Reduction of FDTD Simulation Time with Modal Methods

D. A. Gorodetsky and P. A. Wilsey
University of Cincinnati, USA

Abstract—In order to simulate electromagnetic phenomena at high frequencies, full wave solvers such as the FDTD method must be used. An alternative to the conventional FDTD method is to compute the zero state response with convolution. Convolution results in an increased computation time with every time step. By performing eigenmodal decomposition of the inputs, a constant time for the convolution can be achieved. We show how the solution can be constructed analytically in terms of the eigenvalues and the eigenvectors of the state transition matrix.

1. Introduction

FDTD is an evolutionary scheme that solves Maxwell's equations in the time domain [1, 2]. The evolution continues until steady state or stability in the output is achieved. Schemes of this type are often used when the analytical solution to an electromagnetic problem is prohibitive. Problems to be solved with FDTD are abundant in simulations of aircraft radar cross section at high frequency, microwave ICs, optical pulse propagation, antennas, bioelectromagnetic systems, bodies of revolution, etc. [1]. Situations where it is important to model on-chip interconnect include various microwave circuits such as amplifiers and optoelectronic circuits fabricated in CMOS technology. Reference [3] discusses the design of on-chip waveguides at optical frequencies and reference [4] discusses microwave frequencies. Such real-life problems often require grids with very large numbers of points, due to fine features of the simulated objects and high excitation frequencies. The end result of the fine grids is unreasonable simulation time. With the method proposed in this paper it may be possible to reduce this simulation time to a more acceptable level.

The starting point for the FDTD solution can be the initial conditions, such as an excitation signal. If the solution grid is partitioned into sub grids (i.e., for distributed computation) each containing N field variables, then the starting point is either the initial conditions or the inputs from the adjacent sub grids. We use the $N \times 1$ vector $\mathbf{Q}(n)$ to denote the state of every electric and magnetic field variable in the sub grid. The $N \times N$ state transition matrix $\mathbf{A}(n-i)$ is used to obtain the state at time n from the state at time i. We also define the $N \times 1$ input vector $\mathbf{X}(n)$, to represent the inputs to the sub grid at time n. The manipulation of these matrices in order to get the output of the sub grid, also called a module, was discussed in [5] and only the basic results are given here.

If the inputs are combined in $\mathbf{X}(n)$, an $I \times 1$ vector, then $\mathbf{Y}(n)$, the $O \times 1$ output vector of the module is given as:

$$\mathbf{Y}(n) = [\mathbf{C}\mathbf{A}(n)\mathbf{B}] * \mathbf{X}(n) \tag{1}$$

where the $*$ symbol represents convolution, and the term in brackets is the impulse response $\mathbf{h}(n)$ of the FDTD module. From the results in [5] and from Equation (1) we can observe that the computing time grows with every time step, due to the properties of convolution. Therefore this method is useful only at early stages in the simulation when the number of inputs is small, and the convolution workload does not exceed the time to simulate the module with the standard FDTD.

A strategy to overcome this limitation for the TLM method was discussed in [6]. It involves writing each entry in the location (i, o) of the impulse response matrix as a sum of the eigenvalues of the state transition matrix as follows:

$$h(i,o,n) = \sum_{p=1}^{P} b_{iop}\lambda_p^n = \sum_{p=1}^{P} b_{iop}\left|\lambda_p\right|^n e^{j\omega_p n\Delta t} \tag{2}$$

This can be interpreted as the sum of P matrices, each modulated by a different eigenvalue λ_p. Instead of requiring the storage of the entire history of the inputs, this method requires storage of P matrices, where P is some fraction of N, as will be described later in this paper. This method takes a constant amount of time for every time step of the algorithm, with the number of multiplications given by IOP. In this paper we propose an alternative method that involves decomposing the input vector into a sum of eigenvectors. With the proposed method, the number of multiplications is reduced to OP.

2. Proposed Method

A technique to express the state $\mathbf{Q}(n)$ as a superposition of eigenvectors and to solve for the zero-input response was discussed in [7]. We propose the extension of that work to the zero-state response. In [7] the initial state is written as:

$$\mathbf{Q}(0) = a_0\mathbf{y}_0 + a_1\mathbf{y}_1 + \ldots + a_N\mathbf{y}_N \tag{3}$$

where $\mathbf{y}_k$ are the eigenvectors of the state transition matrix. Using Equation (3), the evolution in time can be expressed as:

$$\mathbf{Q}(n) = (\lambda_0)^n a_0\mathbf{y}_0 + (\lambda_1)^n a_1\mathbf{y}_1 + \ldots + (\lambda_N)^n a_N\mathbf{y}_N \tag{4}$$

Our modification involves expressing the inputs as follows:

$$\begin{aligned} \mathbf{X}(0) &= a_{00}\mathbf{y}_0 + a_{01}\mathbf{y}_1 + \ldots + a_{0N}\mathbf{y}_N \\ \mathbf{X}(1) &= a_{10}\mathbf{y}_0 + a_{11}\mathbf{y}_1 + \ldots + a_{1N}\mathbf{y}_N \\ &\ldots. \\ &\ldots. \\ \mathbf{X}(T) &= a_{T0}\mathbf{y}_0 + a_{T1}\mathbf{y}_1 + \ldots + a_{TN}\mathbf{y}_N \end{aligned} \tag{5}$$

With the inputs expressed as in Equation (5), the convolution will involve keeping only a sum for each column as shown below:

$$\begin{aligned} &\mathbf{Y}(1) = \mathbf{h}(1)\mathbf{X}(0) \\ &= \lambda_0 a_{00}\mathbf{y}_0 + \lambda_1 a_{01}\mathbf{y}_1 + \ldots + \lambda_N a_{0N}\mathbf{y}_N \\ &= \quad s_{10}\mathbf{y}_0 + \quad s_{11}\mathbf{y}_1 + \ldots + s_{1N}\mathbf{y}_N \end{aligned} \qquad \begin{aligned} &\mathbf{Y}(2) = \mathbf{h}(2)\ \mathbf{X}(0) + \mathbf{h}(1)\mathbf{X}(1) \\ &= \lambda_0 s_{10}\mathbf{y}_0 + \lambda_1 s_{11}\mathbf{y}_1 + \ldots + \lambda_N s_{1N}\mathbf{y}_N \\ &+ \lambda_0 a_{10}\mathbf{y}_0 + \lambda_1 a_{11}\mathbf{y}_1 + \ldots + \lambda_N a_{1N}\mathbf{y}_N \\ &= s_{20}\mathbf{y}_0 \quad + s_{21}\mathbf{y}_1 \quad + \ldots + s_{2N}\mathbf{y}_N \end{aligned} \tag{6}$$

From Equation (6) it is clear that a running sum of each column is kept and that convolution involves the multiplication of each column by its eigenvalue. In general the number of multiplications will depend on the number of entries in the $\boldsymbol{y}_k$ vectors and N, the total number of points in the module. Assuming that P out of N eigenvectors are kept for the solution and the remaining ones are discarded, that the size of $\mathbf{Y}$ is $O \times 1$, thus the number of multiplications per time step is reduced to OP.

The complex eigenvalues have a non-zero characteristic frequency obtained by finding the phase angle of the eigenvalue and indicated by $\Omega_i = 2\pi f_i$. The corresponding frequency domain frequency is given by $\omega_i = \Omega_I/\Delta t$ [7]. By properly selecting p, eigenmodes that satisfy the criteria $\Omega_i > 2\pi p$ can be eliminated since it is known that the discretization mechanism of the numerical simulation does not properly propagate these higher frequencies [9]. Adhering to the constraint that only wavelengths that are greater than 10 times the length of a side of a cell can be propagated allows p to be set at 1/10. After the elimination, P indicates the number of remaining eigenmodes.

Hence, the storage of the complex eigenvectors will take up the equivalent of $2OPk_1$ bytes, where k_1 is the number of bytes per double. As can be seen from Equation (6), during every time step two multiplications must be performed for every complex double that is stored. Also, the solution of Equation (5) requires $4OPk_1$ multiplications because the coefficients will in general be complex. Therefore, neglecting additions, every cycle will take roughly $8OPk_2$ milliseconds, where k_2 is the time per multiplication.

3. Results

A module with one interface was analyzed. The dimensions of this module were $1 \times 20 \times 2$ cells. Because the field was assumed zero on the boundary, the module contained only 175 points that participated in the calculation. This resulted in 175 eigenvalues, 116 of which consisted of 58 complex conjugate pairs while the remaining ones were either zero or unity and could be discarded. By setting $p = 0.1$, all but one of the complex conjugate pairs were discarded.

The module was attached to the terminating face a parallel plate waveguide structure that was simulated with the conventional FDTD and with the algorithm presented in this paper. At the excitation face a constant plane wave source of 10 GHz. was introduced. The dimensions of the waveguide without the module were $58 \times 20 \times 2$ cells, which translates to the dimensions of $2 \times 0.0229 \times 0.002$ wavelengths at 10 GHz. The electric field at various points along the length of the waveguide was obtained for the first 10,000 iterations. The results were always virtually identical between the conventional FDTD and our methods. In Figure 1 the electric field

variation with time in cell (29, 10, 1) is shown. Figure 2 demonstrates that the simulation results match the predictions from electromagnetic theory.

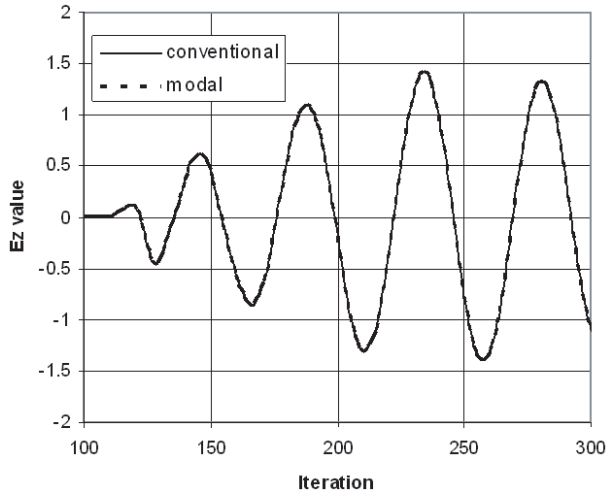

Figure 1: Comparison of results of conventional and proposed methods. The point is located in the middle of the waveguide.

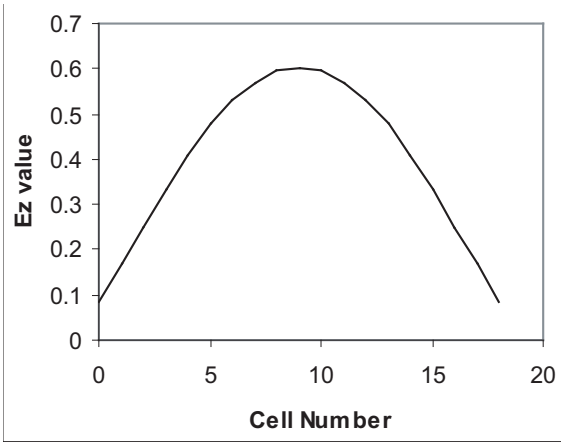

Figure 2: The electric field in the transverse direction. The results correctly indicate the presence of the TM_0 mode.

In order to demonstrate the case when the results were not identical, a module with dimensions of $3 \times 20 \times 2$ cells was utilized. This translated to 525 points. In order to get accurate results from the larger module, p had to be increased to 0.14 in this case. This caused the final system to end up with 16 complex conjugate pairs. The electric field variation in the cell adjacent to the excitation face of the waveguide is displayed in Figure 3. The comparison with the situation where the module produces zero output proves the functionality of the module. Figure 4 shows the small difference between the output of the module at its interface and the electric field produced by the conventional FDTD method at the same point. This difference is barely noticeable in the beginning of the simulation and increases as the simulation progresses in time.

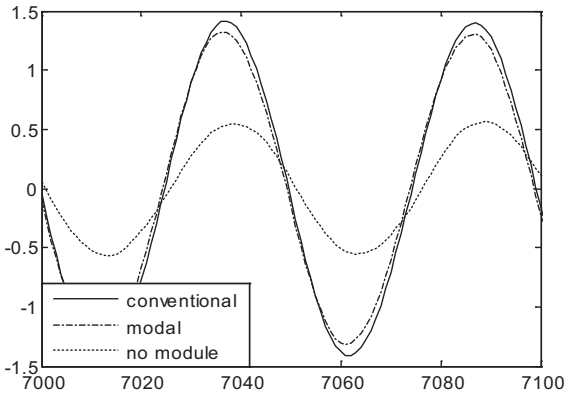

Figure 3: The electric field (Ez) near the waveguide entrance. The overall effect of the module on the simulation can be observed.

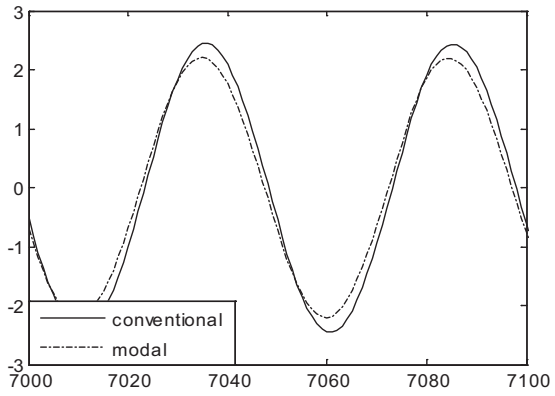

Figure 4: The electric field at the module interface.

4. Conclusion

In this paper we discussed the full-wave simulation of interconnect that is found on high frequency integrated circuits. To speed up the simulation, we developed a recursive algorithm for convolution. This recursive algorithm is based on the modal decomposition approach to the impulse response of the finite-difference time-

domain numerical simulation. Its advantages over an earlier approach [5] is that the storage of the history of the impulse responses (IOT) is no longer required. The only storage required is that of the eigenvectors $(2OP)$, eigenvalues $(2P)$, and coefficients $(2P)$. Another improvement over [5] is that the storage of the inputs (IT) is replaced by the much smaller storage of the coefficients. In regards to the approach published in [6], the storage requirement is improved from $\sim (IOP)$ to $\sim (OP)$ and the number of multiplications per time step is improved in the same manner.

The methods discussed in this paper for interconnect can be extended to a majority of other electromagnetic simulation scenarios such as antennas and radar cross section simulation. An important application is the use of the FDTD method to simulate the propagation of electromagnetic waves in semiconductor devices. This is done by coupling the electron transport equations with Maxwell's equations [10].

Future work will involve the investigation into the techniques, such as change of basis, with which multiple modules can be combined together to reduce the overall simulation time. As was seen in the results, the relationship between the N and P varies with the size of the module as well as the choice of p. More insight into this relationship will be required in order to be able to optimize the module for speed or accuracy requirements.

REFERENCES

1. Taflove, A. and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method, Second Edition*, Artech House, Boston, 2000.
2. De Moura, C. A., "Non-sequential numerical algorithms for time marching models," *RT06/99, IC-UFF*, July 1999.
3. Fullin, E., G. Voirin, M. Chevroulet, A. Lagos, and J. M. Moret, "CMOS-based technology for integrated optoelectronics: a modular approach," *IEEE International Electron Devices Meeting*, 527–530, 1994.
4. Kim, B.-K., B.-K. Ko, K. Lee, J.-W. Jeong, K.-S. Lee, and S.-C. Kim, "Monolithic planar RF inductor and waveguide structures on silicon with performance comparable to those in gaas mmic," *IEEE International Electron Devices Meeting*, 717–720, 1995.
5. Gorodetsky, D. A. and P. A. Wilsey, "A signal processing approach to finite-difference time-domain computations," *IEEE International Midwest Symposium on Circuits and Systems*, 2005.
6. Chen, Z., "Analytic johns matrix and its application in tlm diakoptics," *IEEE MTT-S Digest*, Vol. 2, 777–780, 1995.
7. Chen, Z. and P. P. Silvester, "Analytic solutions for the finite-difference time-domain and transmission-line-matrix methods," *Microwave and Optical Technology Letters*, Vol. 7, No. 1, 5–8, 1994.
8. Grossman, S. I., *Elementary Linear Algebra*, Saunders College Publishing, Fort Worth, 1994.
9. Kunz, K. S. and R. J. Luebbers, *The Finite Difference Time Domain Method for Electromagnetics*, CRC Press, Boca Raton, 1993.
10. El-Ghazaly, S. M. and T. Itoh, "Electromagnetic interfacing of semiconductor devices and circuits," *IEEE MTT-S International Microwave Symposium Digest*, Vol. 1, 151–154, 1997.