To appear in Numerical Functional
Analysis and Optimization

ACCELERATING DYNAMIC ITERATION METHODS WITH APPLICATION TO PARALLEL 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 waveform relaxation, or dynamic iteration, methods for solving linear time-varying initial-value problems. It is shown that the Galerkin approximations can be computed iteratively using Krylov-subspace 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 to demonstrate that waveform Krylov-subspace methods converge significantly faster than classical waveform relaxation, and are better able to exploit the parallelism available in loosely coupled parallel machines than parallel versions of standard point-wise iterative schemes.

Key words. Krylov-subspace methods, dynamic iteration, Galerkin method, waveform relaxation.

AMS subject classifications. 65L60, 65L05, 65R20, 65J10

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

$$\begin{aligned} (\tfrac{d}{dt} + \boldsymbol{A}(t))\boldsymbol{x}(t) &= \boldsymbol{b}(t) \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0, \end{aligned} \tag{1.1}$$

where $\boldsymbol{A}(t) \in \mathbb{R}^{N\times N}$, $\boldsymbol{b}(t) \in \mathbb{R}^N$ is a given right-hand side, and $\boldsymbol{x}(t) \in \mathbb{R}^N$ 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 (1.1) 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 *pointwise* 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 differential equation 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.

Since the WR algorithm was first introduced as an efficient technique for solving the large sparsely-coupled differential equation systems generated by simulation of integrated circuits [10], its properties have been under substantial theoretical and practical investigation. The precise

* Submitted for publication. This work was supported by a grant from IBM, the Defense Advanced Research Projects Agency contract N00014-91-J-1698, and National Science Foundation grant CCR92-09815.

† 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)

nature of the loose-coupling in integrated circuits, which was responsible for WR's rapid convergence for those examples, was first made clear in [22]. The more formal theory for WR applied to linear time-invariant systems in normal form is described in [16], and theoretical aspects which arise when WR is applied to the more general form $(\boldsymbol{C}\frac{d}{dt} + \boldsymbol{A})\boldsymbol{x}(t) = \boldsymbol{b}(t)$ are examined in [21]. Since the WR method decomposes before time-discretization, it has been used as a tool for examining the stability properties of multirate integration methods [35]. Though WR's major practical success has been in accelerating the simulation of integrated circuits [36, 23], it has also been examined for the certain specific problems. For example, the effects of time interval selection has been examined for RC circuit problems [9], and the method has been applied to semiconductor device simulation [27].

As the above body of work makes clear, for WR to be a computational competitor to pointwise methods, its convergence must be accelerated. Approaches to accelerating the convergence of WR include multigrid [12, 33], SOR [16], convolution SOR [25], Krylov-subspace methods [13], and adaptive window size selection [8]. In this paper, we describe primarily practical aspects of using Krylov-subspace techniques to accelerate WR convergence.

In the next section, we begin by describing the system of second-kind Volterra integral equations obtained by applying a "dynamic preconditioner" to (1.1). 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 Krylov-subspace 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 briefly describe how to apply the hybrid WR and WGMRES algorithms to solving the time-dependent drift-diffusion equations used to describe transient phenomena in semiconductors, and experiment results on serial and parallel computers are given in Section 5. Finally, our conclusions and suggestions for future work are contained in Section 6.

2. Waveform Krylov-Subspace Methods. In (1.1), let $\boldsymbol{A}(t) = \boldsymbol{M}(t) - \boldsymbol{N}(t)$ be a splitting of $\boldsymbol{A}(t)$. The waveform relaxation algorithm based on this splitting is expressed as

Algorithm 2.1 (Waveform Relaxation for Linear Systems).

1. *Initialize:* Pick $\boldsymbol{x}^0$
2. *Iterate:* For $k = 0, 1, \ldots$

$$\text{Solve } (\tfrac{d}{dt} + \boldsymbol{M})\boldsymbol{x}^{k+1} = \boldsymbol{N}\boldsymbol{x}^k + \boldsymbol{f}$$
$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

for $\boldsymbol{x}^{k+1}$ on $[0, T]$.

The solution $\boldsymbol{x}$ to (1.1) is thus a fixed point of the WR algorithm, satisfying the Volterra integral operator equation

$$(\boldsymbol{I} - \mathcal{K})\boldsymbol{x} = \boldsymbol{\psi}. \tag{2.1}$$

Here, (2.1) is defined on the space $\mathbb{H} = \mathbb{L}_2([0, T], \mathbb{R}^N)$, $\boldsymbol{I} : \mathbb{H} \to \mathbb{H}$ is the identity operator,

$\mathcal{K} : \mathbb{H} \to \mathbb{H}$ is defined by

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

$\boldsymbol{\psi} \in \mathbb{H}$ is given by

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

and $\boldsymbol{\Phi}_M$ is the state transition matrix [3] associated with $\boldsymbol{M}(t)$.

The following are standard results (see, e.g., [5, 7]) which will be used in subsequent discussions of (2.1).

Lemma 2.1. If $\boldsymbol{M}$ and $\boldsymbol{N}$ are piecewise continuous with respect to t, then $\mathcal{K} : \mathbb{H} \to \mathbb{H}$ 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 Krylov-subspace 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 problem (2.1):

$$\boldsymbol{x}^{k+1} = \mathcal{K}\boldsymbol{x}^k + \boldsymbol{\psi}. \tag{2.2}$$

This approach is known as the method of successive approximations, waveform relaxation, or the Picard-Lindelöf iteration [1, 7, 11, 16, 37].

Example. Let $\boldsymbol{M}(t)$ be the diagonal part of $\boldsymbol{A}(t)$. Then (2.2) becomes the Jacobi WR algorithm in which we solve the following IVP at each iteration k for each $x_i^{k+1}(t)$:

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

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

Theorem 2.2. Under the assumptions of Lemma 2.1, the method of successive approximations, defined in (2.2), 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 [16].

2.2. The Galerkin Method. Another approach to solving (2.1) is to apply a Galerkin method to solving a variational formulation of the problem. This approach leads directly to the Krylov-subspace 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, [14]). With the Krylov-subspace 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 [17] and [24] where the approach is called the method of moments (see also [34]).

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

$$\mathcal{A}\boldsymbol{x} = \boldsymbol{b} \tag{2.3}$$

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

By a Galerkin method, we mean any scheme by which the solution $\boldsymbol{x}$ in (2.3) 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 $\mathbb{X}^n \subset \mathbb{X}$ and $\mathbb{Y}^n \subset \mathbb{Y}$ with $\dim \mathbb{X}^n = \dim \mathbb{Y}^n = n$ and require the Galerkin approximation $\boldsymbol{x}^n$ to satisfy

$$\langle \boldsymbol{b} - \mathcal{A}\boldsymbol{x}^n, \boldsymbol{y} \rangle = 0 \quad \forall \boldsymbol{y} \in \mathbb{Y}^n. \tag{2.4}$$

In general, it is sufficient to satisfy (2.4) over some basis of $\mathbb{Y}^n$. That is, we define $\mathbb{X}^n = \text{span}\{\boldsymbol{u}^0, \boldsymbol{u}^1, \ldots, \boldsymbol{u}^{n-1}\}$ and $\mathbb{Y}^n = \text{span}\{\boldsymbol{v}^0, \boldsymbol{v}^1, \ldots, \boldsymbol{v}^{n-1}\}$, so that the solution $\boldsymbol{x}^n$ must satisfy

$$\langle \boldsymbol{b} - \mathcal{A}\boldsymbol{x}^n, \boldsymbol{v}^j \rangle = 0 \quad j = 0, 1, \ldots, n-1. \tag{2.5}$$

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

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

then (2.5) 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 $\mathbb{Y} = \mathbb{X}$ and $\mathbb{Y}^n = \mathbb{X}^n$ 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 (2.3) [18]. Furthermore, if $\mathcal{A}$ is self-adjoint, the Galerkin approximations can be computed iteratively with the conjugate-gradient method (appropriately extended from $\mathbb{R}^N$ to $\mathbb{X}$, of course) [7].

For our particular problem, the operator $(\boldsymbol{I} - \mathcal{K})$ is not self-adjoint, yet we still seek a Krylov-subspace method appropriate for solving (2.1). Such methods can be derived by considering the Galerkin method where $\mathbb{Y} = \mathcal{A}(\mathbb{X})$ and $\mathbb{Y}^n = \mathcal{A}(\mathbb{X}^n)$. That is, we require $\boldsymbol{x}^n$ to satisfy

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

We have the following convergence result for such Galerkin methods, and we refer the reader to [13] for the proof.

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

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

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

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

We again refer to [13] for the proof of the corollary. However, note that to show $\boldsymbol{x} \in \operatorname{cl} \mathbb{S}$, we need only realize that

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

where the Neumann series for $(\boldsymbol{I} - \mathcal{K})^{-1}$ 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) [28] can be adapted quite readily to the space $\mathbb{H}$ instead of $\mathbb{R}^N$.

Algorithm 2.2 (Waveform GMRES).

1. *Start:* Set $\boldsymbol{r}^0 = \boldsymbol{\psi} - (\boldsymbol{I} - \mathcal{K})\boldsymbol{x}^0$, $\boldsymbol{v}^1 = \boldsymbol{r}^0/\|\boldsymbol{r}^0\|$
2. *Iterate:* For $k = 1, 2, \ldots$, until satisfied do:
   - $h_{j,k} = \langle (\boldsymbol{I} - \mathcal{K})\boldsymbol{v}^k, \boldsymbol{v}^j \rangle$, $j = 1, 2, \ldots, k$
   - $\hat{\boldsymbol{v}}^{k+1} = (\boldsymbol{I} - \mathcal{K})\boldsymbol{v}^k - \sum_{j=1}^{k} h_{j,k}\boldsymbol{v}^j$
   - $h_{k+1,k} = \|\hat{\boldsymbol{v}}^{k+1}\|$
   - $\boldsymbol{v}^{k+1} = \hat{\boldsymbol{v}}^{k+1}/h_{k+1,k}$
3. *Form approximate solution:*
   - $\boldsymbol{x}^k = \boldsymbol{x}^0 + \boldsymbol{V}^k\boldsymbol{y}^k$, where $\boldsymbol{y}^k$ minimizes $\|\beta\boldsymbol{e}_1 - \bar{\boldsymbol{H}}^k\boldsymbol{y}^k\|$

The two fundamental operations in Algorithm 2.2 are the operator-function product, $(\boldsymbol{I} - \mathcal{K})\boldsymbol{p}$, and the inner product, $\langle \cdot, \cdot \rangle$. When solving (2.1) in the space $\mathbb{H}$, these operations are as follows:

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

1. Solve the IVP

$$\begin{aligned} (\tfrac{d}{dt} + \boldsymbol{M}(t))\boldsymbol{y}(t) &= \boldsymbol{N}(t)\boldsymbol{p}(t) \\ \boldsymbol{y}(0) &= \boldsymbol{p}_0 = 0 \end{aligned}$$

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

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

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

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 Krylov-subspace 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. Consider the problem of numerically solving the nonlinear IVP:

$$\begin{aligned} \tfrac{d}{dt}\boldsymbol{x}(t) + \boldsymbol{F}(\boldsymbol{x}(t), t) &= 0 \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0. \end{aligned} \tag{3.1}$$

To solve (3.1), we apply Newton's method directly to the nonlinear ODE system (in a process sometimes referred to as the waveform Newton method (WN) [29]) to obtain the following iteration:

$$\begin{aligned} \left(\tfrac{d}{dt} + \boldsymbol{J}_F(\boldsymbol{x}^m)\right) \boldsymbol{x}^{m+1} &= \boldsymbol{J}_F(\boldsymbol{x}^m)\boldsymbol{x}^m - \boldsymbol{F}(\boldsymbol{x}^m) \\ \boldsymbol{x}^{m+1}(0) &= \boldsymbol{x}_0. \end{aligned} \tag{3.2}$$

Here, $\boldsymbol{J}_F$ is the Jacobian of $\boldsymbol{F}$. We note that (3.2) is a linear time-varying IVP to be solved for $\boldsymbol{x}^{m+1}$, which can be accomplished with a waveform Krylov-subspace method. The resulting operator Newton/Krylov-subspace algorithm, a member of the class of hybrid Krylov methods [4], is shown below.

Algorithm 3.1 (Waveform Newton/WGMRES).

1. *Initialize:* Pick $\boldsymbol{x}^0$
2. *Iterate:* For $m = 0, 1, \ldots$ until converged
   - Linearize (3.1) to form (3.2)
   - Solve (3.2) with WGMRES
   - Update $\boldsymbol{x}^{m+1}$

For the WGMRES algorithm applied to solving (3.2), 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 [13].

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

$$(\tfrac{d}{dt} + \boldsymbol{M}(t))\boldsymbol{y}(t) = \boldsymbol{M}(t)\boldsymbol{x}^m(t) - \boldsymbol{F}(\boldsymbol{x}^m(t))$$
$$\boldsymbol{y}(0) = \boldsymbol{x}_0$$

   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:

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

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, 31].

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

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

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) [30]. On an N-node rectangular mesh, the spatial discretization yields a differential-algebraic system of $3N$ equations in $3N$ unknowns denoted by

$$\boldsymbol{f}_1(\boldsymbol{u}(t), \boldsymbol{n}(t), \boldsymbol{p}(t)) = 0 \tag{4.1}$$
$$\boldsymbol{f}_2(\boldsymbol{u}(t), \boldsymbol{n}(t), \boldsymbol{p}(t)) = \tfrac{d}{dt}\boldsymbol{n}(t) \tag{4.2}$$
$$\boldsymbol{f}_3(\boldsymbol{u}(t), \boldsymbol{n}(t), \boldsymbol{p}(t)) = \tfrac{d}{dt}\boldsymbol{p}(t) \tag{4.3}$$

where $t \in [0, T]$, and $\boldsymbol{u}(t), \boldsymbol{n}(t), \boldsymbol{p}(t) \in \mathbb{R}^N$ are vectors of normalized potential, electron concentration, and hole concentration, respectively. Here, $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3 : \mathbb{R}^{3N} \to \mathbb{R}^N$ 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 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 [15]. Another approach is to apply relaxation directly to the differential-algebraic equation system with a WR algorithm [10, 26].

Algorithm 4.1 (WR for Device Simulation).

1. *Initialize:* Guess $\boldsymbol{u}^0, \boldsymbol{n}^0, \boldsymbol{p}^0$ waveforms at all nodes
2. *Iterate:* 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:

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

In our approach, we apply the hybrid Krylov method described in Section 3 to solving (4.1)–(4.3). Therefore we use the WGMRES algorithm to solve the following IVP on each operator Newton iteration m:

$$\begin{bmatrix} 0 \\ \frac{d}{dt}\boldsymbol{n}^{m+1} \\ \frac{d}{dt}\boldsymbol{p}^{m+1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_{f_{11}} & \boldsymbol{J}_{f_{12}} & \boldsymbol{J}_{f_{13}} \\ \boldsymbol{J}_{f_{21}} & \boldsymbol{J}_{f_{22}} & \boldsymbol{J}_{f_{23}} \\ \boldsymbol{J}_{f_{31}} & \boldsymbol{J}_{f_{32}} & \boldsymbol{J}_{f_{33}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{m+1} \\ \boldsymbol{n}^{m+1} \\ \boldsymbol{p}^{m+1} \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{J}_{f_{11}} & \boldsymbol{J}_{f_{12}} & \boldsymbol{J}_{f_{13}} \\ \boldsymbol{J}_{f_{21}} & \boldsymbol{J}_{f_{22}} & \boldsymbol{J}_{f_{23}} \\ \boldsymbol{J}_{f_{31}} & \boldsymbol{J}_{f_{32}} & \boldsymbol{J}_{f_{33}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{m} \\ \boldsymbol{n}^{m} \\ \boldsymbol{p}^{m} \end{bmatrix} - \begin{bmatrix} \boldsymbol{f}_1(\boldsymbol{u}^m, \boldsymbol{n}^m, \boldsymbol{p}^m) \\ \boldsymbol{f}_2(\boldsymbol{u}^m, \boldsymbol{n}^m, \boldsymbol{p}^m) \\ \boldsymbol{f}_3(\boldsymbol{u}^m, \boldsymbol{n}^m, \boldsymbol{p}^m) \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{u}^{m+1}(0) \\ \boldsymbol{n}^{m+1}(0) \\ \boldsymbol{p}^{m+1}(0) \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{n}_0 \\ \boldsymbol{p}_0 \end{bmatrix}.$$

5. Experimental Results. Numerical experiments were conducted to compare the performance of classical waveform relaxation methods with Krylov-subspace methods. WR,

TABLE 5.1
Comparison of WR, WRN, WGMRES, and WCGS. CPU times shown are for an IBM RS/6000 model 540.

Example	Method	FEvals	CPU sec
kD	WR	1.22×10^6	1526
	WRN	3.94×10^5	559
	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
	WGMRES	1.03×10^5	316
	WCGS	Non-Convergence	

WN/WGMRES (Algorithm 3.1), and WN/WCGS [32] were implemented using the WR based device simulation programs WORDS [26] and a parallel variant, pWORDS. In addition, the waveform-relaxation-Newton (WRN) algorithm [37] was also implemented in the WORDS and pWORDS programs. The WORDS program uses a red/black vertical line Gauss-Seidel scheme, and our Krylov-subspace implementations use the corresponding preconditioner.

5.1. Serial Results. For performance comparison on a serial computer, experiments were conducted using a two-dimensional n-channel MOS transistor model discretized with a 19×31 mesh. Two examples were used to compare the performance of the relaxation and Krylov-subspace 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.

The parameters used with the Krylov-subspace 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 5.1 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. Figure 5.1 compares the convergence of WR, WRN, WGMRES, and WCGS for the **kD** example. 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 5.1 indicates, Krylov-subspace methods significantly reduced the number of function evaluations and CPU time over WR and WRN. In a manner analogous to the algebraic case, WN/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 function evaluation reduction, and this is partly due to the cost of inner product computations required for each iteration of the Krylov-subspace methods. The difference is especially apparent with WGMRES, 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].

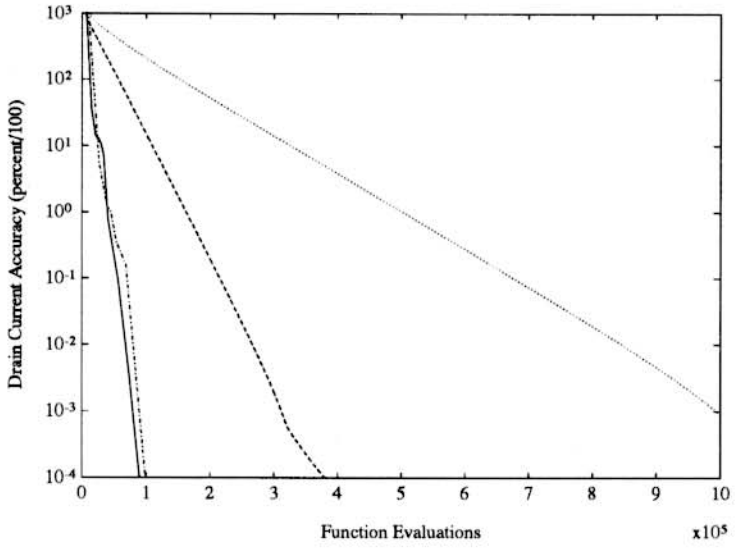

FIG. 5.1. *Convergence comparison between WR (dotted), WRN (dashed), WGMRES (solid), and WCGS (dash-dotted) for* **kD** *example. The max-norm of the relative drain terminal current error is plotted against the number of function evaluations.*

5.2. Example Analysis. It was suggested in [20] that the Krylov-subspace methods will not converge significantly faster than WR methods, because the associated operator has a continuous spectrum with substantial volume in the complex plane. However, the above experimental results are not consistent with such a conclusion. To try to reconcile this inconsistency, we will analyze a specific example problem, the discretized heat equation, using techniques described in [19, 25].

Consider the finite-difference discretized one-dimensional heat equation with Dirichlet boundary conditions,

$$\frac{d}{dt}\boldsymbol{x}(t) + (\boldsymbol{I} - \boldsymbol{N})\boldsymbol{x}(t) = 0 \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{5.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^N$, $\boldsymbol{I}, \boldsymbol{N} \in \mathbb{R}^{N\times N}$, and the only nonzero entries in $\boldsymbol{N}$ are $\boldsymbol{N}_{i,i+1} = 0.5$, for $i \in \{1, ..., N-1\}$ and $\boldsymbol{N}_{i,i-1} = 0.5$, for $i \in \{2, ..., N\}$. Applying a backward-Euler time discretization yields the discrete-time equation

$$\frac{1}{h}(\boldsymbol{x}[j] - \boldsymbol{x}[j-1]) + (\boldsymbol{I} - \boldsymbol{N})\boldsymbol{x}[j] = 0. \tag{5.2}$$

where h is the discretization timestep. Solving (5.2) with a discrete-time WR algorithm results in the discrete-time iteration equation

$$\frac{1}{h}(\boldsymbol{y}^{k+1}[j] - \boldsymbol{y}^{k+1}[j-1]) + \boldsymbol{y}^{k+1}[j] - \boldsymbol{N}\boldsymbol{y}^{k}[j] = 0. \tag{5.3}$$

where k is the waveform iteration index, $\boldsymbol{y}^{k+1}[j] \equiv \boldsymbol{x}^{k+1}[j] - \boldsymbol{x}^{k}[j]$, and therefore $\boldsymbol{y}^{k}[0] = 0$.

When considering (5.3) on the semi-infinite interval, that is for all integers $j > 0$, the

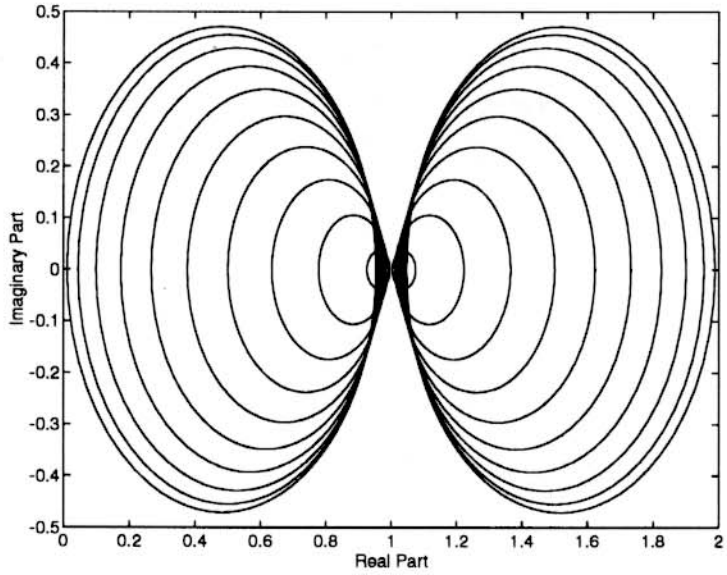

FIG. 5.2. *The spectrum of $\boldsymbol{T}$ in (5.4) for the case of $N = 20$ and $h = 0.1$. Note, the spectrum is the union of the regions bounded by the depicted circles.*

waveform iterates satisfy

$$\begin{bmatrix} \boldsymbol{y}^{k+1}[1] \\ \boldsymbol{y}^{k+1}[2] \\ \boldsymbol{y}^{k+1}[3] \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} \boldsymbol{y}^{k}[1] \\ \boldsymbol{y}^{k}[2] \\ \boldsymbol{y}^{k}[3] \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}, \tag{5.4}$$

where $\boldsymbol{T}$ is the inverse of a semi-infinite block Toeplitz matrix, and $\boldsymbol{T}$'s symbol is given by

$$\chi(z) = \frac{1}{\frac{1-z^{-1}}{h} + 1} \boldsymbol{N}.$$

Alternatively, $\chi(z)$ can be derived by computing the z-transform of (5.3). The spectrum of $\boldsymbol{T}$, $\lambda(\boldsymbol{T})$, is then given by [38]

$$\lambda(T) = \{\chi(z) \mid |z| \leq 1\}.$$

Now consider the specific example where $n = 20$. In Figure 5.2, the spectrum of $\boldsymbol{T}$ for $h = 0.1$ is given. In this case, the timestep is significantly smaller than the time constant associated with the fastest mode of (5.1). Note that $\lambda(T)$ covers a significant fraction of a disc of radius two centered at one. This implies that relaxation, whose associated iteration polynomial has all its zeros at precisely one, is reasonably close to optimal. Therefore, Krylov-subspace based approaches will be of limited additional advantage. This is demonstrated experimentally in Figure 5.3, where the convergence rates of WR and WGMRES are compared.

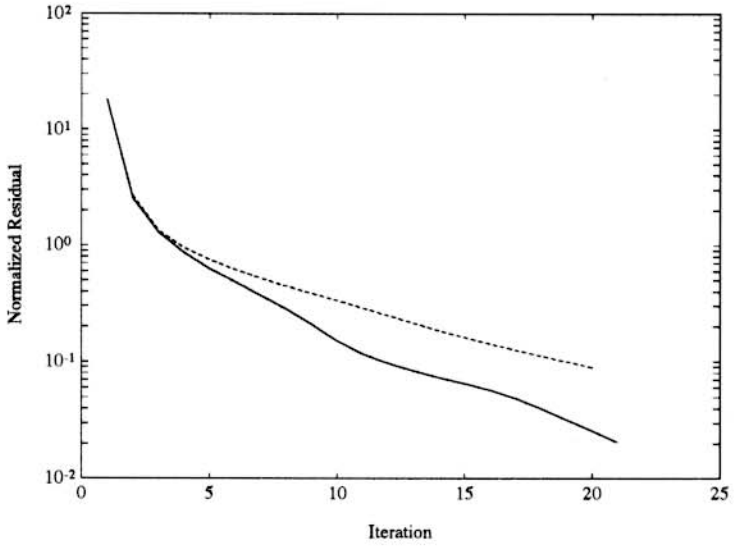

FIG. 5.3. *The normalized residuals as a function of iteration for WR (dashed line) and WGMRES (solid line) applied to solving (5.3) with* $N = 20$ *and* $h = 0.1$. *The waveforms were computed using* 500 *timesteps.*

Systems like the spatially discretized heat equation are stiff, with both rapidly and very slowly decaying modes. If a stiffly-stable time-integration formula like backward-Euler is used to solve (5.1), then most of the timesteps will be selected to accurately capture the slowest modes. In particular, for the discretized heat equation example with $N = 20$, a more practical case to analyze is when $h = 10$, rather than the $h = 0.1$ case above, as this larger timestep will still insure the slower modes are accurately computed. In Figure 5.4, the spectrum of $\boldsymbol{T}$ for $h = 10$ is given. As is clear from the figure, the spectrum of $\boldsymbol{T}$ tightly hugs the real axis, much more so than in the $h = 0.1$ case. Since this spectrum covers a small fraction of the radius two disc centered at one, Krylov-subspace based approaches should have a significant advantage over WR. That is is the case is made clear in the comparisons in Figure 5.5.

5.3. Parallel Results. Parallel numerical experiments were conducted to compare the practical efficiency of the waveform methods with the best known serial methods (as well as with parallel versions of the best known serial methods). The experiments were conducted using a two-dimensional n-channel MOS transistor model discretized with a 19×31 mesh. The experiments simulated the effect of a 5 volt pulse applied to the drain terminal with the gate terminal held at a constant 5 volts.

The experimental parallel computing environment consisted of eight IBM RS/6000 workstations — five model 320s, one model 320H, and two model 540s — and one Sun SparcStation 2. The Sun SparcStation 2 was used as the Master for all experiments and the IBM RS/6000 machines were used as the Slaves. To make the parallel results as meaningful as possible, serial results were obtained on a single model 320, results with two and four processors were obtained on two and four model 320 Slaves, respectively, and results with eight processors were obtained with all eight machines. The mesh was divided as evenly as possible among the Slave processors — no load balancing was attempted.

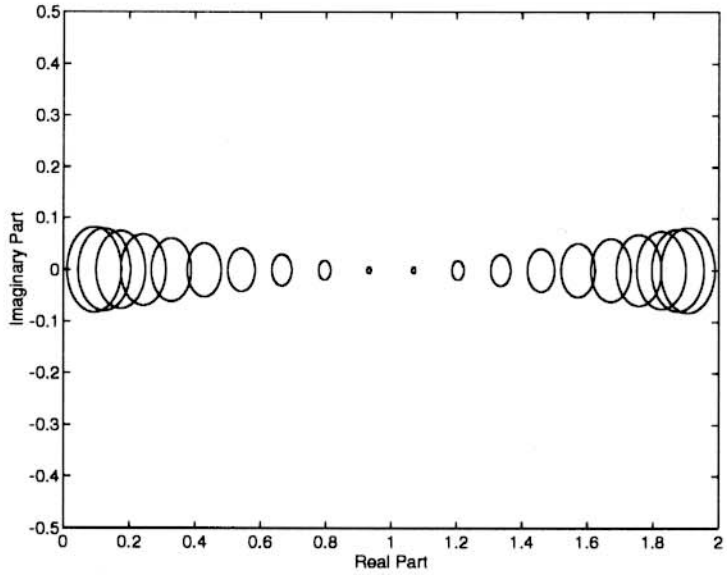

FIG. 5.4. *The Spectrum of* $\boldsymbol{T}$ *in (5.4) for the case of* $N = 20$ *and* $h = 10$. *Note, the spectrum is the union of the regions bounded by the depicted circles.*

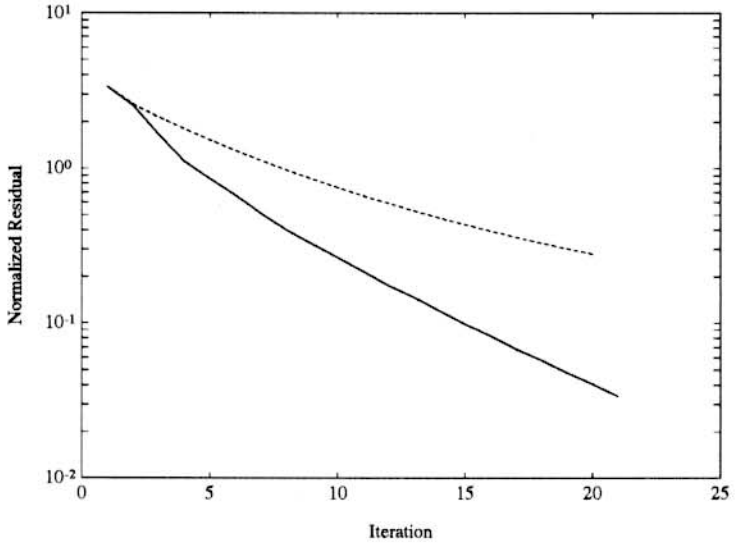

FIG. 5.5. *The normalized residuals as a function of iteration for WR (dashed line) and WGMRES (solid line) applied to solving (5.3) with* $N = 20$ *and* $h = 10$. *The waveforms were computed using* 500 *timesteps.*

Table 5.2 shows a comparison of the execution times (measured in elapsed wall clock seconds) required to complete a transient simulation of the test device using WRN and WN/WGMRES. For all experiments, first order BDF and 256 fixed timesteps were used over a simulation interval of 51.2×10^{-11} seconds. To establish a uniform measure for purposes of comparison, the convergence criterion for all experiments was the requirement that the maximum error over the simulation interval in the value of any terminal current be less than one part in 1×10^{-4}. To provide an initial guess for WRN and for WN/WGMRES, 16 and 8 initial WR iterations were

Method	# Procs	Time
WRN	1	8230.23
WRN	2	4469.91
WRN	4	2712.58
WRN	8	1571.92
WN/WGMRES	1	*
WN/WGMRES	2	*
WN/WGMRES	4	925.60
WN/WGMRES	8	504.50
Pointwise (Direct)	1	2462.48
Pointwise (GMRES)	1	1221.98
Pointwise (GMRES)	2	6931.86

TABLE 5.2

*Execution times (measured in elapsed wall clock seconds) required to complete a transient simulation of the test device using WRN, WN/WGMRES, and point at a time methods. A * indicates that the experiment was not able to be run because of memory restrictions.*

performed, respectively, after which WRN and WN/WGMRES required 499 and 75 iterations to converge, respectively.

In addition, Table 5.2 shows the execution times required to perform traditional point at a time simulation of the test device, using direct and vertical-line preconditioned GMRES (PGMRES) linear system solvers. Parallel runs with the PGMRES point at a time method were conducted, but as is shown in the table, execution time *increased* — a result of the large number of communication and synchronization steps required by PGMRES at each timestep and the high latency of PVM and standard Ethernet communication. Note that we did not try to parallelize direct factorization.

As can be seen from the table, the WN/WGMRES method has very good parallel performance (although because of its large memory requirements, it could not be accommodated by the smaller model 320 machines for runs on just one or two machines). Because it is necessarily more synchronous, WN/WGMRES might appear to be at a disadvantage (when compared to WR or WRN) in a parallel implementation, however its vastly superior convergence rate makes it the clear overall winner.

6. 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 hybrid Krylov-subspace 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 and Mark Reichelt for many valuable discussions. In addition, the authors would like to acknowledge F. Odeh for his

guidance in this subject, he will be remembered for much more than the papers which bear his name. F. Odeh has influenced most of the researchers in the field of waveform relaxation. Almost all of us either knew the man, or were students of researchers whose careers F. Odeh helped shape.

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] B. LEIMKUHLER, *Estimating waveform relaxation convergence*, SIAM J. Sci. Comput., 14 (1993), pp. 872–889.

[9] B. LEIMKUHLER AND A. RUEHLI, *Rapid convergence of waveform relaxation*, Applied Numerical Mathematics, 11 (1993), pp. 221–224.

[10] 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.

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

[12] C. LUBICH AND A. OSTERMAN, *Multigrid dynamic iteration for parabolic problems*, BIT, 27 (1987), pp. 216–234.

[13] 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.

[14] R. C. MACCAMY AND P. WEISS, *Numerical solution of Volterra integral equations*, Nonlinear Anal., 3 (1979), pp. 677–695.

[15] 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.

[16] U. MIEKKALA AND O. NEVANLINNA, *Convergence of dynamic iteration methods for initial value problems*, SIAM J. Sci. Stat. Comp., 8 (1987), pp. 459–467.

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

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

[19] O. NEVANLINNA, *Remarks on Picard-Lindelöf iteration, Part II*, BIT, 29 (1989), pp. 535–562.

[20] ——, *Linear acceleration of Picard-Lindelöf iteration*, Numer. Math., 57 (1990), pp. 147–156.

[21] O. NEVANLINNA AND F. ODEH, *Remarks on the convergence of the waveform relaxation method*, Numerical Functional Anal. Optimization, 9 (1987), pp. 435–445.

[22] F. ODEH, A. RUEHLI, AND C. CARLIN, *Robustness aspects of an adaptive wave-form relaxation scheme*, in Proceedings of the IEEE Int. Conf. on Circuits and Comp. Design, Rye,N.Y., October 83, pp. 396–440.

[23] F. ODEH, A. RUEHLI, AND P. DEBEFVE, *Waveform techniques*, in Circuit Analysis, Simulation and Design, Part 2, A. Ruehli, ed., North-Holland, 1987, pp. 41–127.

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

[25] M. REICHELT, *Accelerated Waveform Relaxation Techniques for the Parallel Transient Simulation of Semiconductor Devices*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1993.

[26] 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.

[27] M. REICHELT, J. WHITE, J. ALLEN, AND F. ODEH, *Waveform relaxation applied to transient device simulation*, in Proceedings of the IEEE Int. Conf. on Circuits and Systems, Espoo, Finland, October 83, pp. 396–440.

[28] 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.

[29] 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.

[30] 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.

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

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

[33] S. VANDEWALLE AND R. PIESSENS, *Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1330–1346.

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

[35] J. WHITE AND F. ODEH, *A connection between the convergence properties of waveform relaxation and the A-stability of multirate integration methods*, in Proceedings of the NASECODE VII Conference, Copper Mountain, Colorado, 1991.

[36] J. WHITE, F. ODEH, A. VINCENTELLI, AND A. RUEHLI, *Waveform relaxation: Theory and practice*, Trans. of the Society for Computer Simulation, 2 (1985), pp. 95–133.

[37] 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.

[38] H. WIDOM, *Toeplitz matrices*, in Studies in Real and Complex Analysis, J. I. I. Hirschman, ed., vol. Vol. 3, The Mathematical Association of America, 1965, pp. 179–209.