ACCELERATING DYNAMIC ITERATION METHODS WITH APPLICATION TO SEMICONDUCTOR DEVICE SIMULATION*

ANDREW LUMSDAINE† AND JACOB K. WHITE‡

Abstract. In this paper, we apply a Galerkin method to solving the system of second-kind Volterra integral equations that characterize the classical dynamic iteration methods for the linear time-varying initial-value problem. It is shown that the Galerkin approximations can be computed iteratively using conjugate-direction algorithms. The resulting iterative methods are combined with an operator Newton method and applied to solving the nonlinear differential-algebraic system generated by spatial discretization of the time-dependent semiconductor device equations. Experimental results are included that demonstrate the conjugate-direction methods are significantly faster than classical dynamic iteration methods.

Key words. Conjugate-direction methods, dynamic iteration, Galerkin method, waveform relaxation

AMS(MOS) subject classifications. 65L60, 65L05, 65R20, 65J10

1. Introduction.

Consider the problem of numerically solving the linear time-varying initial-value problem (IVP),

$$\begin{aligned} (\tfrac{d}{dt} + \boldsymbol{A}(t))\boldsymbol{x}(t) &= \boldsymbol{b}(t) \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0, \end{aligned} \tag{1}$$

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 $t \in [0, T]$. There are several approaches to solving the IVP. The traditional numerical approach is to begin by discretizing (1) in time with an implicit integration rule (since large dynamical systems are typically stiff) and then solving the resulting matrix problem at each time step. This approach can be disadvantageous for a parallel implementation, especially for MIMD parallel computers having a high communication latency, since the processors will have to synchronize repeatedly for each timestep.

A more suitable approach to solving the IVP with a parallel computer is to decompose the problem at the ODE level. That is, the large system is decomposed into smaller subsystems, each of which is assigned to a single processor. The IVP is solved iteratively by solving the smaller IVPs for each subsystem, using fixed values from previous iterations for the variables from other subsystems. This dynamic iteration process is known as waveform relaxation (WR) or sometimes as the Picard-Lindelöf iteration [13].

In this paper, we describe both theoretical and practical aspects of using conjugate-direction approaches to accelerate dynamic iteration convergence. In the next section, we begin by describing the system of second-kind Volterra integral equations obtained by applying a "dynamic preconditioner" to (1). A Galerkin method for solving an operator equation formulation of the integral equation system over a Krylov space is then described and a convergence result given. It is noted that certain conjugate-direction techniques applied to the integral equation system iteratively generate the Galerkin approximations. One such method, the waveform GMRES method, is described. In Section 3, we combine the waveform GMRES method with an operator-Newton algorithm to create a hybrid scheme for solving nonlinear initial-value problems. In Section 4, we present results from experiments where the hybrid scheme is used to solve the nonlinear differential-algebraic systems generated from two-dimensional spatial discretization of the time-dependent drift-diffusion equations used to describe transient phenomena in semiconductors. Finally, our conclusions and suggestions for future work are contained in Section 5.

* This work was supported by a grant from IBM, the Defense Advanced Research Projects Agency contract N00014-91-J-1698, and the National Science Foundation.

† This work was conducted while the first author was in the Dept. of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology. He is now with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556. (Andrew.Lumsdaine@nd.edu)

‡ Research Laboratory of Electronics, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139. (white@rle-vlsi.mit.edu)

2. Waveform Conjugate-Direction Derivation. In (1), let $\boldsymbol{A}(t) = \boldsymbol{M}(t) - \boldsymbol{N}(t)$, and consider the system of second kind Volterra integral equations given by

$$\boldsymbol{x}(t) - \boldsymbol{\Phi}_M(t,0)\boldsymbol{x}(0) - \int_0^t \boldsymbol{\Phi}_M(t,s)\boldsymbol{N}(s)\boldsymbol{x}(s)ds = \int_0^t \boldsymbol{\Phi}_M(t,s)\boldsymbol{b}(s)ds, \tag{2}$$

where $\boldsymbol{\Phi}_M$ is the state transition matrix [3] for the equation

$$\tfrac{d}{dt}\boldsymbol{x}(t) = \boldsymbol{M}(t)\boldsymbol{x}(t).$$

We assume throughout that $\boldsymbol{A}$, $\boldsymbol{M}$, and $\boldsymbol{N}$ are such that (1) and (2) each have a unique solution, a sufficient condition for which is that $\boldsymbol{A}$, $\boldsymbol{M}$, and $\boldsymbol{N}$ be piecewise continuous with respect to t. Note that (2) is obtained from (1) by the application of a "dynamic preconditioner," to both sides of (1). More precisely, this preconditioner, denoted $\mathcal{M}^{-1}$, is defined by:

$$(\mathcal{M}^{-1}\boldsymbol{x})(t) = \int_0^t \boldsymbol{\Phi}_M(t,s)\boldsymbol{x}(s)ds.$$

Informally, one can think of $\mathcal{M}^{-1}$ as roughly being $(\frac{d}{dt} + \boldsymbol{M}(t))^{-1}$.

Equation (2) can be expressed as an operator equation over a space $\mathbf{H}$ as

$$(\boldsymbol{I} - \mathcal{K})\boldsymbol{x} = \boldsymbol{\psi}, \tag{3}$$

where $\mathbf{H} = \mathbf{L}_2([0,T], \mathbf{R}^N)$, $\boldsymbol{I} : \mathbf{H} \to \mathbf{H}$ is the identity operator, $\mathcal{K} : \mathbf{H} \to \mathbf{H}$ is defined by

$$(\mathcal{K}\boldsymbol{x})(t) = \int_0^t \boldsymbol{\Phi}_M(t,s)\boldsymbol{N}(s)\boldsymbol{x}(s)ds$$

and $\boldsymbol{\psi} \in \mathbf{H}$ is given by

$$\boldsymbol{\psi}(t) = \boldsymbol{\Phi}_M(t,0)\boldsymbol{x}(0) + \int_0^t \boldsymbol{\Phi}_M(t,s)\boldsymbol{b}(s)ds.$$

The following are standard results (see, e.g., [5, 7]) which will be used in subsequent discussions of (3).

Lemma 2.1. If $\boldsymbol{M}$ and $\boldsymbol{N}$ are piecewise continuous with respect to t, then $\mathcal{K} : \mathbf{H} \to \mathbf{H}$ is compact, has a spectral radius of zero, and $\mathcal{K}^*$, the adjoint operator for $\mathcal{K}$, is given by

$$(\mathcal{K}^*\boldsymbol{x})(t) = \int_t^T \left[\boldsymbol{\Phi}_M(s,t)\boldsymbol{N}(t)\right]^\dagger \boldsymbol{x}(s)ds,$$

where superscript † denotes algebraic transposition.

It should be apparent from Lemma 2.1 that, in general, $\mathcal{K}$ is not self adjoint. We therefore restrict our attention to those conjugate-direction methods which are appropriate for non-self-adjoint operators.

2.1. Classical Dynamic Iteration Methods. The classical dynamic iteration is obtained by applying the Richardson iteration to the "preconditioned" problem (3):

$$\boldsymbol{x}^{k+1} = \mathcal{K}\boldsymbol{x}^k + \boldsymbol{\psi}. \tag{4}$$

This approach is known as the method of successive approximations, waveform relaxation, or the Picard-Lindelöf iteration [1, 7, 9, 13, 24].

Example. Let $\boldsymbol{M}(t) = 0$. Then $\boldsymbol{\Phi}_M = \boldsymbol{I}$ so that (2) becomes

$$\boldsymbol{x}(t) - \boldsymbol{x}(0) + \int_0^t \boldsymbol{A}(s)\boldsymbol{x}(s)ds = \int_0^t \boldsymbol{b}(s)ds.$$

The corresponding dynamic iteration is

$$\boldsymbol{x}^{k+1}(t) = \boldsymbol{x}(0) + \int_0^t \left(\boldsymbol{b}(s) - \boldsymbol{A}(s)\boldsymbol{x}^k(s)\right) ds$$

which is the familiar Picard iteration.

Example. Let $\boldsymbol{M}(t)$ be the diagonal part of $\boldsymbol{A}(t)$. Then (4) becomes the Jacobi WR algorithm in which we solve the following IVP at each iteration k for each $x_i^{k+1}(t)$:

$$\begin{aligned} \left(\tfrac{d}{dt} + a_{ii}(t)\right) x_i^{k+1}(t) + \sum_{j \neq i} a_{ij}(t) x_j^k(t) &= b_i(t) \\ x_i(0) &= x_{0i}. \end{aligned}$$

As $\mathcal{K}$ has zero spectral radius, the a straightforward convergence result can be stated.

Theorem 2.2. Under the assumptions of Lemma 2.1, the method of successive approximations, defined in (4), converges.

A more detailed analysis of convergence can be derived by considering cases for which $\mathcal{K}$ is defined as $T \to \infty$, in which case $\mathcal{K}$ has nonzero spectral radius [13].

2.2. The Galerkin Method. Another approach to solving (3) is to apply a Galerkin method to solving a variational formulation of the problem. This approach leads directly to the conjugate-direction methods. Galerkin methods have been well studied for second-kind Fredholm integral equations [1, 7], of which second-kind Volterra equations are a special case, but infrequently studied for second-kind Volterra equations in particular (see, however, [11]). With the conjugate-direction approach, instead of applying the Galerkin method over a space of polynomials or splines, as is typical, one applies the Galerkin method over a Krylov space generated by $(\boldsymbol{I} - \mathcal{K})$. The use of a Galerkin method over a Krylov space generated by $(\boldsymbol{I} - \mathcal{K})$ is discussed in [14] and [16] where the approach is called the method of moments (see also [23]).

Let $\mathbf{X}$ and $\mathbf{Y}$ be Hilbert spaces and consider the operator equation

$$\mathcal{A}\boldsymbol{x} = \boldsymbol{b} \tag{5}$$

where $\boldsymbol{x} \in \mathbf{X}, \boldsymbol{b} \in \mathbf{Y}$ and $\mathcal{A} : \mathbf{X} \to \mathbf{Y}$ is a bounded injective operator.

By a Galerkin method, we mean any scheme by which the solution $\boldsymbol{x}$ in (5) is computed by solving the problem in a sequence of finite-dimensional subspaces via the use of orthogonal projections. That is, we take the subspaces $\mathbf{X}^n \subset \mathbf{X}$ and $\mathbf{Y}^n \subset \mathbf{Y}$ with $\dim \mathbf{X}^n = \dim \mathbf{Y}^n = n$ and require the Galerkin approximation $\boldsymbol{x}^n$ to satisfy

$$\langle \boldsymbol{b} - \mathcal{A}\boldsymbol{x}^n, \boldsymbol{y} \rangle = 0 \quad \forall \boldsymbol{y} \in \mathbf{Y}^n. \tag{6}$$

In general, it is sufficient to satisfy (6) over some basis of $\mathbf{Y}^n$. That is, we define $\mathbf{X}^n = \text{span}\{\boldsymbol{u}^0, \boldsymbol{u}^1, \ldots, \boldsymbol{u}^{n-1}\}$ and $\mathbf{Y}^n = \text{span}\{\boldsymbol{v}^0, \boldsymbol{v}^1, \ldots, \boldsymbol{v}^{n-1}\}$, so that the solution $\boldsymbol{x}^n$ must satisfy

$$\langle \boldsymbol{b} - \mathcal{A}\boldsymbol{x}^n, \boldsymbol{v}^j \rangle = 0 \quad j = 0, 1, \ldots, n-1. \tag{7}$$

If we take $\boldsymbol{x}^n$ to be

$$\boldsymbol{x}^n = \sum_{i=0}^{n-1} \gamma^i \boldsymbol{u}^i$$

then (7) generates a linear system of equations for $\{\gamma^i\}$:

$$\langle \mathcal{A} \sum_{i=0}^{n-1} \gamma^i \boldsymbol{u}^i, \boldsymbol{v}^j \rangle = \langle \boldsymbol{b}, \boldsymbol{v}^j \rangle.$$

The particular Galerkin method in which $\mathbf{Y} = \mathbf{X}$ and $\mathbf{Y}^n = \mathbf{X}^n$ is often called the Bubnov-Galerkin method. If $\mathcal{A}$ is positive definite in addition to being bounded and injective, it is well known that the Bubnov-Galerkin method is convergent for (5) [15]. Furthermore, if $\mathcal{A}$ is self-adjoint, the Galerkin approximations can be computed iteratively with the conjugate-gradient method (appropriately extended from $\mathbf{R}^N$ to $\mathbf{X}$, of course) [7].

For our particular problem, the operator $(\boldsymbol{I} - \mathcal{K})$ is not self-adjoint, yet we still seek a conjugate-direction method appropriate for solving (3). Such methods can be derived by considering the Galerkin method where $\mathbf{Y} = \mathcal{A}(\mathbf{X})$ and $\mathbf{Y}^n = \mathcal{A}(\mathbf{X}^n)$. That is, we require $\boldsymbol{x}^n$ to satisfy

$$\langle \boldsymbol{b} - \mathcal{A}\boldsymbol{x}^n, \mathcal{A}\boldsymbol{u}^j \rangle = 0 \quad j = 0, 1, \ldots, n-1.$$

```
Algorithm 2.1 (WGMRES).

Set $r^0 = b - (I - \mathcal{K})x^0$, $\beta = \|r^0\|$, and $v^0 = r^0/\beta$
For $k = 1, 2, \ldots$ until $\langle r^k, r^k\rangle < \epsilon$,
    $h_{j,k} = \langle (I - \mathcal{K})v^j, v^k\rangle$, $i = 1, 2, \ldots, k$
    $\hat{v}^{k+1} = (I - \mathcal{K})v^k - \sum_{j=1}^k h_{j,k}v^j$
    $h_{k+1,k} = \|\hat{v}^{k+1}\|$
    $v^{k+1} = \hat{v}^{k+1}/h_{k+1,k}$
Set $x^k = x^0 + V^k y^k$
    Here, $y^k$ minimizes $\|\beta e_1 - \bar{H}^k y^k\|$ where
        $\bar{H}^k$ is the $(k+1) \times k$ matrix with nonzero entries $h_{i,j}$,
        $V^k = [v^1, \ldots, v^k]$, and
        $e_1 = [1, 0, \ldots, 0]^T$
```

We have the following convergence result for such Galerkin methods, and we refer the reader to [10] for the proof.

Theorem 2.3. Let $\mathbf{X}$ be a Hilbert space and let $\mathcal{A} : \mathbf{X} \to \mathbf{X}$ be a bounded bijective linear operator. Let $\mathbf{X}^n \subset \mathbf{X}$ be a finite-dimensional subspace with $\mathbf{X}^n \subset \mathbf{X}^{n+1}$ for all $n \in \mathbf{N}$. If x is in the closure of $\mathbf{S} = \cup_{n=1}^{\infty}\mathbf{X}^n$, then the Galerkin method for (5) is convergent. Moreover, there exists the estimate

$$\|x - x^n\| \leq C\|b - \mathcal{A}x^n\|$$

for some constant C depending only on $\mathcal{A}$.

Corollary 2.4. The Galerkin method described in Theorem 2.3 is convergent for $(I - \mathcal{K})x = \psi$ in the space $\mathbf{H}$, with finite-dimensional subspaces $\mathbf{H}^n = \{\psi, \mathcal{K}\psi, \ldots \mathcal{K}^{n-1}\psi\}$ for all $n \in \mathbf{N}$.

We again refer to [10] for the proof of the corollary. However, note that to show $x \in \operatorname{cl} \mathbf{S}$, we need only realize that

$$x = (I - \mathcal{K})^{-1}\psi = \sum_{j=0}^{\infty} \mathcal{K}^j \psi$$

where the Neumann series for $(I - \mathcal{K})^{-1}$ converges, since the spectral radius of $\mathcal{K}$ is zero.

2.3. Iterative Algorithms. Various iterative algorithms exist which can be used to implement the Galerkin method described in Corollary 2.4. For example, the generalized minimum residual algorithm (GMRES) [18] can be adapted quite readily to the space $\mathbf{H}$ instead of $\mathbf{R}^N$. The resulting algorithm, WGMRES (waveform GMRES) is given in Algorithm 2.1.

The two fundamental operations in Algorithm 2.1 are the operator-function product, $(I - \mathcal{K})p$, and the inner product, $\langle \cdot, \cdot \rangle$. When solving (3) in the space $\mathbf{H}$, these operations are as follows:

Operator-Function Product: To calculate $w \equiv (I - \mathcal{K})p$:

1. Solve the IVP

$$\begin{aligned} (\tfrac{d}{dt} + M(t))y(t) &= N(t)p(t) \\ y(0) &= p_0 = 0 \end{aligned}$$

for $y(t)$, $t \in [0, T]$; this gives us $y = \mathcal{K}p$.

2. Set $w = p - y$

Inner Product: The inner product $\langle x, y\rangle$ is given by

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

```
Algorithm 3.1 (Nonlinear WGMRES).

Pick $x^0$, $\epsilon^0$, $\nu < 1$
For $m = 0, 1, \ldots$ until $\langle r^m, r^m \rangle < \phi$,
    Linearize (8) to form (9)
    Solve (9) with Algorithm 2.1 using $\epsilon^m$
    Update $x^{m+1}$ and $r^{m+1}$
    Set $\epsilon^{m+1} = \epsilon^m \cdot \nu$
```

Step 1 of the operator-function product is equivalent to one step of the classical dynamic iteration, hence WGMRES can be considered as a scheme for accelerating the convergence of dynamic iterations. This also implies that computing the operator-function product in the conjugate-direction based methods is as amenable to parallel implementation as classical dynamic iteration methods. Also, the inner products required by the WGMRES algorithm can be computed by N separate integrations of the pointwise product $x_i(t)y_i(t)$, which can be performed in parallel, followed by a global sum of the results.

3. Hybrid Methods for Nonlinear Systems. Many interesting applications are not necessarily described by a linear system of ODEs, but rather by a nonlinear system of ODEs:

$$(8) \qquad \begin{aligned} \tfrac{d}{dt}\boldsymbol{x}(t) + \boldsymbol{F}(\boldsymbol{x}(t), t) &= 0 \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0. \end{aligned}$$

To solve (8), we apply Newton's method directly to the nonlinear ODE system (in a process sometimes referred to as the waveform Newton method (WN) [19]) to obtain the following iteration:

$$(9) \qquad \begin{aligned} \left(\tfrac{d}{dt} + \boldsymbol{J}_F(\boldsymbol{x}^m)\right)\boldsymbol{x}^{m+1} &= \boldsymbol{J}_F(\boldsymbol{x}^m)\boldsymbol{x}^m - \boldsymbol{F}(\boldsymbol{x}^m) \\ \boldsymbol{x}^{m+1}(0) &= \boldsymbol{x}_0. \end{aligned}$$

Here, $\boldsymbol{J}_F$ is the Jacobian of $\boldsymbol{F}$. We note that (9) is a linear time-varying IVP to be solved for $\boldsymbol{x}^{m+1}$, which can be accomplished with a waveform conjugate direction method. The resulting operator Newton/conjugate-direction algorithm, a member of the class of hybrid Krylov methods [4], is given in Algorithm 3.1.

For the WGMRES algorithm applied to solving (9), the required operator-function product can be computed using the formulas in Section 2.3, with the substitution

$$\boldsymbol{M}(t) - \boldsymbol{N}(t) = \boldsymbol{J}_F(\boldsymbol{x}^m(t)).$$

It is also possible to use a Jacobian-free approach, but the nature of the linearization in the operator-Newton algorithm makes that approach somewhat unreliable [10].

Because of the preconditioning, the initial residual for the WGMRES algorithm must be computed, and this computation must be performed for every operator-Newton iteration. If the initial guess for $\boldsymbol{x}^{m+1}$ in the WGMRES part of the hybrid algorithm, denoted $\boldsymbol{x}^{m+1,0}$, is given by $\boldsymbol{x}^m$, then the initial residual for the WGMRES algorithm, denoted $\boldsymbol{r}^{m+1,0}$, can be computed using a two-step approach as follows:

1. Solve the IVP

$$\begin{aligned} \left(\tfrac{d}{dt} + \boldsymbol{M}(t)\right)\boldsymbol{y}(t) &= \boldsymbol{M}(t)\boldsymbol{x}^m(t) - \boldsymbol{F}(\boldsymbol{x}^m(t)) \\ \boldsymbol{y}(0) &= \boldsymbol{x}_0 \end{aligned}$$

for $\boldsymbol{y}(t)$, $t \in [0, T]$.

2. Set $\boldsymbol{r}^{m+1,0} = \boldsymbol{y} - \boldsymbol{x}^m$

4. Device Transient Simulation. A device is assumed to be governed by the Poisson equation, and the electron and hole continuity equations:

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

where u is the normalized electrostatic potential, n and p are the electron and hole concentrations, $\boldsymbol{J}_n$ and $\boldsymbol{J}_p$ 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 [2, 21].

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

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

where D_n and D_p are the diffusion coefficients, which are assumed here to be related to the electron and hole mobilities by the Einstein relations, that is $D = \frac{kT}{q}\mu$. $\boldsymbol{J}_n$ and $\boldsymbol{J}_p$ 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) [20]. On an N-node rectangular mesh, the spatial discretization yields a differential-algebraic system of $3N$ equations in $3N$ unknowns denoted by

$$\begin{aligned}
&(10) & \boldsymbol{f}_1(\boldsymbol{u}(t), \boldsymbol{n}(t), \boldsymbol{p}(t)) &= 0\\
&(11) & \boldsymbol{f}_2(\boldsymbol{u}(t), \boldsymbol{n}(t), \boldsymbol{p}(t)) &= \tfrac{d}{dt}\boldsymbol{n}(t)\\
&(12) & \boldsymbol{f}_3(\boldsymbol{u}(t), \boldsymbol{n}(t), \boldsymbol{p}(t)) &= \tfrac{d}{dt}\boldsymbol{p}(t)
\end{aligned}$$

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, $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3 : \mathbf{R}^{3N} \to \mathbf{R}^N$ are specified component-wise as

$$\begin{aligned}
f_{1_i}(u_i, n_i, p_i, u_j) &= \frac{\epsilon kT}{q}\sum_j \frac{d_{ij}}{L_{ij}}(u_i - u_j) - qA_i\,(p_i - n_i + N_D - N_A)\\
f_{2_i}(u_i, n_i, u_j, n_j) &= \frac{D_n}{A_i}\sum_j \frac{d_{ij}}{L_{ij}}\Big[n_j B(u_j - u_i) - n_i B(u_i - u_j)\Big] - R_i\\
f_{3_i}(u_i, p_i, u_j, p_j) &= \frac{D_p}{A_i}\sum_j \frac{d_{ij}}{L_{ij}}\Big[p_j B(u_i - u_j) - p_i B(u_j - u_i)\Big] - R_i.
\end{aligned}$$

The sums above are taken over the four nodes adjacent to node i (north, south, east, and west), L_{ij} is the 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, used to exponentially fit potential variation to electron concentration variation.

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 implicit 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 [12]. Another approach is to apply relaxation directly to the differential-algebraic equation system with a WR algorithm [8, 17], as given in Algorithm 4.1.

Algorithm 4.1 (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:
        $f_{1_i}(u_i^{k+1}, n_i^{k+1}, p_i^{k+1}, u_j^k) = 0$
        $f_{2_i}(u_i^{k+1}, n_i^{k+1}, u_j^k, n_j^k) = \frac{d}{dt} n_i^{k+1}$
        $f_{3_i}(u_i^{k+1}, p_i^{k+1}, u_j^k, p_j^k) = \frac{d}{dt} p_i^{k+1}$
```

In our approach, we apply the hybrid Krylov method described in Section 3 to solving (10)–(12). Therefore we use the WGMRES algorithm to solve the following IVP on each operator Newton iteration m:

$$\begin{aligned}
\begin{bmatrix} 0 \\ \frac{d}{dt}\boldsymbol{n}^{m+1} \\ \frac{d}{dt}\boldsymbol{p}^{m+1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_{f_{11}} & \boldsymbol{J}_{f_{12}} & \boldsymbol{J}_{f_{13}} \\ \boldsymbol{J}_{f_{21}} & \boldsymbol{J}_{f_{22}} & \boldsymbol{J}_{f_{23}} \\ \boldsymbol{J}_{f_{31}} & \boldsymbol{J}_{f_{32}} & \boldsymbol{J}_{f_{33}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{m+1} \\ \boldsymbol{n}^{m+1} \\ \boldsymbol{p}^{m+1} \end{bmatrix} & \\
= \begin{bmatrix} \boldsymbol{J}_{f_{11}} & \boldsymbol{J}_{f_{12}} & \boldsymbol{J}_{f_{13}} \\ \boldsymbol{J}_{f_{21}} & \boldsymbol{J}_{f_{22}} & \boldsymbol{J}_{f_{23}} \\ \boldsymbol{J}_{f_{31}} & \boldsymbol{J}_{f_{32}} & \boldsymbol{J}_{f_{33}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{m} \\ \boldsymbol{n}^{m} \\ \boldsymbol{p}^{m} \end{bmatrix} & - \begin{bmatrix} \boldsymbol{f}_1(\boldsymbol{u}^m, \boldsymbol{n}^m, \boldsymbol{p}^m) \\ \boldsymbol{f}_2(\boldsymbol{u}^m, \boldsymbol{n}^m, \boldsymbol{p}^m) \\ \boldsymbol{f}_3(\boldsymbol{u}^m, \boldsymbol{n}^m, \boldsymbol{p}^m) \end{bmatrix} \\
\begin{bmatrix} \boldsymbol{u}^{m+1}(0) \\ \boldsymbol{n}^{m+1}(0) \\ \boldsymbol{p}^{m+1}(0) \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{n}_0 \\ \boldsymbol{p}_0 \end{bmatrix} & .
\end{aligned}$$

4.1. Experimental Results. To apply the nonlinear WGMRES algorithm described in Section 3 to solving the system of differential-algebraic equations (10), (11) and (12), the algorithm was implemented using *WORDS* [17], a WR-based device transient simulation program. In addition, a waveform-relaxation-Newton algorithm (WRN), a waveform generalized-conjugate-residual (WGCR), and a waveform conjugate-gradient-squared algorithm (WCGS) were implemented for comparison purposes [22, 24]. It should be noted that the *WORDS* program uses a red/black vertical line Gauss-Seidel scheme, and that our conjugate-direction based implementations use the corresponding preconditioner.

Four N-channel MOSFET examples were used to compare the performance of the relaxation and conjugate-direction waveform methods:
kG: 2.2 μm channel-length; 50 psec, 0-5V ramp on the gate with the drain at 5V.
kD: 2.2 μm channel-length; 50 psec, 0-5V ramp on the drain with the gate at 5V.
jG: 0.17 μm channel-length; 5 psec, 0-1V ramp on the gate with the drain at 1V.
jD: 0.17 μm channel-length; 5 psec, 0-1V ramp on the drain with the gate 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. The graphs in Figures 1 and 2 compare the convergence of WR, WRN, WGCR, WGMRES, and WCGS for the **jD** and **kD** examples, respectively. In the graphs, the terminal current error versus number of function evaluations is plotted, and clearly demonstrates the rapid convergence of the conjugate-direction methods.

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 **jG** example, WCGS is 22 times faster than ordinary WR. Like the algebraic case, WGMRES and WGCR perform similarly in terms of function evaluations, but WGMRES is computationally more efficient because it avoids several waveform inner products on each iteration. Also like the algebraic case, WCGS performs very well on most problems, but can also exhibit convergence difficulty on others. Note that the CPU time reductions are not as large as the

TABLE 1

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

Example	Method	FEvals	CPU sec
jD	WR	8.43×10^6	14469
	WRN	3.77×10^6	7088
	WGCR	2.21×10^5	1138
	WGMRES	2.21×10^5	991
	WCGS	2.77×10^5	820
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
	WCGS	1.97×10^5	568
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
	WCGS	9.92×10^4	214
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
	WCGS	Non-Convergence	

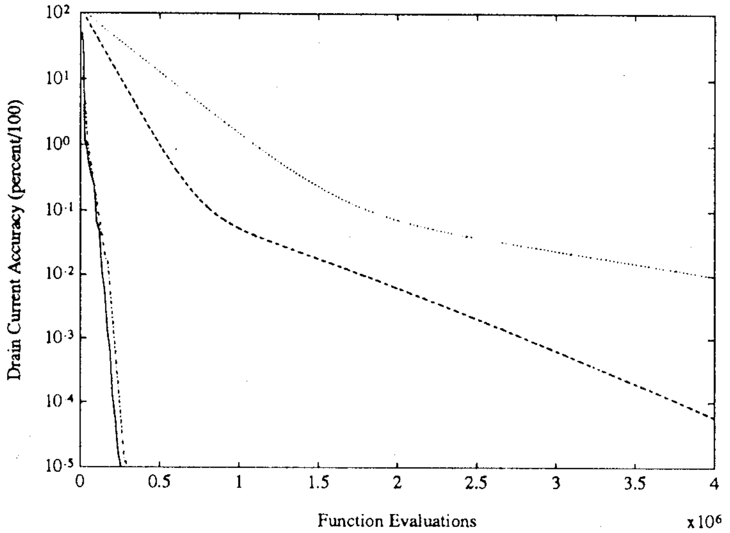

FIG. 1. *Convergence comparison between WR (dotted), WRN (dashed), WGCR/WGMRES (solid), and WCGS (dash-dotted) for* **jD** *example. The max-norm of the relative drain terminal current error is plotted against the number of function evaluations.*

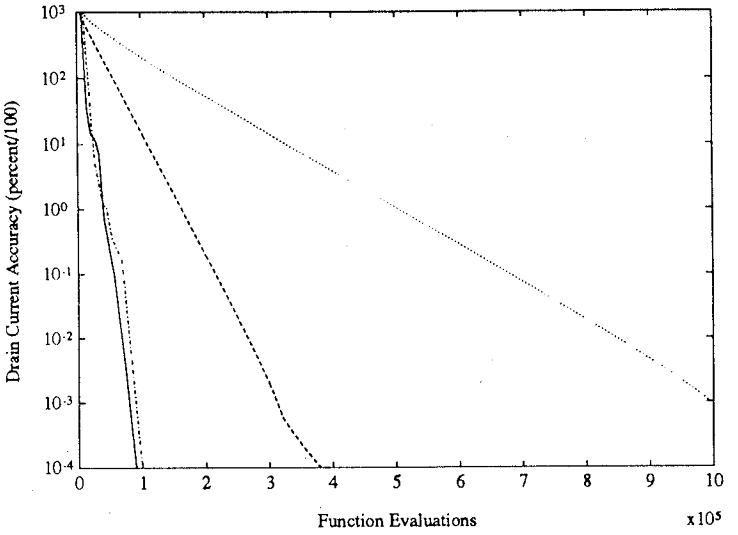

FIG. 2. *Convergence comparison between WR (dotted), WRN (dashed), WGCR/WGMRES (solid), and WCGS (dash-dotted) for* **kD** *example. The max-norm of the relative drain terminal current error is plotted against the number of function evaluations.*

function evaluation reduction, and this is partly due to the cost of inner product computations required for each iteration of the conjugate-direction methods. The difference is especially apparent with WGMRES and WGCR, because the number of inner products which must be computed on each iteration grows linearly with the number of iterations. On the other hand, WCGS requires constant work per iteration but can become unstable and fail to converge. For this reason, we are currently investigating generalizing the recently developed QMR algorithm [6].

5. Conclusion. In this paper we presented some new dynamic iterative methods to accelerate the convergence of the WR algorithm. The methods are based on the application of the Galerkin method to an operator equation formulation of the linear time-varying initial-value problem. Experimental results demonstrated that this acceleration significantly reduces the computation time for device transient simulation.

Future work is primarily focused on improving the theoretical results about the convergence of linear and nonlinear conjugate-direction methods for differential-algebraic systems of equations. In addition, the effect of using multirate integration must also be examined. Finally, we are investigating function-space generalizations of the QMR algorithm.

Acknowledgments. The authors would like to thank Ibrahim Elfadel at MIT, and F. Odeh at the IBM T. J. Watson research center for many valuable discussions. The authors also express their deep appreciation to Mark Reichelt for his implementation of, and help with, the *WORDS* program.

REFERENCES

[1] K. E. ATKINSON, *A Survey of Numerical Methods for the Solution of Fredholm Integral Equations of the Second Kind*, SIAM, Philadelphia, 1976.

[2] R. BANK, W. COUGHRAN, JR., W. FICHTNER, E. GROSSE, D. ROSE, AND R. SMITH, *Transient simulation of silicon devices and circuits*, IEEE Trans. CAD, 4 (1985), pp. 436–451.

[3] R. W. BROCKETT, *Finite Dimensional Linear Systems*, Wiley, New York, 1970.

[4] P. BROWN AND Y. SAAD, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.

[5] J. B. CONWAY, *A Course in Functional Analysis, Second Edition*, Springer-Verlag, New York, 1990.

[6] R. W. FREUND AND N. M. NACHTIGAL, *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Tech. Report 90.51, RIACS, NASA Ames Research Center, December 1990.

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

[8] 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, 1 (1982), pp. 131–145.

[9] P. Linz, *Analytical and Numerical Methods for Volterra Equations*, SIAM, Philadelphia, 1985.

[10] A. Lumsdaine, *Theoretical and Practical Aspects of Parallel Numerical Algorithms for Initial Value Problems, with Applications*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1992.

[11] R. C. MacCamy and P. Weiss, *Numerical solution of Volterra integral equations*, Nonlinear Anal., 3 (1979), pp. 677–695.

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

[13] U. Miekkala and O. Nevanlinna, *Convergence of dynamic iteration methods for initial value problems*, SIAM J. Sci. Stat. Comp., 8 (1987), pp. 459–467.

[14] G. Miel, *Iterative refinement of the method of moments*, Numer. Funct. Anal. and Optimiz., 9(11-12) (1987–1988), pp. 1193–1200.

[15] S. G. Mikhlin, *Variational Methods in Mathematical Physics*, Macmillan, New York, 1964.

[16] P. Omari, *On the fast convergence of a Galerkin-like method for equations of the second kind*, Math. Z., 201 (1989), pp. 529–539.

[17] 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, November 1989, pp. 412–415.

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

[19] R. Saleh and J. White, *Accelerating relaxation algorithms for circuit simulation using waveform-newton and step-size refinement*, IEEE Trans. CAD, 9 (1990), pp. 951–958.

[20] D. Scharfetter and H. Gummel, *Large-signal analysis of a silicon read diode oscillator*, IEEE Transactions on Electron Devices, ED-16 (1969), pp. 64–77.

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

[22] P. Sonneveld, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 36–52.

[23] Y. V. Vorobyev, *Method of Moments in Applied Mathematics*, Gordon and Breach, New York, 1965.

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