RC24310 (W0707-106) July 17, 2007
Electrical Engineering

IBM Research Report

Convergence of Transverse Waveform Relaxation for the Electrical Analysis of Very Wide On-chip Transmission Line Buses

I. M. Elfadel
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Convergence of Transverse Waveform Relaxation for the Electrical Analysis of Very Wide On-chip Transmission Line Buses

I. M. Elfadel
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
elfadel@us.ibm.com
July 10, 2007

This paper is dedicated to the authors of [1] on the occasion of its Silver Jubilee.

***Abstract*— In this paper, we study the convergence and the approximation error of the transverse waveform relaxation (TWR) method for the analysis of very wide, on-chip multiconductor transmission line systems. Significant notational simplicity is achieved in the analysis using a splitting framework for the per-unit-length matrix parameters of the transmission lines. This splitting enables us to show that the state transition matrix of the coupled lines satisfies a linear Volterra integral equation of the second kind whose solution is generated by the TWR method as a summable series of iterated kernels with decreasing norms. The upper bounds on these norms are proved to be $O(k^r/r!)$ where r is the number of iterations and k is a measure of the electromagnetic couplings between the lines. Very fast convergence is guaranteed in the case of weak coupling ($k << 1$). These favorable convergence properties are illustrated on a fully embedded implementation of TWR, called TWRTL, that uses the application programming interface of an industrial circuit analysis program (IBM's PowerSpice). TWRTL is shown to converge with few (≈ 3) Gauss-Jacobi iterations to the exact solution on a full test suite of industrial VLSI buses in a modern $65nm$ CMOS process.**

Index Terms - **VLSI, circuit simulation, waveform relaxation, coupled transmission lines, on-chip buses.**

I. Introduction

The basic idea of the transverse waveform relaxation (TWR) algorithm for the analysis of coupled transmission lines is to analyze the multiconductor system one transmission line at a time [2], [3]. This is achieved iteratively until convergence by transforming the electromagnetic couplings between lines into independent voltage and current sources that are sometimes called *relaxation* sources. The waveform imprinted by a given source at iteration $r + 1$ is computed based on the states of the lines at iteration r. This method of waveform relaxation is well adapted to the parallel data paths of the transmission lines and can be construed a form of decoupling between lines in order to facilitate their analysis. But unlike modal decoupling which requires the explicit diagonalization of the per-unit-length (PUL) $\boldsymbol{RLGC}$ matrices [4], the TWR decoupling, in its simplest form, is based on an explicit splitting of these $\boldsymbol{RLGC}$ matrices. There are several motivations for developing and implementing an algorithm such as TWR. One such motivation is the limited capacity of available circuit analysis programs in handling a large number of coupled transmission lines. When the generalized method of characteristics is used as for instance in HSPICE's W-element [5] or PowerSpice's RLINE [6], the maximum number of coupled transmission lines that can be efficiently analyzed is limited to about 10 lines [3]. This is because of the need to fit rational approximations to the transcendental functions involved in the formulation of the generalized method of characteristics. This fitting is needed *even* when the PUL matrix parameters of the lines are independent of frequency. The dependence of PUL parameters on frequency will make matters worse. For on-chip transmission lines, synthesis methods of the type used in [7], [8] can extend the analysis capacity to about 24 lines [9], but it is well known that these synthesis methods are not suitable for low-loss lines where they tend to require an excessive number of π sections in order to capture the time-of-flight delays of the lines. Furthermore, using the synthesis approach for a large number of lines (≥ 25) in the presence of a large number of nonlinear terminations seems to challenge the capabilities of present day SPICE-level simulators [9]. Another motivation for the need to study and implement TWR is the ever more stringent signal integrity and noise immunity requirements on on-chip VLSI buses and off-chip MCM and PCB communications links of present and future computing systems. Taking the digital on-chip VLSI case as an example, the very wide data buses used to transfer data between caches and processors are typically enmeshed within the chip power bus. Analyzing the interaction between various kinds of interconnect noise (capacitive, inductive, and the one induced by the finite impedance of the return paths) on the one hand, and the delta-I noise of the power supply, on the other, is based on the premise of a robust method for simulating a large number of both signal and power transmission lines interacting together [9].

Since TWR deals with coupled transmission lines one line at a time, it is expected that the simulation bottleneck of the generalized method of characteristics or the synthesized circuit approach will be overcome. The main concerns for TWR as for any waveform relaxation process are convergence and convergence to the right solution. In [3], [10], [11] very encouraging results regarding the convergence behavior of TWR have been reported. The several TWR examples therein were observed to converge in fewer than five iterations *even* for buses containing as many as 80 lines with randomly selected PUL parameters. One main contribution of this paper is to provide a rigorous conceptual and theoretical framework for explaining the convergence results reported in these earlier publications. Our work shows that the TWR iterates of the state transition matrix of the coupled transmission lines are guaranteed to converge as they are associated with the Picard iterates of a Volterra integral equation that can be readily obtained from the Telegrapher's equations. Furthermore, we show that the state transition matrix of the coupled transmission lines can be written as the limit of the Volterra series whose terms are TWR iterated kernels. We also derive a TWR error estimate based on the truncation of the Volterra series at a given number of terms corresponding to a maximum number of TWR iterations.

Another contribution of this paper is in benchmarking our own implementation of TWR against the results of a traditional circuit analysis program for a realistic on-chip bus architecture. This benchmarking is meant to address the issue of "TWR convergence to the right solution." The main observation we make based on our experiments is that TWR converges to the correct solution faster on active lines than on quiet lines. It also converges faster on the far ends than on the near ends. In the crosstalk experiments that we have conducted, most of the latter TWR iterations are used to converge on the noise waveforms while the waveforms on the active lines are essentially correct at the first or second iteration.

The third contribution of this paper is in terms of the software architecture of our TWR implementation. According to [12], a mixture of MATLAB scripts and HSPICE netlists were used in deriving the results of [3], [10], [11]. Given the corporate CAD context of our work, this implementation style was not an option we could entertain. Given the limited resources in time and headcount, doing development work on the core engine of the corporate circuit analysis program, PowerSpice, was not an option either. We have therefore opted for a software architecture in which the TWR process is implemented as part of the CAD application that needs it (e.g., bus signal integrity) rather than in the core engine of the SPICE-level simulator. The communication between the CAD application and the circuit application is accomplished via an application programming interface that enables both TWR and the simulation engine to run in-core without any file IO.

This paper is organized as follows. In Section II, the basic idea of TWR is presented along with its circuit representation. In Section III, the TWR iteration is recast in a matrix form by splitting the PUL matrices of the multiconductor transmission system. Furthermore, the Volterra integral equation satisfied by the state transition matrix of the transmission system is introduced. In Section IV, the main theoretical result of this paper, namely, the unconditional convergence of TWR to a solution of the ordinary differential equation satisfied by the state transition matrix, is stated and proved. In Section V, an error estimate is given for the TWR iterations that clearly shows the impact of electromagnetic couplings on the TWR convergence rate. Section VI is devoted to the description of an embedded implementation of TWR as part of a CAD flow for the analysis of very-wide, global on-chip RLC buses. This implementation is contrasted with the earlier implementations of the WR algorithm in the context of VLSI circuit analysis. In Section VII, an extensive set of numerical examples to validate TWR for on-chip RLC buses is described. The very favorable convergence behavior of TWR is illustrated on wide industrial buses with as many as 40 bit-lines. Conclusions and suggestions for future research work are given in section VIII.

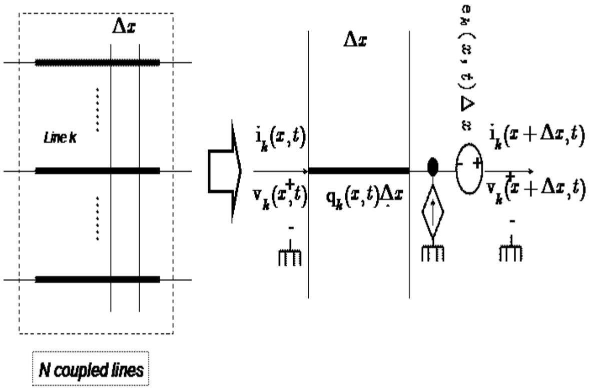

Fig. 1. Transformation of electromagnetic couplings into distributed voltage and current relaxation sources.

II. Transverse Waveform Relaxation for Transmission Lines

We assume we have an interconnect bus of N parallel coupled transmission lines of length L. The electrical state at any point $x \in [0, L]$ and any time instant $t \geq 0$ is described by the first-order Telegrapher's partial differential equation

$$\begin{aligned} \frac{\partial}{\partial x}\boldsymbol{v}(x,t) &= -\boldsymbol{R}\boldsymbol{i}(x,t) - \boldsymbol{L}\frac{\partial}{\partial t}\boldsymbol{i}(x,t) \\ \frac{\partial}{\partial x}\boldsymbol{i}(x,t) &= -\boldsymbol{G}\boldsymbol{v}(x,t) - \boldsymbol{C}\frac{\partial}{\partial t}\boldsymbol{v}(x,t) \end{aligned} \qquad (1)$$

where $\boldsymbol{R}, \boldsymbol{L}, \boldsymbol{G}, \boldsymbol{C} \in \mathbb{R}^N \times \mathbb{R}^N$ are the PUL matrices of the N lines and $\boldsymbol{v}(x,t), \boldsymbol{i}(x,t) \in \mathbb{R}^N$ are the voltages and currents at point $x \in [0, L], t \geq 0$. The voltage and current equations of the k-th line can be written as [3], [10], [11]

$$
\begin{aligned}
\frac{\partial}{\partial x} v_k(x,t) &= -R_{kk} i_k(x,t) - L_{kk}\frac{\partial}{\partial t} i_k(x,t) \\
&- \sum_{l=1, l\neq k}^{N} \left[R_{kl} i_l + L_{kl} \frac{\partial}{\partial t} i_l(x,t) \right] \\
\frac{\partial}{\partial x} i_k(x,t) &= -G_{kk} v_k(x,t) - C_{kk}\frac{\partial}{\partial t} v_k(x,t) \\
&- \sum_{l=1, l\neq k}^{N} \left[G_{kl} v_l(x,t) + C_{kl} \frac{\partial}{\partial t} v_l(x,t) \right]
\end{aligned}
$$

The summations in the above equations represent the impact of electromagnetic couplings on line k due to all the other lines in the bus. The equations of Line k can be rewritten as

$$
\begin{aligned}
\frac{\partial}{\partial x} v_k(x,t) &= -R_{kk} i_k(x,t) - L_{kk}\frac{\partial}{\partial t} i_k(x,t) \\
&+ e_k(x,t) \\
\frac{\partial}{\partial x} i_k(x,t) &= -G_{kk} v_k(x,t) - C_{kk}\frac{\partial}{\partial t} v_k(x,t) \\
&+ q_k(x,t)
\end{aligned}
$$

where the aggregate electromagnetic couplings $e_k(x,t)$ and $q_k(x,t)$ are given by

$$
\begin{aligned}
e_k(x,t) &= -\sum_{l=1, l\neq k}^{N} \left[R_{kl} i_l(x,t)) + L_{kl} \frac{\partial}{\partial t} i_l(x,t) \right] \\
q_k(x,t) &= -\sum_{l=1, l\neq k}^{N} \left[G_{kl} v_l(x,t) + C_{kl} \frac{\partial}{\partial t} v_l(x,t) \right]
\end{aligned} \tag{2}
$$

Thus the equations describing the state of each line k in the bus are those of a single transmission line under excitation by distributed voltage and current sources whose strengths per-unit-length are given by $e_k(x,t)$ and $q_k(x,t)$, respectively. The basic idea of transverse waveform relaxation for transmission lines [11] is to assume that the state of these sources is known at iteration r, and that the state of each transmission line can then be determined according to the recursions

$$
\begin{aligned}
\frac{\partial}{\partial x} v_k^{(r+1)} &= -R_{kk} i_k^{(r+1)} - L_{kk}\frac{\partial}{\partial t} i_k^{(r+1)} \\
&+ e_k^{(r)}
\end{aligned} \tag{3}
$$

$$
\begin{aligned}
\frac{\partial}{\partial x} i_k^{(r+1)} &= -G_{kk} v_k^{(r+1)}(x,t) - C_{kk}\frac{\partial}{\partial t} v_k^{(r+1)} \\
&+ q_k^{(r)}
\end{aligned} \tag{4}
$$

$$
e_k^{(r+1)} = -\sum_{l=1, l\neq k}^{N} \left[R_{kl} i_l^{(r+1)} + L_{kl} \frac{\partial}{\partial t} i_l^{(r+1)} \right] \tag{5}
$$

$$
q_k^{(r+1)} = -\sum_{l=1, l\neq k}^{N} \left[G_{kl} v_l^{(r+1)} + C_{kl} \frac{\partial}{\partial t} v_l^{(r+1)} \right] \tag{6}
$$

where the argument (x,t) has been dropped throughout the equations. The initial conditions for the distributed coupling sources in the above recursion are

$$
\begin{aligned}
e_k^{(0)} &= 0,\ 1 \leq k \leq N \\
q_k^{(0)} &= 0,\ 1 \leq k \leq N
\end{aligned}
$$

In the context of TWR, the distributed coupling sources are sometimes called *relaxation sources*.

Figure 1 illustrates this transverse decomposition of transmission lines for a segment of length Δx of the bus. It is to be noted that this decomposition can work with any electrical representation of the Δx segment, be it distributed or lumped. Figure 2 shows this decomposition when the Δx segment is represented with an RLC section. In this latter case, the circuit representation of TWR amounts to replacing the electrostatic coupling of the floating capacitance with an independent current source and the magnetostatic coupling of the mutual inductance with an independent voltage source. The states of these current and voltage sources are determined iteratively according to the equations given in (2).

The TWR recursion described above as well as the ones in [3], [10], [11] are for what is known as a Gauss-Jacobi iteration [13]. Conceivably the updates of the electrical states of the transmission lines can be done according to a Gauss-Seidel iteration where the most recent states are included in the calculation of the sources. Assuming that the states of transmission lines $1 \leq l \leq k-1$ have been updated, the state of transmission line k at iteration $r+1$ is updated as follows:

$$
\begin{aligned}
e_k^{(r)} &= -\sum_{l=1}^{k-1} \left[R_{kl} i_l^{(r+1)} + L_{kl} \frac{\partial}{\partial t} i_l^{(r+1)} \right] \\
&- \sum_{l=k+1}^{N} \left[R_{kl} i_l^{(r)} + L_{kl} \frac{\partial}{\partial t} i_l^{(r)} \right] \\
q_k^{(r)} &= -\sum_{l=1}^{k-1} \left[G_{kl} v_l^{(r+1)} + C_{kl} \frac{\partial}{\partial t} v_l^{(r+1)} \right] \\
&- \sum_{l=k+1}^{N} \left[G_{kl} v_l^{(r)} + C_{kl} \frac{\partial}{\partial t} v_l^{(r)} \right] \\
\frac{\partial}{\partial x} v_k^{(r+1)} &= -R_{kk} i_k^{(r+1)} - L_{kk}\frac{\partial}{\partial t} i_k^{(r+1)} \\
&+ e_k^{(r)} \\
\frac{\partial}{\partial x} i_k^{(r+1)} &= -G_{kk} v_k^{(r+1)}(x,t) - C_{kk}\frac{\partial}{\partial t} v_k^{(r+1)} \\
&+ q_k^{(r)}
\end{aligned}
$$

The main difference between the Gauss-Jacobi TWR and the Gauss-Seidel TWR is in the way the relaxation sources are updated. In Gauss-Jacobi only the states of the sources from the previous iteration are involved in the update, while in Gauss-Seidel, the most recent state information is used in the relaxation source update. The theoretical and numerical results of this paper are for the Gauss-Jacobi TWR process. The results of the Gauss-Seidel TWR will be given in a future paper.

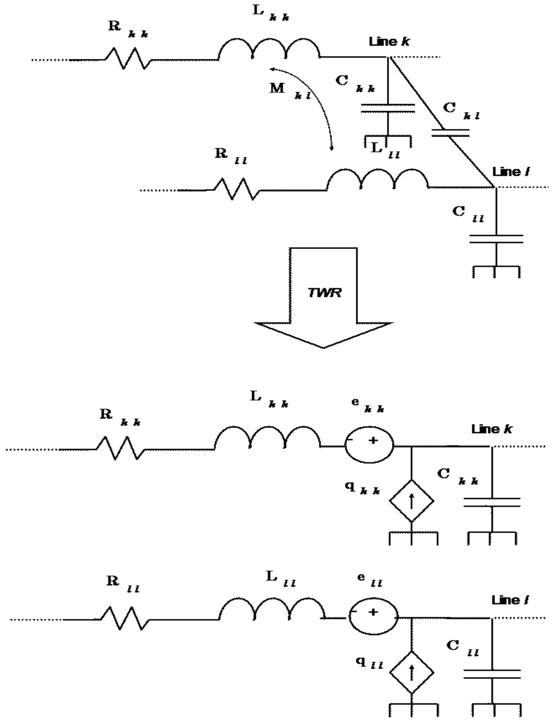

Fig. 2. Transformation of electromagnetic couplings into independent voltage and current relaxation sources when the the transmission line macromodel is made of cells of lumped parameters.

III. TWR Convergence Framework

In this section, we set the stage for proving the convergence of the TWR iteration for transmission lines. While convergence proofs for waveform relaxation are as old as the WR idea itself [1], [14]–[16], they were all for lumped parameters systems described by ordinary or ordinary-algebraic differential equations. WR has been applied to distributed parameter systems described by partial differential equations after their spatial dependence has been discretized. Of note in this regard is the work on the nonlinear diffusion equations describing the charge behavior of semiconductor devices [17], [18] and the work on the reaction diffusion equation [19]. In this prior art, WR convergence proofs and convergence acceleration algorithms are all within the dynamic iteration framework of ordinary differential equations. The present paper as well as [3], [10], [11] apply dynamic iteration in the partial differential equation context of the Telegrapher's equations. Indeed, the dynamic iteration equations (3) and (4) preserve the distributed-parameter nature of the coupled transmission lines, and so does the TWR convergence analysis presented in this paper.

The TWR convergence proof has two ingredients. The first is the splitting of the PUL $N \times N$ $\boldsymbol{RLGC}$ matrices describing the transmission lines and the second is to express the TWR process in terms of an iterated integral kernel resulting from a Volterra integral equation of the second kind. The PUL matrix splitting is the mathematical counterpart of the decoupling of transmission lines into single lines excited by distributed sources. The Volterra integral equation is the one that results from "solving" for the electrical states of the single lines under the assumption that the states of the excitation sources are known.

A. Splitting the Telegrapher's PDE's

In the Gauss-Jacobi TWR, each of the PUL $\boldsymbol{R}, \boldsymbol{L}, \boldsymbol{G}, \boldsymbol{C} \in \mathbb{R}^N \times \mathbb{R}^N$ matrices is split into a diagonal part and a non-diagonal part. The diagonal part will be denoted with $\boldsymbol{D}_{\boldsymbol{X}}$ and the non-diagonal part with $\boldsymbol{N}_{\boldsymbol{X}}$ where $\boldsymbol{X}$ refers to the original matrix as follows:

$$\begin{aligned} \boldsymbol{R} &= \boldsymbol{D}_{\boldsymbol{R}} + \boldsymbol{N}_{\boldsymbol{R}} \\ \boldsymbol{L} &= \boldsymbol{D}_{\boldsymbol{L}} + \boldsymbol{N}_{\boldsymbol{L}} \\ \boldsymbol{G} &= \boldsymbol{D}_{\boldsymbol{G}} + \boldsymbol{N}_{\boldsymbol{G}} \\ \boldsymbol{C} &= \boldsymbol{D}_{\boldsymbol{C}} + \boldsymbol{N}_{\boldsymbol{C}} \end{aligned} \tag{7}$$

The $\boldsymbol{D}$ matrices are the self parameters of the lines while the $\boldsymbol{N}$ matrices are the coupling parameters. We also use the notation

$$\begin{aligned} \boldsymbol{D}_{\boldsymbol{Z}}(\omega) &= \boldsymbol{D}_{\boldsymbol{R}} + j\omega \boldsymbol{D}_{\boldsymbol{L}} \\ \boldsymbol{N}_{\boldsymbol{Z}}(\omega) &= \boldsymbol{N}_{\boldsymbol{R}} + j\omega \boldsymbol{N}_{\boldsymbol{L}} \\ \boldsymbol{D}_{\boldsymbol{Y}}(\omega) &= \boldsymbol{D}_{\boldsymbol{G}} + j\omega \boldsymbol{D}_{\boldsymbol{C}} \\ \boldsymbol{N}_{\boldsymbol{Y}}(\omega) &= \boldsymbol{N}_{\boldsymbol{G}} + j\omega \boldsymbol{N}_{\boldsymbol{C}} \end{aligned} \tag{8}$$

where $j\omega$ is the Fourier variable. Using the above matrix splittings, the Telegrapher's equation can be written in the time domain as

$$\begin{aligned} \frac{\partial}{\partial x}\boldsymbol{v}(x,t) &= -\boldsymbol{D}_{\boldsymbol{R}}\boldsymbol{i}(x,t) - \boldsymbol{D}_{\boldsymbol{L}}\frac{\partial}{\partial t}\boldsymbol{i}(x,t) - \boldsymbol{N}_{\boldsymbol{R}}\boldsymbol{i}(x,t) \\ &- \boldsymbol{N}_{\boldsymbol{L}}\frac{\partial}{\partial t}\boldsymbol{i}(x,t) \\ \frac{\partial}{\partial x}\boldsymbol{i}(x,t) &= -\boldsymbol{D}_{\boldsymbol{G}}\boldsymbol{v}(x,t) - \boldsymbol{D}_{\boldsymbol{C}}\frac{\partial}{\partial t}\boldsymbol{v}(x,t) - \boldsymbol{N}_{\boldsymbol{G}}\boldsymbol{v}(x,t) \\ &- \boldsymbol{N}_{\boldsymbol{C}}\frac{\partial}{\partial t}\boldsymbol{v}(x,t) \end{aligned} \tag{9}$$

and in the frequency domain as

$$\begin{aligned} \frac{\partial}{\partial x}\boldsymbol{V}(x,\omega) &= -\boldsymbol{D}_{\boldsymbol{Z}}(\omega)\boldsymbol{I}(x,\omega) - \boldsymbol{N}_{\boldsymbol{Z}}(\omega)\boldsymbol{I}(x,\omega) \\ \frac{\partial}{\partial x}\boldsymbol{I}(x,\omega) &= -\boldsymbol{D}_{\boldsymbol{Y}}(\omega)\boldsymbol{V}(x,\omega) - \boldsymbol{N}_{\boldsymbol{Y}}(\omega)\boldsymbol{V}(x,\omega) \end{aligned}$$

The above two first-order equations can be combined in a single first order equation of order $2N$

$$\frac{\partial}{\partial x}\boldsymbol{\Phi}(x,\omega) = -\boldsymbol{D}(\omega)\boldsymbol{\Phi}(x,\omega) - \boldsymbol{N}(\omega)\boldsymbol{\Phi}(x,\omega) \tag{10}$$

where we have defined $\boldsymbol{\Phi}(x,\omega) = \begin{bmatrix} \boldsymbol{V}(x,\omega) \\ \boldsymbol{I}(x,\omega) \end{bmatrix} \in \mathbb{R}^{2N}$ and

$$\begin{aligned} \boldsymbol{D}(\omega) &= \begin{bmatrix} \boldsymbol{O} & \boldsymbol{D}_{\boldsymbol{Z}}(\omega) \\ \boldsymbol{D}_{\boldsymbol{Y}}(\omega) & \boldsymbol{O} \end{bmatrix} \in \mathbb{R}^{2N} \times \mathbb{R}^{2N} \\ \boldsymbol{N}(\omega) &= \begin{bmatrix} \boldsymbol{O} & \boldsymbol{N}_{\boldsymbol{Z}}(\omega) \\ \boldsymbol{N}_{\boldsymbol{Y}}(\omega) & \boldsymbol{O} \end{bmatrix} \in \mathbb{R}^{2N} \times \mathbb{R}^{2N} \end{aligned}$$

Equation (10) describes a linear system with a dynamic matrix $\boldsymbol{D}(\omega)$ and an input matrix $\boldsymbol{N}(\omega)$ through which the electromagnetic couplings impact the electrical states of the individual lines. Note that if $\boldsymbol{N}(\omega)$ is the zero matrix, we get N independent single-line Telegrapher's equations. The general solution of Equation (10) is given by

$$\begin{aligned}\boldsymbol{\Phi}(x,\omega) &= e^{-\boldsymbol{D}(\omega)x}\boldsymbol{\Phi}(0,\omega) \\ &- \int_0^x e^{-\boldsymbol{D}(\omega)(x-\zeta)}\boldsymbol{N}(\omega)\boldsymbol{\Phi}(\zeta,\omega)d\zeta.\end{aligned} \tag{11}$$

If the initial vector $\boldsymbol{\Phi}(0,\omega)$ (voltages and current at the near end ports) were known, the above equation would have been, for each spatial variable x and at each frequency ω, a Volterra integral equation of the second kind. This type of integral equations can be solved according to the Picard iteration [20]

$$\begin{aligned}\boldsymbol{\Phi}^{(r+1)}(x,\omega) &= e^{-\boldsymbol{D}(\omega)x}\boldsymbol{\Phi}(0,\omega) \\ &- \int_0^x e^{-\boldsymbol{D}(\omega)(x-\zeta)}\boldsymbol{N}(\omega)\boldsymbol{\Phi}^{(r)}(\zeta,\omega)d\zeta\end{aligned} \tag{12}$$

Note that this equation is *almost* identical to Equation (6) in [3] which we re-write here as

$$\begin{aligned}\boldsymbol{\Phi}^{(r+1)}(x,\omega) &= e^{-\boldsymbol{D}(\omega)x}\boldsymbol{\Phi}^{(r+1)}(0,\omega) \\ &- \int_0^x e^{-\boldsymbol{D}(\omega)(x-\zeta)}\boldsymbol{N}(\omega)\boldsymbol{\Phi}^{(r)}(\zeta,\omega)d\zeta\end{aligned} \tag{13}$$

The main difference between Equation (12) and Equation (13) is in the initial-condition term where it is kept at its original value in the Picard iteration (it is a real initial condition) but is updated in the waveform relaxation, using the boundary conditions of each transmission line. The value of $\boldsymbol{\Phi}(0,\omega)$ is not known a priori but depends on the near-end *and* far-end terminations of the transmission line. [1] The main implication of this difference between the initial-value problem in the Volterra iteration and the boundary-value problem of the Telegrapher's equation is that the traditional analysis of the Picard iteration and its proof of convergence to the unique solution of the Volterra integral equation does not apply to the boundary-value problem. Nonetheless, one can "recover" the initial-value problem from the boundary-value one and eliminate the dependence on $\boldsymbol{\Phi}(0,\omega)$ in (12) if instead of writing the integral equation in terms of currents and voltages as in Equation (11), we write it in terms of the *state transition matrix* of the multiconductor transmission lines. This we do in the following section.

B. Volterra Integral Equation for the State Transition Matrix

Assume that the $2N \times 2N$ matrix, $\boldsymbol{T}(x,\omega)$, is a solution of the initial-value, ordinary matrix differential equation

$$\frac{\partial}{\partial x}\boldsymbol{T}(x,\omega) = -(\boldsymbol{D}(\omega)+\boldsymbol{N}(\omega))\boldsymbol{T}(x,\omega) \tag{14}$$

with the initial condition $\boldsymbol{T}(x,\omega) = \boldsymbol{I}$, where $\boldsymbol{I}$ is the $2N \times 2N$ identity matrix. Then the $2N$ vector $\boldsymbol{\Psi}(x,\omega) = \boldsymbol{T}(x,\omega)\boldsymbol{\Psi}(0,\omega)$ is the unique solution of Equation (10) with the initial condition $\boldsymbol{\Phi}(0,\omega) = \boldsymbol{\Psi}(0,\omega)$. This initial condition is trivially satisfied because the $\boldsymbol{T}(x,\omega) = \boldsymbol{I}$ when $x = 0$. As for the differential equation (10), we have successively

$$\begin{aligned}\frac{\partial}{\partial x}\boldsymbol{\Psi}(x,\omega) &= \frac{\partial}{\partial x}\left[\boldsymbol{T}(x,\omega)\boldsymbol{\Psi}(0,\omega)\right] \\ &= \frac{\partial}{\partial x}\left[\boldsymbol{T}(x,\omega)\right]\boldsymbol{\Psi}(0,\omega) \\ &= \left[-(\boldsymbol{D}(\omega)+\boldsymbol{N}(\omega))\boldsymbol{T}(x,\omega)\right]\boldsymbol{\Psi}(0,\omega) \\ &= -(\boldsymbol{D}(\omega)+\boldsymbol{N}(\omega))\left[\boldsymbol{T}(x,\omega)\boldsymbol{\Psi}(0,\omega)\right] \\ &= -(\boldsymbol{D}(\omega)+\boldsymbol{N}(\omega))\boldsymbol{\Psi}(x,\omega)\end{aligned} \tag{15}$$

Note that the above result remains valid even when the line is nonuniform, i.e., when the line parameters are dependent on the co-ordinate x, in which case $\boldsymbol{D}(\omega)$ and $\boldsymbol{N}(\omega)$ have to be replaced with $\boldsymbol{D}(x,\omega)$ and $\boldsymbol{N}(x,\omega)$ respectively. It is also valid when the $\boldsymbol{RLGC}$ matrices are frequency-dependent due, for instance, to skin and/or or proximity effects. When the solution is computed at the far-end ports, we have $x = L$, and

$$\boldsymbol{\Psi}(L,\omega) = \boldsymbol{T}(L,\omega)\boldsymbol{\Psi}(0,\omega) \tag{16}$$

Now it is well known in the theory of linear systems [21] (Chapter 4) that the exact solution of (14) is given by an exponential matrix

$$\boldsymbol{T}(x,\omega) = e^{-[\boldsymbol{D}(\omega)+\boldsymbol{N}(\omega)]x} \tag{17}$$

so if the matrix $\boldsymbol{T}(L,\omega)$ is conformally partitioned according to the N port currents and N port voltages, we get the *chain-parameter-matrix* representation [4] of the N transmission lines

$$\begin{bmatrix}\boldsymbol{V}(L,\omega) \\ \boldsymbol{I}(L,\omega)\end{bmatrix} = \begin{bmatrix}\boldsymbol{\mathcal{A}}(L,\omega) & \boldsymbol{\mathcal{B}}(L,\omega) \\ \boldsymbol{\mathcal{C}}(L,\omega) & \boldsymbol{\mathcal{D}}(L,\omega)\end{bmatrix}\begin{bmatrix}\boldsymbol{V}(0,\omega) \\ \boldsymbol{I}(0,\omega)\end{bmatrix} \tag{18}$$

The above equation must be augmented with termination conditions. Linear, time-invariant, termination conditions are represented in the frequency domain by the equations

$$\boldsymbol{\mathcal{E}}(0,\omega)\boldsymbol{V}(0,\omega) + \boldsymbol{\mathcal{F}}(0,\omega)\boldsymbol{I}(0,\omega) = \boldsymbol{\mathcal{S}}(0,\omega) \tag{19}$$

$$\boldsymbol{\mathcal{E}}(L,\omega)\boldsymbol{V}(L,\omega) + \boldsymbol{\mathcal{F}}(L,\omega)\boldsymbol{I}(L,\omega) = \boldsymbol{\mathcal{S}}(L,\omega) \tag{20}$$

where $\boldsymbol{\mathcal{S}}(0,\omega)$ and $\boldsymbol{\mathcal{S}}(L,\omega)$ denote the frequency-domain representations of the sources at the near and far end, respectively. Note that the boundary conditions in Equations (19) and (20) are written so that both Norton and Thevenin representations of the terminations are included. At each frequency ω, the solution of the near and far end port quantities is given by the $4N \times 4N$ linear system

$$\begin{bmatrix}\boldsymbol{\mathcal{A}}(L,\omega) & \boldsymbol{\mathcal{B}}(L,\omega) & -\boldsymbol{I}_N & \boldsymbol{O} \\ \boldsymbol{\mathcal{C}}(L,\omega) & \boldsymbol{\mathcal{D}}(L,\omega) & \boldsymbol{O} & -\boldsymbol{I}_N \\ \boldsymbol{\mathcal{E}}(0,\omega) & \boldsymbol{\mathcal{F}}(0,\omega) & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{\mathcal{E}}(L,\omega) & \boldsymbol{\mathcal{F}}(L,\omega)\end{bmatrix}\begin{bmatrix}\boldsymbol{V}(0,\omega) \\ \boldsymbol{I}(0,\omega) \\ \boldsymbol{V}(L,\omega) \\ \boldsymbol{I}(L,\omega)\end{bmatrix}$$

[1] The most natural initial guess for starting the WR iteration in the transmission line case is to solve for the N transmission lines using the existing terminations with no coupling, i.e., $\boldsymbol{N}(\omega) = 0$.

$$= \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{O} \\ \boldsymbol{\mathcal{S}}(0,\omega) \\ \boldsymbol{\mathcal{S}}(L,\omega) \end{bmatrix}$$

The matrix of the above linear system is the frequency-domain stamp for the AC analysis of the N transmission lines in the presence of linear, time-invariant terminations.

From the viewpoint of the state transition matrix $\boldsymbol{T}(x,\omega)$, the main idea of TWR is to avoid the direct computation of the full exponential matrix in (17), and to compute it iteratively by treating the $\boldsymbol{N}(\omega)$ matrix as if it were a perturbation on $\boldsymbol{D}(\omega)$ whose exponential is relatively easy to compute. Equation (14) can be written as

$$\frac{\partial}{\partial x}\boldsymbol{T}(x,\omega) = -\boldsymbol{D}(\omega)\boldsymbol{T}(x,\omega) - \boldsymbol{N}(\omega)\boldsymbol{T}(x,\omega) \qquad (21)$$

where the term $\boldsymbol{N}(\omega)\boldsymbol{T}(x,\omega)$ is now considered an input excitation into a linear system whose dynamic matrix is $\boldsymbol{D}(\omega)$. Again, based on linear system theory [4] (Chapter 7), the general solution of the above equation is written as

$$\boldsymbol{T}(x,\omega) = e^{-\boldsymbol{D}(\omega)x} - \int_0^x e^{-\boldsymbol{D}(\omega)(x-\zeta)}\boldsymbol{N}(\omega)\boldsymbol{T}(\zeta,\omega)d\zeta \qquad (22)$$

IV. Convergence of TWR

The matrix integral equation (22) is a Volterra integral equation of the second kind [20]. If we denote the kernel matrix function by

$$\boldsymbol{K}(x,\omega) = -e^{-\boldsymbol{D}(\omega)x}\boldsymbol{N}(\omega) \qquad (23)$$

and the initial condition by

$$\boldsymbol{\kappa}(x,\omega) = e^{-\boldsymbol{D}(\omega)x},$$

the Volterra integral equation (22) can be rewritten more compactly as

$$\boldsymbol{T}(x,\omega) = \boldsymbol{\kappa}(x,\omega) + \boldsymbol{K} * \boldsymbol{T}(x,\omega) \qquad (24)$$

where the convolution $\boldsymbol{K} * \boldsymbol{T}$ is defined by

$$(\boldsymbol{K} * \boldsymbol{T})(x,\omega) = \int_0^x \boldsymbol{K}(x-\zeta,\omega)\boldsymbol{T}(\zeta,\omega)d\zeta \qquad (25)$$

In order to get to the essence of the TWR iteration from an analytic viewpoint, we need to introduce few notations. We denote by $\boldsymbol{\mathcal{T}}$ the function space $\boldsymbol{\mathcal{C}}(\mathbb{D}, \mathbb{C}^{2N} \times \mathbb{C}^{2N})$ of complex matrix functions that are continuous over the domain $\mathbb{D} := [0,L] \times [\omega_{min}, \omega_{max}]$. Note that $\boldsymbol{\mathcal{T}}$ is a Banach space with respect to the norm of uniform convergence. We also denote by $\boldsymbol{\mathcal{K}} : \boldsymbol{\mathcal{T}} \to \boldsymbol{\mathcal{T}}$ the operator mapping $\boldsymbol{U} \in \boldsymbol{\mathcal{T}}$ to $\boldsymbol{V} \in \boldsymbol{\mathcal{T}}$ according to

$$\boldsymbol{V} = \boldsymbol{\mathcal{K}}\boldsymbol{U} = \boldsymbol{\kappa} + \boldsymbol{K} * \boldsymbol{U} \qquad (26)$$

Then clearly Equation (24) states that $\boldsymbol{T}$ is a fixed point of the operator $\boldsymbol{\mathcal{K}}$ in the Banach space $\boldsymbol{\mathcal{T}}$. This fixed point can be found according to the following iterative process:

1) Set $\boldsymbol{T}^{(0)} = \boldsymbol{\kappa}$.

2) Update

$$\boldsymbol{T}^{(r+1)} = \boldsymbol{\kappa} + \boldsymbol{K} * \boldsymbol{T}^{(r)} \qquad (27)$$

3) Repeat (2) until convergence.

This is a Picard iteration that is guaranteed to converge if the operator $\boldsymbol{\mathcal{K}} : \boldsymbol{\mathcal{T}} \to \boldsymbol{\mathcal{T}}$ is contracting (see Chapter 5 in [22]). The objective of the rest of this section is to give a direct proof for the convergence of the recursion in (27). The proof will help us gain more insight into the TWR process and derive an upper bound on the TWR error when the process is stopped after a finite number of iterations.

First, we have the following

Lemma 4.1: The r-th TWR iterate $\boldsymbol{T}^{(r)}$ is the sum of the first $r+1$ iterated kernels

$$\boldsymbol{T}^{(r)} = \sum_{q=0}^{r} \boldsymbol{K}^q * \boldsymbol{\kappa} \qquad (28)$$

where

$$\boldsymbol{K}^q = \underbrace{\boldsymbol{K} * \boldsymbol{K} * \cdots * \boldsymbol{K}}_{q \text{ times}}$$

with the convention that $\boldsymbol{K}^0 = \boldsymbol{I}$.

Proof: By induction on r using (27). ■

The above lemma suggests that the fixed-point solution of (24) is the sum of the series in (28) as $r \to \infty$ provided that the series converge. This is indeed the case, and in order to prove it, we need first to look more closely at the iterated kernels $\boldsymbol{K}^r * \boldsymbol{\kappa}$.

We denote by $\boldsymbol{S}^{(r)} := \boldsymbol{T}^{(r)} - \boldsymbol{T}^{(r-1)} = \boldsymbol{K}^r * \boldsymbol{\kappa}$. Then we have

$$\boldsymbol{S}^{(r+1)}(x,\omega) = \int_0^x e^{-\boldsymbol{D}(\omega)(x-\zeta)}(-\boldsymbol{N}(\omega))\boldsymbol{S}^{(r)}(\zeta,\omega)d\zeta \qquad (29)$$

which is the recursive equation defining all the $\boldsymbol{S}^{(r)}(x,\omega)$ iterates. Taking the matrix norm of both sides, we get the following inequalities

$$\begin{aligned} &\|\boldsymbol{S}^{(r+1)}(x,\omega)\| \\ &= \|\int_0^x e^{-\boldsymbol{D}(\omega)(x-\zeta)}(-\boldsymbol{N}(\omega))\boldsymbol{S}^{(r)}(\zeta,\omega)d\zeta\| \\ &\le \int_0^x \|e^{-\boldsymbol{D}(\omega)(x-\zeta)}(-\boldsymbol{N}(\omega))\boldsymbol{S}^{(r)}(\zeta,\omega)\|d\zeta \\ &\le \int_0^x \|e^{-\boldsymbol{D}(\omega)(x-\zeta)}\|\|(-\boldsymbol{N}(\omega))\|\|\boldsymbol{S}^{(r)}(\zeta,\omega)\|d\zeta \\ &\le \|\boldsymbol{N}(\omega)\| \int_0^x \|e^{-\boldsymbol{D}(\omega)(x-\zeta)}\|\|\boldsymbol{S}^{(r)}(\zeta,\omega)\|d\zeta \qquad (30) \end{aligned}$$

Since the function $(x,\zeta) \to \|e^{-\boldsymbol{D}(\omega)(x-\zeta)}\|$ is continuous on the square $[0,L] \times [0,L]$ it has an upper bound that we denote by $E(\omega)$. Note also that $\|e^{-\boldsymbol{D}(\omega)x}\| \le E(\omega)$. Now we can state the following:

Lemma 4.2: The rth iterated kernel satisfies the upper bound

$$\|\boldsymbol{S}^{(r)}(x,\omega)\| \le \frac{x^r}{r!}\|\boldsymbol{N}(\omega)\|^r \left[E(\omega)\right]^{r+1}, \ \ \forall (x,\omega) \in \mathbb{D}. \qquad (31)$$

Proof: The proof is by induction on r. First for $r = 1$, we have

$$\boldsymbol{S}^{(1)}(x,\omega) = \int_0^x e^{-\boldsymbol{D}(\omega)(x-\zeta)}(-\boldsymbol{N}(\omega))e^{-\boldsymbol{D}(\omega)\zeta}d\zeta.$$

Taking the matrix norm of both sides and following the same series of norm inequalities in (30), we get

$$\begin{aligned} &\|\boldsymbol{S}^{(1)}(x,\omega)\| \\ &\le \|\boldsymbol{N}(\omega)\| \int_0^x \|e^{-\boldsymbol{D}(\omega)(x-\zeta)}\| \|e^{-\boldsymbol{D}(\omega)(\zeta)}\| d\zeta \\ &\le \|\boldsymbol{N}(\omega)\| [E(\omega)]^2 \int_0^x d\zeta = x\|\boldsymbol{N}(\omega)\| [E(\omega)]^2 \qquad (32) \end{aligned}$$

which proves inequality (31) for $r = 1$. Assume now the inequality is satisfied for all $q \le r$, and let us prove that its satisfied for $q = r + 1$. From the line before Inequality (30) we get

$$\|\boldsymbol{S}^{(r+1)}(x,\omega)\| \le \|\boldsymbol{N}(\omega)\| E(\omega) \int_0^x \|\boldsymbol{S}^{(r)}(\zeta,\omega)\|$$

which by applying the induction assumption to the integrand gives

$$\begin{aligned} &\|\boldsymbol{S}^{(r+1)}(x,\omega)\| \\ &\le \|\boldsymbol{N}(\omega)\| E(\omega) \|\boldsymbol{N}(\omega)\|^r [E(\omega)]^{r+1} \int_0^x \frac{\zeta^r}{r!} d\zeta \\ &= \frac{x^{r+1}}{(r+1)!} \|\boldsymbol{N}(\omega)\|^{r+1} [E(\omega)]^{r+2} \qquad (33) \end{aligned}$$

which completes the proof. ■

Corollary 4.3: For each $\omega \in [\omega_{min}, \omega_{max}]$, the r-th iterated kernel is uniformly bounded on $[0, L]$ as follows

$$\begin{aligned} \|\boldsymbol{K}^r * \boldsymbol{\kappa}(x,\omega)\| &= \|\boldsymbol{S}^{(r)}(x,\omega)\| \\ &\le \frac{L^r}{r!} \|\boldsymbol{N}(\omega)\|^r [E(\omega)]^{r+1} \\ &\forall (x,\omega) \in \mathbb{D} \qquad (34) \end{aligned}$$

With this upper bound (34), we can now prove the following

Theorem 4.4: For each $\omega \in [\omega_{min}, \omega_{max}]$ the series of iterated TWR kernels

$$\boldsymbol{T}^{(r)}(x,\omega) = \sum_{k=0}^{r} \boldsymbol{K}^k * \boldsymbol{\kappa}(x,\omega) \qquad (35)$$

converges uniformly on the interval $[0, L]$ to the unique solution $\boldsymbol{T}(x,\omega)$ of (24) as $r \to \infty$.

Proof: According to (34), for each $\omega \in [\omega_{min}, \omega_{max}]$, every term in the series (35) is uniformly (with respect to $x \in [0, L]$) dominated by the upper bound

$$E(\omega)\frac{L^r}{r!} \|\boldsymbol{N}(\omega)\|^r [E(\omega)]^r$$

Further, the series

$$\sum_{r=0}^{\infty} \frac{1}{r!} [LE(\omega)\|\boldsymbol{N}(\omega)\|]^r$$

converges to

$$\exp\left(L\|\boldsymbol{N}(\omega)\|E(\omega)\right).$$

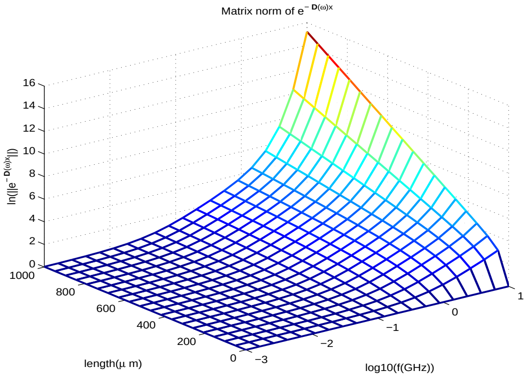

Fig. 3. Surface plot showing the dependence of the matrix norm of the Volterra kernel $e^{-\boldsymbol{D}(\omega)x}$ on line length and frequency.

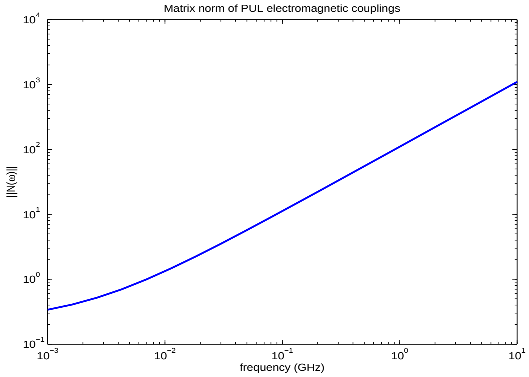

Fig. 4. Dependence of the norm of the electromagnetic coupling matrix on frequency.

■

In Figures 3 and 4, typical plots of the functions $(x,\omega) \to \|\exp(-\boldsymbol{D}(\omega)x)\|$ and $\omega \to \|\boldsymbol{N}(\omega)\|$ are shown. The corresponding PUL $\boldsymbol{R}, \boldsymbol{L}, \boldsymbol{G}, \boldsymbol{C}$ matrices are those of an industrial 24-bitline on-chip bus. The matrix norm used in these plots is the maximum-singular-value norm induced by the Euclidean norm on $\mathbb{C}^{2N}$. The function $\omega \to E(\omega)$ used in the above inequalities is the curve traced by the intersection of the surface in Figure 3 with the $x = 1000\mu m$ plane. It is clear from these plots that since $E(\omega)$ and $\|\boldsymbol{N}(\omega)\|$ are monotonically increasing functions of ω, the TWR convergence bounds are worse the higher the frequency and the longer the line. Furthermore since $L\|\boldsymbol{N}(\omega)\|$ measures the total cumulative electromagnetic coupling along the line, the TWR iterates are smaller the weaker the electromagnetic coupling between the lines.

Finally, one can state the TWR convergence result in terms of the TWR function operators as follows

Theorem 4.5: In the normed space $\boldsymbol{\mathcal{C}}(\mathbb{D}, \mathbb{C}^{2N} \times \mathbb{C}^{2N})$ of continuous matrix functions on $\mathbb{D} := [0, L] \times [\omega_{min}, \omega_{max}]$ equipped with the uniform-convergence norm, the iteration

$$\boldsymbol{T}^{(r+1)} = \boldsymbol{\kappa} + \boldsymbol{K} * \boldsymbol{T}^{(r)}$$

always converges to the unique solution of the Volterra integral equation of the second kind

$$\boldsymbol{T} = \boldsymbol{\kappa} + \boldsymbol{K} * \boldsymbol{T}$$

Proof: Use the maximum values $\max_{\omega}(E(\omega))$ and $\max_{\omega}(||\boldsymbol{N}(\omega||)$ with $\omega \in [\omega_{min}, \omega_{max}]$ to insure that the series of iterated kernel in (35) converges absolutely on $\mathbb{D}$. ■

In the next section, the series of iterated kernels in (35) is used to derive an error estimate for the TWR algorithm..

V. Error Analysis of TWR

The results of the previous section allows us to conduct a rigorous error analysis of the TWR algorithm. When applied to multiconductor transmission lines, the residual error $\boldsymbol{R}^{(n)}(x,\omega)$ after n iterations is given by

$$\begin{aligned} \boldsymbol{R}^{(n)}(x,\omega) &= \boldsymbol{T}(x,\omega) - \sum_{k=0}^{n} \boldsymbol{S}^{(k)}(x,\omega) \\ &= \sum_{k=0}^{\infty} \boldsymbol{S}^{(k)}(x,\omega) - \sum_{k=0}^{n} \boldsymbol{S}^{(k)}(x,\omega) \\ &= \sum_{k=n+1}^{\infty} \boldsymbol{S}^{(k)}(x,\omega) \qquad (36) \end{aligned}$$

Taking the matrix norms of both sides and using the upper bound of (34) we get the following upper bound on the residual error

$$\begin{aligned} ||\boldsymbol{R}^{(n)}(x,\omega)|| &\leq \sum_{k=n+1}^{\infty} \frac{L^k}{k!} ||\boldsymbol{N}(\omega)||^k \left[E(\omega)\right]^{k+1} \\ &= \frac{L^{n+1}}{(n+1)!} ||\boldsymbol{N}(\omega)||^{n+1} \left[E(\omega)\right]^{n+2} \\ &\times \sum_{k=0}^{\infty} \frac{L^k}{k!} ||\boldsymbol{N}(\omega)||^k \left[E(\omega)\right]^k \\ &= \frac{L^{n+1}}{(n+1)!} ||\boldsymbol{N}(\omega)||^{n+1} \left[E(\omega)\right]^{n+2} \\ &\times \exp\left(L||\boldsymbol{N}(\omega)||E(\omega)\right) \qquad (37) \end{aligned}$$

The exponential in the final inequality is independent of the iteration n and is in fact an upper bound on the matrix norm $||\boldsymbol{T}(x,\omega)||$. An estimate of the maximum relative error $\rho_n(\omega)$ of the WR approximation of the multiconductor transmission lines can be given by

$$\begin{aligned} &\rho_n(\omega) = \\ &\quad \exp\left(-L||\boldsymbol{N}(\omega)||E(\omega)\right) \max_{x\in[0,L]} ||\boldsymbol{R}^{(n)}(x,\omega)|| \leq \\ &\quad \frac{L^{n+1}}{(n+1)!} ||\boldsymbol{N}(\omega)||^{n+1} \left[E(\omega)\right]^{n+2} \qquad (38) \end{aligned}$$

The matrix norm $||\boldsymbol{N}(\omega)||$ measures the size of the coupling terms in the PUL matrices. This relative error estimate allows us to predict that the weaker the couplings the faster the convergence. Furthermore, the longer the line, the slower the convergence. These predictions are borne out by the recent computational evidence described in [3] and in Section VII of this paper.

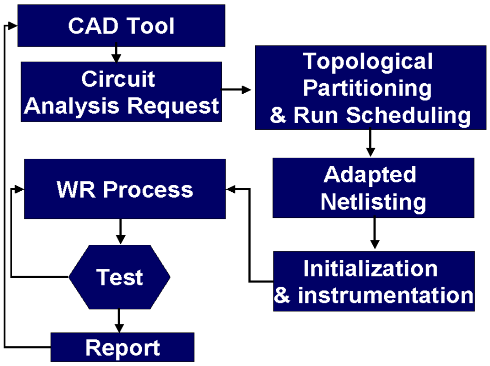
CAD Tool
Circuit Analysis Request
Topological Partitioning & Run Scheduling
WR Process
Adapted Netlisting
Test
Initialization & instrumentation
Report

Fig. 5. Methodology flow of an embedded WR algorithm.

Discussion: Based on the above convergence analysis of the Gauss-Jacobi TWR algorithm, it is clear that one can estimate the number of iterations needed to achieve a given level of accuracy. This number is smaller the smaller the total coupling between the lines as measured by the norm $||\boldsymbol{N}(\omega)L||$. When some of the lines exhibit strong coupling the splitting of the Telegrapher's equation can be adapted so as to be *block diagonal* rather than diagonal. Each diagonal block contains the PUL submatrices of a given line and the lines that are strongly coupled to it. These diagonal blocks will affect the norm $E(\omega)$ and will very likely make it larger. The PUL matrix $\boldsymbol{N}(\omega)$ representing the weak couplings is likely to have a smaller norm. The essence of the Volterra integral equation analysis and series convergence will however remain the same, and the conclusion as to the convergence of the Volterra series to the state-transition matrix remains unchanged. It is also to be noted that the same framework can be used to analyze the convergence of the Gauss-Seidel TWR as well as the SOR TWR. In a future paper, we will show how this framework can be augmented to include linear near-end and far-end terminations. The importance of including terminations in the convergence analysis stems from the intuitive fact that coupled transmission lines with multiple reflections due to load and source impedance mismatches will have more difficulty converging than perfectly matched lines.

VI. Embedded Implementation of TWR

The practical implementation of any WR process for the analysis of electrical circuits, be they of lumped or distributed nature, requires the following steps:

1) Partitioning of the original circuit into small subcircuits.
2) Instrumentation of each subcircuit with sources (for inputs) and probes (for outputs).
3) Scheduling the different subcircuits for visitation by the WR process.
4) Initiating, monitoring, and stopping the WR iteration.
5) Management, storage, and update of probe and source waveforms from one iteration to another.

Depending on the nature of the analyzed circuit, there may be other steps that the WR process needs to guarantee speed, accuracy, or numerical stability. Among these steps:

1) Windowing the simulation time interval (WR converges faster on “short” time windows).
2) Management of waveform hand-offs from one time window to another.
3) “Windowing” the length of the transmission line (TWR converges faster on “short” lines. See convergence analysis in Section IV).
4) Organization of the multi-rate analysis of subcircuits with vastly differing time constants.
5) Management of waveforms at the boundaries between subcircuits analyzed with different rates.

All these steps are important and can have a significant impact on the rate of convergence of the WR process. Borrowing an expression from [23], they can all be assembled under the heading of *external WR algorithm*.

Previous implementations of the WR algorithm for the analysis of electrical circuits [1], [24]–[26] were built on the premise that the software implementation of *external WR* should be within the core engine of the SPICE-level circuit analysis program. This implementation philosophy resulted in placing too much burden on the circuit simulator which had to be augmented with sophisticated netlist partitioning and subcircuit scheduling algorithms as well as a full database and a complete library for waveform management and processing.

In this paper we take an “external” perspective on the implementation of WR in the following sense. Our TWR prototype assumes that the implementation of the *external WR algorithm* should be external to the core engine of the SPICE-level circuit analysis program and should rather be owned by the CAD program requesting the WR analysis. Figure 5 illustrates such a flow where the WR process is under the management of the CAD tool be it one for timing, signal integrity, or power analysis. We call such a flow an *embedded WR algorithm*. To insure seamless integration of the circuit analysis step in WR with the CAD tool requesting the WR analysis, the SPICE-level circuit analysis program should have an application programming interface (API) [27] that enables the in-core setup, instrumentation, and probing of the subcircuits as well as the setting of all the run control parameters needed for the electrical analysis at the subcircuit level.

The premise of this API-based, embedded implementation of WR is that the intimate knowledge that the CAD tool has of the application domain and the origins of circuit netlists will enable the use of customized partitioning and scheduling algorithms. Furthermore, if the API interface is standardized accross industrial circuit analysis programs such as HSPICE (Synopsis), Spectre (Cadence), PowerSpice (IBM) or Nexxim (Ansoft), the CAD tool requesting the WR analysis and the circuit simulator will be plug-compatible. As shown in Figure 5, not only does the embedded WR approach move the external WR from the core circuit simulation engine to the CAD tool but it further transfers out the WR convergence test and termination themselves. Here again, knowledge of the application domain plays a crucial role in setting reasonable convergence criteria that will minimize the number of WR iterations needed to get acceptable waveforms.

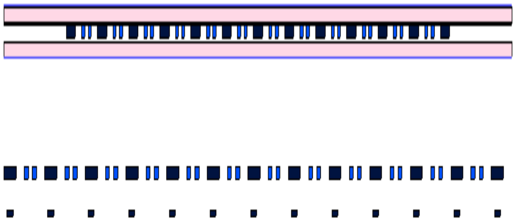

Fig. 6. Cross section of the 24-bitline bus. The top geometry was used for capacitance extraction. The bottom geometry was used for inductance extraction.

Finally, in the specific context of this work, the limited resources in time and headcount as well as a corporate need for the rapid prototyping of TWR have made the API-based embedded approach very attractive. For one thing, it keeps the SPICE-level core engine undisturbed while enabling the TWR R & D effort to proceed. For another, it allows the R & D effort to remain focused on the specific CAD needs, namely, signal integrity for very wide on-chip, global buses rather than the generic issues of a new, full WR-based circuits simulation program.

Beside the SIPCE-level API, the TWR prototype used the *C++ Standard Template Library (STL)*, especially the *map* datastracture for the management of probe and source waveforms as well as an internal *C++* library for waveform processing [28] supporting such functions as windowing, squishing, filtering, etc.

The TWR prototype, code-named *TWRTL*, is written in C++, has about 2000 lines of code, and runs on an IBM PowerPC supporting the AIX5.3 operating system.

VII. Numerical Results

A. Description of the Test Suite

The test suite used in our numerical experiments comprise on-chip buses with number of transmission lines increasing from 4 to 40 bitlines with an increment of 4 bitlines. A bus having a pair of coupled transmission lines, the smallest topology to which TWR can be applied, is also included in the test suite. Every bus was assumed to run on the next to last top metal layer of a 10-metal-layer BEOL cross section corresponding to the 65nm technology node. The top two metal layers of this cross section are 8X layers that are used for global on-chip signal and clock interconnection. In the global buses of the test suite, the signal wire width is $0.8\mu m$ on a pitch of $1.6\mu m$ while the power wire width is $2.4\mu m$ on a pitch of $8.0\mu m$. The structure of the power bus alternates the

VDD and the VSS (ground) wires. In each power bay between a VDD and a VSS, two signal wires are routed at minimum spacing with the net outcome that the distance between a signal wire and its neighboring power wire was $1.60\mu m$. The $\boldsymbol{R}, \boldsymbol{L}, \boldsymbol{G}, \boldsymbol{C}$ matrices of each bus are computed according to the methodology flow described in [29] as implemented in IBM's AQUAIA tool.

The top cross section of Figure 6 shows an example of the geometry used to extract the PUL capacitance matrix, $\boldsymbol{C}$, of a 24-line bus while the bottom cross section shows the geometry used to extract the PUL inductance matrix $\boldsymbol{L}$. One important aspect of TWR is the impact of electromagnetic couplings on convergence. This impact was expressed mathematically in an aggregate form through the matrix $\boldsymbol{N}(\omega)$ and its norm $\|\boldsymbol{N}(\omega)\|$, see Equation (34) and Figure 4. Perhaps a more visual way for capturing the electromagnetic couplings in a bus is to plot the bar graph of the capacitive and inductive coupling coefficients defined as follows:

$$\text{Capacitive: } K_{ij}^{cap} = \frac{c_{ij}}{c_{ii}}, \; 1 \leq i, j \leq N \tag{39}$$

$$\text{Inductive: } K_{ij}^{ind} = \frac{l_{ij}}{l_{ii}}, \; 1 \leq i, j \leq N \tag{40}$$

where $\boldsymbol{C} = [c_{ij}]$ and $\boldsymbol{L} = [L_{ij}]$. In Figure 7 such a bar graph is shown for a 24-line bus in the case when $i = 12$. Note in the top graph how most of the electromagnetic coupling is due to the nearest neighbor line, $j = 11$, that is sharing the same power bay with line $i = 12$. The capacitive and inductive coupling coefficients of this neighbor are 40% and 50%, respectively. The bottom graph is a zoom-in on the region where the coupling coefficient is below 2.5%. The presence of the power bus provides effective capacitive shielding and almost eliminates all the capacitive couplings due to lines that are in adjacent power bays, including the nearest neighbor line $j = 13$. On the other hand, the inductive coupling is not eliminated but decreases gracefully with the distance between the center line $i = 12$ and its right and left neighbors. Finally, note that the inductive coupling coefficients are consistently larger than the capacitive coupling coefficients, indicating that the induced crosstalk noise in the bus is essentially inductive. This situation is typical and should be sufficient to justify the use of inductance-aware and/or transmission line macromodels for global on-chip interconnect.

We note finally that for all these on-chip buses $\boldsymbol{R}$ is a diagonal matrix (in fact, a scalar multiple of the identity matrix) and $\boldsymbol{G} = \boldsymbol{0}$.

Each line in the bus was terminated with a 50Ω resistor on the near end and a $1pF$ capacitor on the far end. The active signal was a trapezoidal waveform with a rise/fall time of $40ps$ between $0V$ and $1V$. The total bus length is $1000\mu m$.

The numerical experiments were all crosstalk experiments in which only the center bitline was active while all other bitlines were quiet. Specifically, in a bus of width $N = 2N_1$ lines, we assumed that line $i = N_1$ was active, all the other lines remaining quiet.

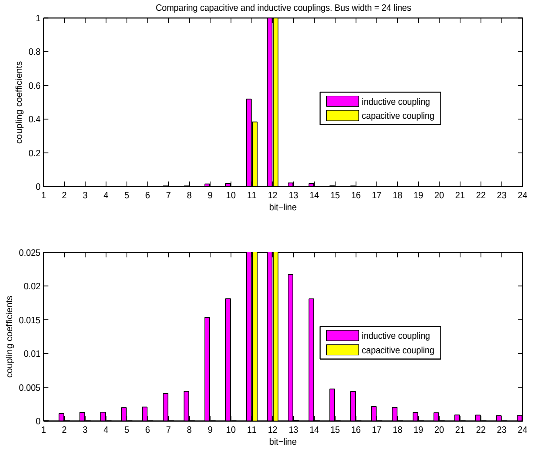

Fig. 7. Capacitive and inductive coupling coefficients for a 24-bitline bus. The reference line is line 12 assumed active.

B. Validation

In this set of results, the question of "convergence to the right solution" is addressed. TWR has been extensively validated against a SPICE-level simulator, namely, PowerSpice (the IBM circuit simulation program). On each bus both TWRTL and PowerSpice were run. Within TWRTL each transmission line was analyzed as a single line excited with distributed sources whose states are updated according to the TWR algorithm. The waveforms of these sources are computed according to Equations (5) and (6) and communicated to PowerSpice using the API interface. The process of decomposing the bus into single transmission lines, scheduling the TWR iterations, probing currents and voltages, and updating the relaxation sources is done in-core (i.e. no file IO) but outside the PowerSpice core engine whose sole responsibility is to analyze the single lines. The run controls of both TWRTL and PowerSpice were exactly the same.

Given the efficiency limitation of PowerSpice in analyzing buses containing more than 10 transmission lines, a lumped RLC circuit model has been used. For the PowerSpice case, each cell in the lumped model is as shown on the top of Figure 2. Because the full electromagnetic couplings are taken into account in the PowerSpice bus models, the analysis is called "full analysis." As for the TWR analysis in TWRTL, the lumped model is as shown on the bottom of Figure 2. The number of cells for both the PowerSpice and TWR analysis was taken to be 10 cells per each mm of transmission line length.

For each bus, the difference between the full analysis and the TWR analysis was measured as follows:

$$\begin{aligned}
\epsilon_k^{ABS}(x) &= \max_{0 \leq t \leq T} |v_k^{SPICE}(x,t) - v_k^{TWR}(x,t)| \\
\epsilon_k^{REL}(x) &= \frac{\max\limits_{0 \leq t \leq T} |v_k^{SPICE}(x,t) - v_k^{TWR}(x,t)|}{\max\limits_{0 \leq t \leq T} |v_k^{SPICE}(x,t)|} \\
& 1 \leq k \leq N
\end{aligned}$$

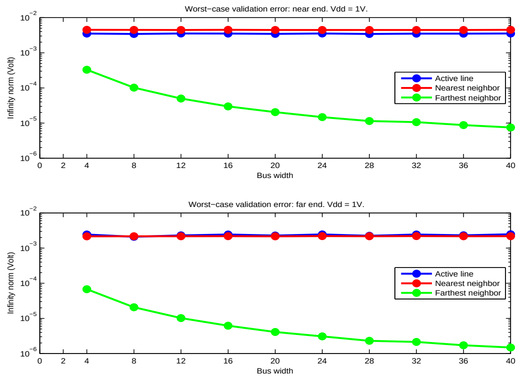

Fig. 8. Worst-case TWR error with respect to full analysis.

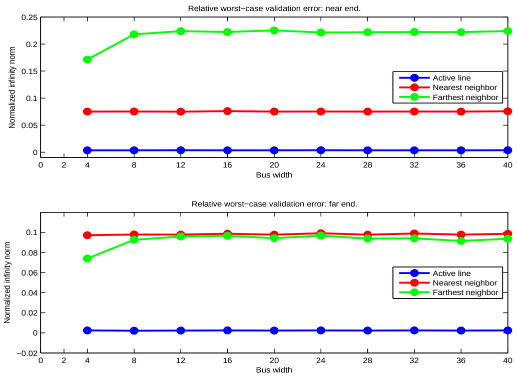

Fig. 9. Relative TWR error with respect to full analysis.

where $\epsilon_k^{ABS}(x)$ and $\epsilon_k^{REL}(x)$ measure, respectively, the worst-case absolute and worst-case relative errors at point x of the k-th line of the bus. Note that $\epsilon_k^{ABS}(0)$ and $\epsilon_k^{ABS}(L)$ are, respectively, the aboslute errors at the near and far end of the k-th line, while $\epsilon_k^{REL}(0)$ and $\epsilon_k^{REL}(L)$ are the corresponding relative errors. For all the buses, the TWR was run to convergence (see next subsection). In Figures 8 and 9, representative error values for the near and far ends are given as functions of bus width. Each plot has three curves corresponding to three values of k: the active line, the nearest-neigbor line and the farthest-neighbor line. It is remarkable that for this series of tests, the quality of the TWR convergence to the PowerSpice solution is independent of the problem size as measured by the bus width. Furthermore, the relative error at the near end is consistently worse than that at the far end. This suggests that a conservative relative convergence criterion should be based on the near-end waveforms.

C. Convergence

In this set of results, the convergence of the TWR process for the on-chip bus test suite is illustrated. For each bus, TWR was run for 4 iterations with the initial iteration having zero initial condition on the relaxation sources. The absolute and relative convergence errors at iteration r for each bus are defined as follows:

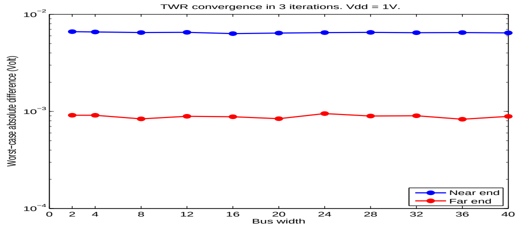

Fig. 10. Worst-case difference between 3rd and 4th TWR iterations.

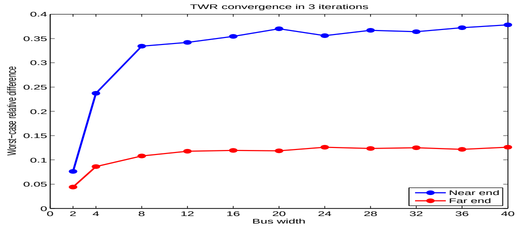

Fig. 11. Worst-case relative difference between 3rd and 4th TWR iterations.

$$\epsilon_r^{ABS}(x) = \max_{1\le k\le N} \max_{0\le t\le T} |v_k^{(r)}(x,t) - v_k^{(r-1}(x,t)|$$

$$\epsilon_r^{REL}(x) = \max_{1\le k\le N} \frac{\max_{0\le t\le T} |v_k^{(r)}(x,t) - v_k^{(r-1)}(x,t)|}{\max_{0\le t\le T} |v_k^{(r)}(x,t)|}$$

The plots of $\epsilon_4^{ABS}(0)$ (near end) and $\epsilon_4^{ABS}(L)$ (far end) vs. bus width are given in Figure 10. The absolute convergence errors are remarkably stable with respect to the bus width. An essentially similar behavior is displayed by the relative convergence errors in Figure 11 where $\epsilon_r^{REL}(0)$ (near end) and $\epsilon_4^{REL}(L)$ (far end) are plotted. As was remarked for the validation results, the near end has consistently worse absolute and relative errors than the far end. This again suggests that the near-end waveforms provide us with the more conservative convergence criterion.

In order to assess the TWR convergence rate, the plots of $\epsilon_r^{ABS}(0)$ (near end) and $\epsilon_r^{ABS}(L)$ (far end) vs. iteration r are given in Figure 12 for a selected number of bus widths: 12, 24, 32 and 40 bitlines. Note that the y-axis on ths plot is logarithmic. There is about a decade of improvement in waveform accuracy on the first three iterations with the far end waveform having a faster convergence rate than the near end waveform. Furthermore, there is no appreciable difference in the rates across the test suite: the plots for all the selected buses land on the top of each other. Similar conclusions can be made about Figure 13 showing the relative convergence errors: $\epsilon_r^{REL}(0)$ at the near end and $\epsilon_r^{REL}(L)$ at the far far end. Note that because of the zero initial condition on the relaxation sources, we will always have $\epsilon_1^{REL}(0) = \epsilon_1^{REL}(L) = 1$ as shown.

The voltage waveforms that result from the first few iterations have the required engineering accuracy needed to conduct on-chip bus signal integrity analysis as we will ilusrtate in the following two subsections.

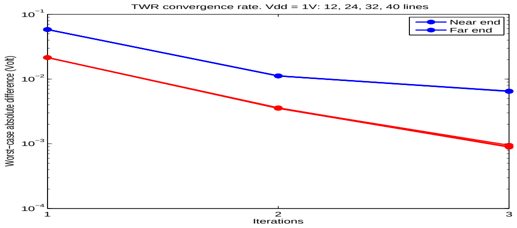

Fig. 12. Worst-case difference showing rate of convergence for 4 wide on-chip transmission line structures. Note that the convergence behavior is essentially independent of the number of lines.

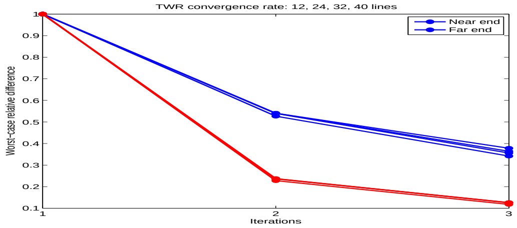

Fig. 13. Worst-case difference showing rate of convergence for 4 wide on-chip transmission line structures. Note that the convergence behavior is essentially independent of the number of lines.

D. Waveform example

In Figures 14, 15, and 16, we show a representative sample of the voltage waveforms for the 40-bitline on-chip bus of the test suite. On each plot four waveforms are shown corresponding to the fully-coupled PowerSpice analysis and the first three TWR iterations. There are several observations that can be made about these plots. First, the TWR waveforms have indeed converged to the PowerSpice ones in the first few iterations. The active line $k = 20$, Figure 14, displays the fastest convergence as the 1st TWR waveform is already very close to the PowerSpice one. The crosstalk waveforms need at least two iterations as the initial iteration results in an identically zero waveform for the crosstalk because of the zero initial condition on the relaxation sources. Recall that in TWR the relaxation sources are the ones responsible for the transmission of electromagnetic couplings from one line to another. The second observation is that TWR is capable of capturing the crosstalk in the farthest line in the bus within very few Gauss-Jacobi iterations as is clear from Figure 16 where a crosstalk level of few tens of microvolts is obtained within essentially two iterations. The third observation is that although in the validation and convergence results we ascertained that the far-end waveforms have smaller errors and faster convergence, the actual waveforms obtained from TWR are all within engineering accuracy for both the near end and far end waveforms as is shown in the 40-bitline plots.

E. Application: Noise analysis

This set of results illustrates even further the quality of the TWR waveforms compared with the PowerSpice ones. The driving input on the active line in these experiments was the rising half of the trapezoidal input of the previous experiements. The crosstalk in a 24-bitline on-chip bus was

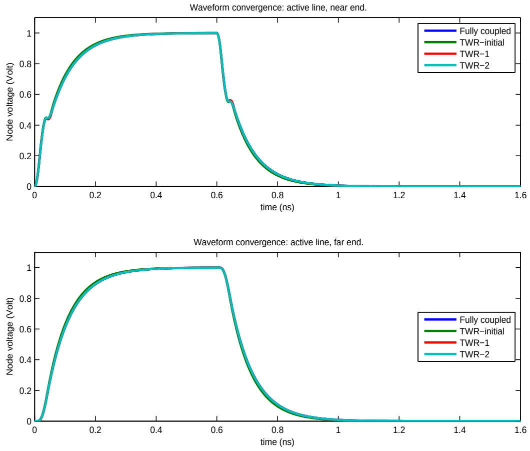

Fig. 14. A 40-line bus waveform example. Near end and far end of the active line. Active lines have typically the fastest TWR convergence.

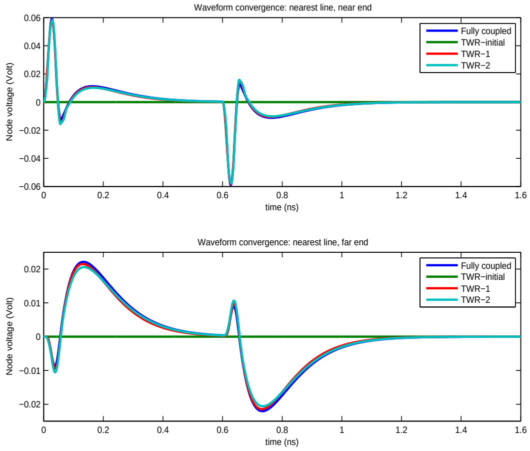

Fig. 15. A 40-line bus waveform example. Near end and far end of the nearest neighbor to the active line. This neighbor shares the same power bay as the active line, and so capacitive coupling is maximized.

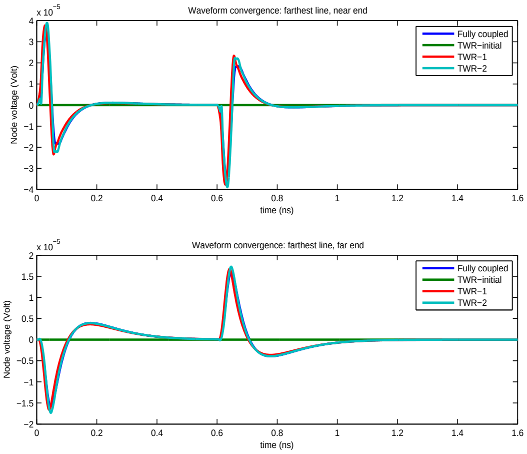

Fig. 16. A 40-line bus waveform example. Near end and far end of the rightmost line in the bus. Note how TWR is capturing the essentially inductive crosstalk waveform.

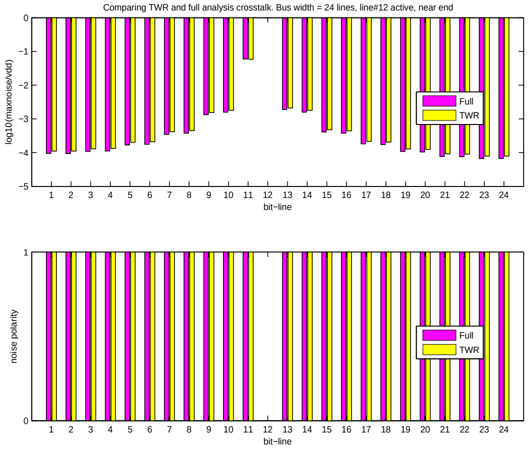

Fig. 17. Worst-case noise and its polarity for all the near ends on the victim lines. Note the orders-of-magnitude difference between the nearest and farthest neighbors on this 24-bitline bus.

analyzed under the assumption that line 12 was active with all the other lines remaining quiet. Figures 17 and 18 show that TWR is faithfully capturing both the polarity and the amplitude of the worst-case noise on *every* single quiet line on the bus. This occurs within 3 Gauss-Jacobi iterations for both the near-end and far-end crosstalk. For example, the bottom bar graph in Figure 18 shows the polarities of the worst-case noise as calculated by both PowerSpice and TWRTL. The polarities are identical. It is worth noting that the worst-case noise is of capacitive nature (positive polarity) for the nearest-neighbor of the active line, line 19, that shares the same power bay with it. The capacitve shielding due to the power grid makes the worst-case noise on the remaining quiet lines of inductive nature (negative polarity). In order to clearly show the relathionships between the PowerSpice worst-case noise and the TWR one, the top bar graph displays their worst-case amplitudes side by side. Because of the differences in crosstalk scale across the bitlines from the nearest neighbor (10^{-1}) to the farthest neighbor (10^{-4}), the amplitude bar graph was in fact plotted in terms of the decimal logarithm of the worst-case noise amplitude divided by the full voltage swing of the active signal (=1Volt).

These signal integrity experiments indicate that TWR can very well be used to find the worst-case switching patterns needed for simultaneous switching and common-mode noise analysis [8], [9].

F. Run times

The TWR run times is essentially linear as function of the bus width. This is illustrated in Figure 19 in which the y-axis displays the TWR run time for the buses of the test suite normalized with respect to the run time of the 2-line bus - the smallest bus topology to which TWR applies. This figure confirms the finding of [3] and is to be compared with Figure 5 therein.

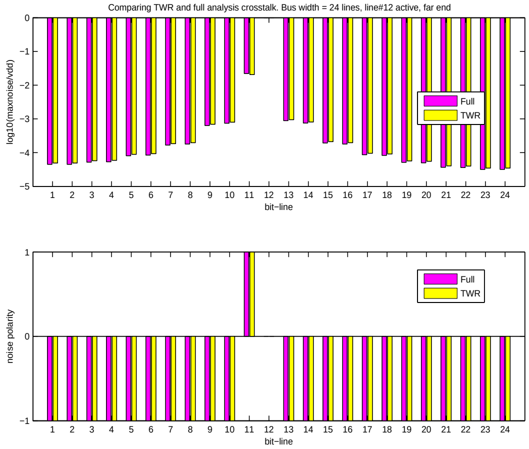

Fig. 18. Worst-case noise and its polarity for all the the far ends on the victim lines. Note the orders-of-magnitude difference between the nearest and farthest neighbors on this 24-bitline bus.

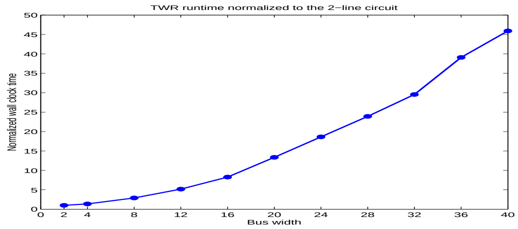

Fig. 19. Normalized TWR run time. The normalization is with respect to the 2-line TWR case.

The more interesting result in our own embedded implementation of TWR is shown in Figure 20 where the cost of updating the relaxation sources from one iteration to another appears to be *the single* major contributor to the TWR computational cost. Since the number of WR iterations needed to achieve engineering precision for the on-chip bus application is small, the WR acceleration techniques studied in [15], [17], [18], [30] might not be needed in this specific context. On the other hand, algorithms for making the relaxation source update more time-efficient are needed. We are currently exploring several techniques for speeding up the relaxation source update in TWR.

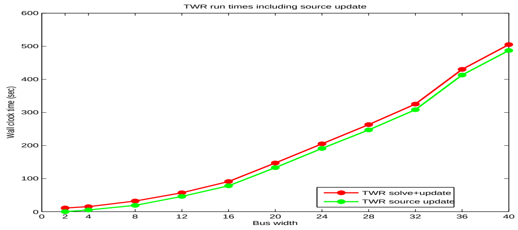

Fig. 20. Computational cost breakdown of TWR. Note that most of the time taken by TWR is due to relaxation source updates from one iteration to the next.

VIII. Conclusions and Future Work

This paper has initiated the formal analysis of convergence of the transverse waveform relaxation (TWR) method for the analysis of very wide, multiconductor transmission line systems. Significant notational simplicity has been achieved in the analysis using a splitting framework for the per-unit-length matrix parameters of the transmission lines. While our proofs, software implementation and numerical examples have focused on the Gauss-Jacobi version of TWR, they can be readily extended to the Gauss-Seidel and SOR versions. The same framework can also be used to analyze the convergence of block TWR, where lines with strong couplings are clustered together and analyzed as a single multiconductor subsystem, as well as overlapped TWR, where clusters of lines are allowed to share some conductors. One practical outcome of our work is the seamless integration of TWR within a signal integrity CAD tool using the API of an industrial circuit simulator. Another outcome is the fact that for on-chip buses, the main TWR issue is not the number of TWR iterations until convergence but rather the cost of relaxation source update. TWR acceleration will come mainly from devising new schemes for efficient source update. Standard issues in WR like circuit decomposition, time windowing, scheduling, and multirate analysis should be re-visited in the TWR context. New issues specific to TWR remain to be fully investigated, including the impact of nonlinear drivers and receivers on convergence, the role of series and dielectric losses in stabilizing TWR convergence, and the interplay between windowing very long time intervals and segmenting very long buses as might be the case in very-wide, off-chip communication links.

Acknowledgments

I would like to thank the following colleagues for many fruitful discussions: A. Ruehli, T. Maestri, R. Kimmel, W. Wright, H.-M. Huang, N. Nakhla, M. Nakhla, A. Deutsch, D. Kaller, H. Smith, G. Gristede, P. Feldmann, C. Visweswariah, and all members of the circuit and interconnect seminar at the IBM T. J. Watson Research Center. I would also like to acknowledge A. Ruehli for his close reading and enlightening comments on an early version of this manuscript.

References

[1] 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*, CAD-1(3):131–145, July 1982.

[2] A. Ruehli, P. Feldmann, A. Deutsch, H. Smith. *System and Method for the Efficient Analysis of Transmission Lines*. US Patent and Trademarks Office, US7006931, 2006.

[3] N. Nahkla, A. Ruehli, M. Nahkla, R. Achar. Simulation of coupled interconnects using waveform relaxation and transverse partitioning. *IEEE Transactions on Advanced Packaging*, 29(1):78–87, 2006.

[4] C. R. Paul. *Analysis of Multiconductor Transmission Lines*. John Wiley and Sons, 1994.

[5] D. B. Kuznetsov and J. E. Schutt-Ainé. Optimal transient simulation of transmission lines. *IEEE Transactions on Circuits and Systems, I*, 43(2):110–121, February 1996.

[6] A. J. Gruodis and C. S. Chang. Coupled lossy transmission line characterization and simulation. *IBM Journal of Research and Development*, 25(1):25–41, January 1981.

[7] A. Deutsch, G. V. Kopcsay, C. W. Surovic, B. J. Rubin, R. P. Dunne, Jr., T. Gallo, R. H. Dennard. Modeling and characterization of long on-chip interconnects for high-performance microporcessors. *IBM Journal of Research and Development*, 39(5):547–567, September 1995.

[8] A. Deutsch, P. W. Coteus, G. V. Kopcsay, H. Smith, C. W. Surovic, B. L. Krauter, D. C. Edelstein, P. J. Restle. On-chip wiring design challenges for GHz operation. *Proceedings of the IEEE*, 98(4):529–555, April 2001.

[9] A. Deutsch, H. Smith, B. Rubin, B. Krauter, G. Kopcsay. New methodology for combined simulation of delta-I noise intercation with interconnect noise for wide, on-chip buses using transmission-line power-blocks. *IEEE Transactions on Advanced Packaging*, 29(1):11–20, February 2006.

[10] N. Nahkla, A. Ruehli, M. Nahkla, R. Achar. Simulation of coupled interconnects using waveform relaxation and transverse partitioning. In *Digest of Electr. Perf. Electronic Packaging*, volume 13, pages 25–28, Portland,OR, October 2004.

[11] N. Nahkla, A. Ruehli, C. Chen, M. Nahkla, R. Achar. Waveform relaxation techniques for simulation of coupled interconnects with frequency-dependent parameters. In *Digest of Electr. Perf. Electronic Packaging*, volume 14, pages 47–50, Austin,TX, October 2005.

[12] N. Nakhla. Private communication. Technical report, Carleton University, March 2006.

[13] R. Varga. *Matrix Iterative Analysis*. Prentice Hall, 1962.

[14] O. Nevanlinna, F. Odeh. Remarks on the convergence of waveform relaxation methods. *Numerical Functional Analysis and Optimization*, 9(3&4):435–445, 1987.

[15] U. Miekkala, O. Nevanlina. Convergence of dynamic iteration methods for initial value problems. *SIAM Journal of Scientific Computation*, 8(4):459–482, 1987.

[16] G. Gristede, A. Ruehli, and C. Zukowski. Convergence properties of waveform relaxation circuit simulation methods. *IEEE Transactions on Circuits and Systems, I*, 45(7):337–348, July 1998.

[17] M. Reichelt, J. K. White, J. Allen. Optimal convolution SOR acceleration of waveform relaxation with application to parallel simulation of semiconductor devices. *SIAM Journal of Scientific Computation*, 16(5):1137–1158, 1995.

[18] A. Lumsdaine, M. Reichelt, J. Squyres, J. White. Accelerated waveform methods for parallel transient simulation of semiconductor devices. *IEEE Transactions on Computer-Aided Design*, 15(7):716–726, 1996.

[19] M. Gander. A waveform relaxation algorithm with overlapping splitting for reaction diffusion equation. *Numerical Linear Algebra and Its Applications*, 6:125–145, 1999.

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

[21] C. T. Chen. *Linear System Theory and Design*. Oxford University Press, 1998.

[22] E. Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley and Sons, 1989.

[23] A. E. Ruehli, T. A. Johnson. Circuit analysis computing by waveform relaxation. In *Encyclopedia of Electrical and Electronics Engineering*, volume 3, Wiley, New York, 1999.

[24] T. LeBlanc, T. Cockerill, P. Ledak, H. Hsieh, A. Ruehli. Recent advances in waveform relaxation based circuit simulation. In *Proc. of the IEEE Int. Conf. on Computer Design*, pages 594–596, 1985.

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

[26] T. Johnson and D. Zukowski. Waveform-relaxation-based circuit simulation on the victor v256 parallel processor. *IBM Journal of Research and Development*, 35(5/6):707–720, Sep/Nov 1991.

[27] R. D. Kimmel. IBM simulator toolkit: Version 1.4. Technical report, IBM Corporation, June 2007.

[28] W. Wright. IBM Wv "Analog Waveform" C++ Class Library Home Page. Technical report, IBM Corporation, 2001.

[29] I. M. Elfadel, A. Deutsch, G. Kopcsay, B. Rubin, and H. Smith. A CAD methodology and tool for the characterization of wide on-chip buses. *IEEE Transactions on Advanced Packaging*, 28(1):63–70, February 2005.

[30] A. Lumsdaine, D. Wu. Krylov subspace acceleration of waveform relaxation. *SIAM Journal of Numerical Analysis*, 41(1):90–111, 2003.