SIMULATION OF VLSI CIRCUITS: RELAXATION TECHNIQUES

Peter Lory, München (Germany)

1. Introduction

Circuit simulation programs have proven to be most important CAD tools for the analysis of the electrical performance of integrated circuits. Depending on the number of modeled transistors, these simulators require the numerical solution of initial value problems for very large, sparse systems of differential (or even differential-algebraic) equations. For a survey see Refs. 1 and 2.

Waveform relaxation (WR) has been proposed in Ref. 5 for the numerical solution of these initial value problems. It applies the well-known Gauss-Seidel and Jacobi principles for the numerical solution of systems of algebraic equations on the function space level (see also Refs. 8 and 11). Each differential equation of the system, which models an integrated circuit, corresponds to one node in this network. In its simplest form, WR solves these equations as single differential equations in one unknown, and these solutions are iterated until convergence. If this kind of node-by-node decomposition strategy is used for circuits with even just a few tightly coupled nodes, the WR algorithm will converge very slowly. Its efficiency can be greatly improved by lumping together tightly coupled nodes and partitioning the system correspondingly. The relaxation principle is then applied in a blockwise manner.

In practical applications, the groups of differential equations in the partitioned system have to be solved numerically in each cycle of the WR iteration. As stiffness is a characteristic feature in VLSI applications, most circuit simulation programs (and the present note) use the stiffly stable backward differentiation formulas for discretization (see Ref. 3). Other approaches, like Runge-Kutta related methods, can be applied in principle, too (see Ref. 9).

The present note shows that in the linear case the discretized WR algorithm is equivalent to the (algebraic) block relaxation method if the latter is applied to a properly defined linear system of algebraic equations (Section 2). Naturally the dimension of this system is greatly enlarged. However, the eigenvalue problem of the corresponding iteration matrix can be reduced to the original dimension. Consequently, estimates of the convex hull of its spectrum can be computed efficiently by an adaptive technique. This suggests a method for accelerating the rate of convergence for WR (Section 3).

409

2. The Discretized Waveform Relaxation in the Linear Case

Valuable insight into the WR method can be gained by an investigation of the linear case

\[(2.1) \quad \dot{C}x = Fx + f, \quad x(0) = x_0, \quad 0 \leq t \leq T. \]

Here, \( C \) ("capacitance matrix") and \( F \) are constant \((n \times n)\)-matrices, and \( f \) is a constant vector. Let Equ. (2.1) be partitioned as motivated in Section 1:

\[(2.2) \quad C = \begin{bmatrix} C_{11} & \cdots & C_{1p} \\ \vdots & \ddots & \vdots \\ C_{p1} & \cdots & C_{pp} \end{bmatrix}, \quad F = \begin{bmatrix} F_{11} & \cdots & F_{1p} \\ \vdots & \ddots & \vdots \\ F_{p1} & \cdots & F_{pp} \end{bmatrix}, \]

where \( C_{ii} \) and \( F_{ii} \) are \((n_i \times n_i)\)-matrices \((i = 1, \ldots, p)\) and \( n_1 + \cdots + n_p = n \). The dimensions of the other submatrices and the partition of the vectors \( f \) and \( x \) are defined accordingly. Let the linear multistep formula

\[(2.3) \quad \sum_{k=0}^{s} \alpha_k^* C_{\ell-k}x_{\ell-k} = h \sum_{k=0}^{s} \beta_k^* F_{\ell-k}x_{\ell-k}, \]

be used, and let the interval \( 0 \leq t \leq T \) be subdivided by equidistant points

\[(2.4) \quad t_{\ell} = \ell \cdot h \quad (\ell = 0, \ldots, N), \quad h = T/N. \]

For the sake of simplicity the same (uniform) stepsize \( h \) and the same formula (2.3) is used for the discretization of all the differential equations. In reality, one of the advantages of the WR algorithm is to let different subsystems be integrated according to their individual dynamical properties. Hence, the last assumption is somewhat artificial. However, this simplification allows a concise description of the basic ideas and valuable first insights. Let Equ. (2.1) be discretized by (2.3):

\[(2.3) \quad \sum_{k=0}^{s} \alpha_k^* C_{\ell-k}x_{\ell-k} = h \sum_{k=0}^{s} \beta_k^* F_{\ell-k}x_{\ell-k}, \]

and let \( x_{\ell} \) denote the approximation of \( x(t_{\ell}) \). Define

\[ A := \begin{bmatrix} \alpha_0 & 0 \\ \alpha_1 & \alpha_0 \\ \vdots & \vdots \\ \alpha_s & \cdots & \alpha_1 & \alpha_0 \\ 0 & \alpha_s & \cdots & \alpha_1 & \alpha_0 \end{bmatrix} \in \mathbb{R}^{N,N}, \quad B := \begin{bmatrix} \beta_0 \\ \beta_1 \beta_0 \\ \beta_s \cdots \beta_1 \beta_0 \\ 0 & \beta_s \cdots \beta_1 \beta_0 \end{bmatrix} \in \mathbb{R}^{N,N}, \]
M_{\pi}(h,N) := C \ast A - h \ast F \ast B \quad (**: \text{direct product}),

z := (x_{11}, \ldots, x_{IN}; \ldots; x_{n1}, \ldots, x_{nN})^T \in \mathbb{R}^{N \times n}.

Let the discretized WR Gauss-Seidel (Jacobi) method be applied to system (2.1), which is partitioned according to (2.2). It is easy to see that this algorithm is equivalent to the standard algebraic block Gauss-Seidel (Jacobi) method if the latter is applied to the following system of linear algebraic equations:

(2.5) \quad M_{\pi}(h,N) \cdot z = \text{right hand side}.

For the definition of block (or group) iterative methods see e.g. Refs. 10 and 12. Throughout this paper, the index \( \pi \) refers to the following partition of the matrices into \( p^2 \) blocks: The diagonal blocks have sizes \( n_{1N} \times n_{1N}, \ldots, n_{pN} \times n_{pN} \); the dimensions of the off-diagonal blocks are chosen accordingly. Please note that the enlargement of the size of the matrices from \( n \) to \( n \times n \) is for theoretical purposes only. In actual computations, the explicit formation of the matrix \( M_{\pi} \) is unnecessary. Further, the eigenvalue problem for the corresponding iteration matrix can be reduced to dimension \( n \). This fact will be proven (and exploited) in the following section.

3. \textbf{Accelerating the Rate of Convergence}

Let

\[
M_{\pi} = D_{\pi} + L_{\pi} + U_{\pi},
\]

where \( D_{\pi} \) is the block diagonal part, \( L_{\pi} \) the strictly lower block triangular part, \( U_{\pi} \) the strictly upper block triangular part. Here, the blocking refers to partition \( \pi \). The \textit{iteration matrix of the block Gauss-Seidel method is}

(3.1) \quad G_{\pi} := -(D_{\pi} + L_{\pi})^{-1} \cdot U_{\pi}.

\( G_{\pi} \) is a \((nN \times nN)\)-matrix. Let the \((n \times n)\)-matrices \( C \) and \( F \) of Equ. (2.1) be split into

\[
C = D_C + L_C + U_C, \quad F = D_F + L_F + U_F,
\]

where \( D_C \) and \( D_F \) are the block diagonal parts, \( L_C \) and \( L_F \) the strictly lower block triangular parts, \( U_C \) and \( U_F \) the strictly upper block triangular parts. Here, the blocking refers to partition (2.2). Then

\[
D_{\pi} = D_C \ast A - h \ast D_F \ast B, \quad L_{\pi} = L_C \ast A - h \ast L_F \ast B,
\]
\[ U_\pi = U_C \ast A - h \ast U_F \ast B \]

With \( P \) as the permutation matrix of Prop. 3 of Chap. 12 in Ref. 4

\[ P^T U_\pi P = A \ast U_C - h \ast B \ast U_F. \]

This matrix is lower block triangular with diagonal blocks

\[ \alpha_0 U_C - h\beta_0 U_F. \]

Similarly,

\[ P^T (D_\pi + L_\pi) P = A \ast (D_C + L_C) - h \ast B \ast (D_F + L_F) , \]

which is a lower block triangular matrix with diagonal blocks

\[ (\alpha_0 D_C - h\beta_0 D_F) + (\alpha_0 L_C - h\beta_0 L_F). \]

Consequently,

\[ P^T G_\pi P = -(P^T (D_\pi + L_\pi) P)^{-1} (P^T U_\pi P) \]

is a lower block triangular matrix with diagonal blocks \( G \), where

\[ G := -(\alpha_0 D_C - h\beta_0 D_F) + (\alpha_0 L_C - h\beta_0 L_F) \]

This \( (n \times n) \)-matrix is the iteration matrix for the block Gauss-Seidel method if applied to \( \alpha_0 C - h\beta_0 F \). The blocking refers to partition (2.2). So the following theorem is proven:

3.1 Theorem: Let \( \lambda_1, ..., \lambda_n \) be the eigenvalues of \( G \) (see Equ. (3.2)). Then \( \lambda_1, ..., \lambda_1 \) (N-fold); \( ..., \lambda_n, ..., \lambda_n \) (N-fold) are the eigenvalues of \( G_\pi \), where \( G_\pi \) is the iteration matrix of the WR Gauss-Seidel algorithm (see Equ. (3.1)).

A corresponding result for the WR Jacobi algorithm can be proven in a completely analogous manner. Theorem 3.1 extends Theorem 6.1 in Ref. 11 and gives a complete explanation of the background of these results.

Chebyshev acceleration is a powerful tool for speeding up the convergence of an iterative method (see Refs. 6 and 7). It depends on two parameters which can be chosen from knowledge of the convex hull of the spectrum of the iteration matrix. Ref. 7 gives an adaptive procedure for estimating this convex hull. This suggests the following algorithm.

3.2 Algorithm (Outline):

Step 1) (Estimating the convex hull of the spectrum of \( G_\pi \)) Perform block Gauss-Seidel iterations (with Chebyshev acceleration) on the \( (n \times n) \)-matrix \( \alpha_0 C - h\beta_0 F \). Here, the blocking refers to partition (2.2). Estimate the convex hull of the spectrum of \( G \) (see Equ. (3.2)). Theorem 3.1 immediately gives an estimate for the
convex hull of the spectrum of the \((nN \times nN)\)-matrix \(G_\pi\).

**Step 2)** Start WR Gauss-Seidel (with Chebyshev acceleration) based upon this knowledge of the convex hull of the spectrum of the corresponding iteration matrix \(G_\pi\) and continue by iteratively improving the above estimates of the convex hull of this spectrum.

Please note that Step 1) is relatively cheap. The basic assumption for Chebyshev acceleration is that the real part for any eigenvalue \(\lambda\) of the iteration matrix satisfies \(\text{Re}(\lambda) < 1\). In view of Theorem 3.1 this can be guaranteed if the spectral radius \(\rho(G) < 1\). This is true if the stepsize \(h\) is sufficiently small and if the spectral radius of the Gauss-Seidel iteration matrix for \(C\) is less than 1. Note that \(C\) is strictly diagonally dominant in many practical applications.

**Experiments**

The following first numerical experiments give an impression of the potential of the method. In the ring oscillator with \(n = 201\) nodes all transistors were replaced by resistors with \(R_T = 10^{12}\Omega\). For the arising RC-network the matrices \(C\) and \(F\) in (2.1) are symmetric. Because of this symmetry, the WR SSOR method was used instead of the WR Gauss-Seidel method. Then the eigenvalues of the corresponding iteration matrix are real. The following table compares the asymptotic rates of convergence \(R_\infty\) for the WR SSOR method without and with Chebyshev acceleration (cf. Ref. 12). In (2.3) the backward differentiation formula with \(s = 4\), \(\alpha_0 = 25/12\), \(\beta_0 = 1\) was used; stepsize \(h = 0.02\) nsec.

<table>
<thead>
<tr>
<th>(C_p) (pF)</th>
<th>(R_\infty) without Ch. acc.</th>
<th>(R_\infty) with Ch. acc.</th>
<th>asymptotic acceleration factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.005</td>
<td>6.62</td>
<td>8.01</td>
<td>1.21</td>
</tr>
<tr>
<td>0.05</td>
<td>2.68</td>
<td>4.03</td>
<td>1.50</td>
</tr>
<tr>
<td>0.5</td>
<td>0.59</td>
<td>1.62</td>
<td>2.75</td>
</tr>
<tr>
<td>5.0</td>
<td>0.08</td>
<td>0.57</td>
<td>7.13</td>
</tr>
</tbody>
</table>

The table indicates that for tightly coupled circuits the asymptotic acceleration factor is quite high.

**Acknowledgement:** The author is indebted to Mr. R. Gollreiter for performing the numerical experiments.

**References**


5 Lelarasmee, E.: The waveform relaxation method for the time domain analysis of large scale integrated circuits: theory and applications. Ph.D. dissertation, University of California, Berkeley; also Memo UCB/ERL M82/40, 1982


8 Miekkala, U.; Nevanlinna, O.: Sets of convergence and stability regions. BIT 27 (1987) 554-584


Author’s address:
Peter Lory, Mathematisches Institut, Technische Universität München, Postfach 20 24 20, D-8000 München 2, Germany (West)