of the acceleration when solving the large, sparsely-connected algebraic and differential system generated by standard spatial discretization of the 2-D time-dependent semiconductor device equations. In the experiments included, the waveform conjugate-direction methods were up to 15 times faster than ordinary WR.

Future work is primarily focused on developing theoretical results about the convergence of linear and nonlinear WGCR, although a preliminary result for linear WGCR can be obtained by realizing that WGCR is a Galerkin method, known to converge for problems of this type [13].

Acknowledgments

The authors would like to thank the members of the custom integrated circuits group at MIT, especially Ibrahim Elfadel, and F. Odeh 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] W. Engl, R. Laur, and H. Dirks, "MEDUSA - A simulator for modular circuits," *IEEE Trans. CAD*, vol. 1, pp. 85–93, April 1982.

[2] M. Reichelt, J. White, and J. Allen, "Waveform relaxation for transient two-dimensional simulation of MOS devices," in *International Conference on Computer Aided-Design*, (Santa Clara, California), pp. 412–415, November 1989.

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

[4] R. Bank, W. Coughran, Jr., W. Fichtner, E. Grosse, D. Rose, and R. Smith, "Transient simulation of silicon devices and circuits," *IEEE Trans. CAD*, vol. 4, pp. 436–451, October 1985.

[5] K. Mayaram and D. Pederson, "CODECS: A mixed-level device and circuit simulator," in *International Conference on Computer Aided-Design*, (Santa Clara, California), pp. 112–115, November 1988.

[6] E. Lelarasmee, A. E. Ruehli, and A. L. Sangiovanni-Vincentelli, "The waveform relaxation method for time domain analysis of large scale integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 1, pp. 131–145, July 1982.

[7] H. C. Elman, *Iterative Methods for Large Sparse Nonsymmetric Systems of Linear Equations*. PhD thesis, Computer Science Dept., Yale University, New Haven, CT, 1982.

[8] J. K. White and A. Sangiovanni-Vincentelli, *Relaxation Techniques for the Simulation of VLSI Circuits*. Engineering and Computer Science Series, Norwell, Massachusetts: Kluwer Academic Publishers, 1986.

[9] R. Saleh and J. White, "Accelerating relaxation algorithms for circuit simulation using waveform-newton and step-size refinement," *IEEE Trans. CAD*, vol. 9, no. 9, pp. 951–958, 1990.

[10] D. Erdman and D. Rose, "A newton waveform relaxation algorithm for circuit simulation," in *International Conference on Computer Aided-Design*, (Santa Clara, California), pp. 404–407, November 1989.

[11] P. Brown and Y. Saad, "Hybrid Krylov methods for nonlinear systems of equations," *SIAM J. Sci. Statist. Comput.*, vol. 11, pp. 450–481, May 1990.

[12] Y. Saad and M. Schultz, "GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Statist. Comput.*, vol. 7, pp. 856–869, July 1986.

[13] R. Kress, *Linear Integral Equations*. New York: Springer-Verlag, 1989.