GMRES Preconditioners for Multivariate Steady-State Time-Domain Method

Anu Lehtovuori, Jarmo Virtanen, and Martti Valtonen

Department of Electrical and Communications Engineering, Helsinki University of Technology, P.O.Box 3000, FIN-02015 HUT, Finland, E-mail: {anu,jarmo,martti}@aplac.hut.fi

Abstract — **Envelope methods play an important part in current RF simulation. We focus on an envelope method called multivariate steady-state method (MSSTD). The method solves the steady state of a circuit especially in the case when signals at two very different frequencies are present, and harmonic balance is inefficient due to the shape of the signals. Different preconditioners for the iterative matrix solving method (GMRES) have been tested.**

Simulation results show that the preconditioner consisting of larger diagonal blocks decreases the number of iteration cycles needed and also the simulation time. However, other types of preconditioners should be studied to find a more robust and reliable solver for this type of problem and therefore one new approach has been proposed.

I. Introduction

The basic problem in the simulation of RF circuits is the existence of a extreme range of operating frequencies, or time scales, in the same circuit. Widely separated time scales arise in various circuits and systems, ranging from wireless/RF systems, VCOs, PLLs and FM discriminators to switched-capacitor filters, Σ-Δ modulators, mixers, switching power converters, and chirp circuits. The analysis of this kind of circuits is one of the most demanding challenges in circuit simulation because existing methods are effective either in strongly nonlinear problems with one-tone excitation or multi-tone problems with weak nonlinearities.

In conventional time-domain techniques, the time step has to be chosen according to the high-frequency signal, but observing the signal as a whole requires a long simulation time and integration over an excessive number of periods to reach the steady-state.

On the other hand, frequency-domain techniques like the Harmonic Balance method, are inefficient when analyzing circuits which contain sharp waveforms, i.e., when signals are far from sinusoidal. In such cases, the large number of Fourier coefficients needed to accurately describe the unknown waveform causes a rapid increase in the memory consumption and simulation time. Furthermore, the diagonal dominance of the Jacobian matrix is lost when nonlinearities are strong causing convergence problems.

The limited capabilities of established tools have opened a new field in RF and microwave circuit simulation: envelope methods. Roychowdhury used multivariate functions to describe time-dependent phenomena and formulated the problem with multivariate functions producing a multivariate partial differential equation (MPDE) [7]. This paper discusses the mathematics of a finite-difference method for solving MPDE's. The one-dimensional form of the method has been implemented in APLAC [1] circuit simulator earlier and called steady-state time-domain (SSTD) method [4], [6]. Therefore, this new method, which has now been implemented, is called a multivariate SSTD (MSSTD). Results with two different preconditioners are presented here, the focus being on the simulation times and number of iteration cycles. Finally, suggestions for future studies are presented.

II. Multivariate Steady-State Analysis Method

The traditional form of circuit equations is the Differential-Algebraic Equation (DAE)

$$\dot{q}(x) + f(x) = b(t), \tag{1}$$

where q denotes the charge and f the resistive terms. The vector of excitations is $b(t)$, and $x(t)$ is the vector of the unknowns of the circuit. All variables are vector-valued.

A circuit with multirate behavior can be represented efficiently using multiple time variables [7]. We study here only the two-dimensional case and denote the vector forms of $x(t)$ and $b(t)$ by $\hat{x}(t_1, t_2)$ and $\hat{b}(t_1, t_2)$, respectively. Now the original differential equation (1) is replaced with

$$\frac{\partial q(\hat{x})}{\partial t_1} + \frac{\partial q(\hat{x})}{\partial t_2} + f(\hat{x}) = \hat{b}(t_1, t_2). \tag{2}$$

An MPDE is the n-dimensional form of this equation. We solve the steady state, so periodicity is required: $\hat{b}(t_1 + T_1, t_2 + T_2) = \hat{b}(t_1, t_2)$, where T_1 and T_2 are the periods of t_1 and t_2, respectively. Then it is sufficient to solve the values of the unknowns when

$t_1 \in [0, T_1]$ and $t_2 \in [0, T_2]$. Many proofs related to this were presented in [7], as well as three ways to solve the MPDE equation. More than two orders of magnitude speedups compared to traditional transient analysis were achieved. In the multivariate finite difference time domain (MFDTD) method, a two-dimensional grid is created by approximating the differential operators with a numerical differentiation formula. Eq. (2) is discretized on a grid in the (t_1, t_2)-plane. Assume a uniform grid and denote $t_{i,j} = (ih_1, jh_2)$, such that $h_1 = T_1/m_1$ and $h_2 = T_2/m_2$, where m_1 and m_2 are the numbers of samples with respect to t_1 and t_2, respectively. This leads to a set of nonlinear algebraic equations, which contain more unknowns than equations. Therefore, the bi-periodic boundary conditions are used to eliminate additional unknowns. The circuit equations in the two-dimensional form are written separately at every grid point, which creates a system of equations of dimension $m_1 \times m_2$. Then the problem can be solved numerically and the results are the voltages of the circuit at all points of the two-dimensional grid.

The block-structured and diagonally dominant Jacobian matrix improves the convergence [7]. However, the MSSTD problem is not mathematically trivial. Different from the normal finite-difference problem, where a response is needed at every point in the two-dimensional grid, we have to solve the voltage at every point and for each node in the circuit. In addition, the requirement of periodicity breaks the structure of the matrix.

A. Model for the MSSTD component

The operation of APLAC is based on VCCSes [8] such that all device models are constructed from VCCSes. Now we derive a circuit model for a dynamic and static VCCS in the MSSTD analysis.

When two time variables are used, the general form for the current of the dynamic component is

$$\begin{cases} i = \dfrac{\mathrm{dq}}{\mathrm{dt}} = \dfrac{\partial q}{\partial t_1} + \dfrac{\partial q}{\partial t_2} \\ q = q(u). \end{cases} \tag{3}$$

First, time is discretized so that $t_1 = ih_1$ and $t_2 = jh_2$, where h_1 and h_2 are the time steps used with respect to periods T_1 and T_2. Denote the current at a certain time moment with $i_{i,j} = i(ih_1, jh_2)$. Applying the Euler numerical integration formula to Eq. (3) yields

$$i_{i,j} = \frac{q_{i,j} - q_{i-1,j}}{h_1} + \frac{q_{i,j} - q_{i,j-1}}{h_2}. \tag{4}$$

Due to the nonlinear charge dependency, we now apply the iteration formula

$$i^{k+1} \approx i^k + \left.\frac{\partial i}{\partial u}\right|_{u=u^k} (u^{k+1} - u^k), \tag{5}$$

where superscript k is the iteration index. For simplicity, the dependency of one voltage only is assumed. Applying this iteration formula to Eq. (4) leads to the general form of the current of the dynamic nonlinear component:

$$i_{i,j}^{k+1} = g_{i,j}^k u_{i,j}^{k+1} + g_{i-1,j}^k u_{i-1,j}^{k+1} + g_{i,j-1}^k u_{i,j-1}^{k+1} + J^k, \tag{6}$$

where

$$\begin{aligned} g_{i,j}^k &= \left(\frac{1}{h_1} + \frac{1}{h_2}\right) \left.\frac{\partial q}{\partial u}\right|_{u_{i,j}^k}, \\ g_{i-1,j}^k &= -\frac{1}{h_1} \left.\frac{\partial q}{\partial u}\right|_{u_{i-1,j}^k}, \\ g_{i,j-1}^k &= -\frac{1}{h_2} \left.\frac{\partial q}{\partial u}\right|_{u_{i,j-1}^k}, \\ J^k &= \left(\frac{1}{h_1} + \frac{1}{h_2}\right) \left[q_{i,j}^k - \left.\frac{\partial q}{\partial u}\right|_{u_{i,j}^k} u_{i,j}^k \right] \\ &\quad - \frac{1}{h_2} \left[q_{i,j-1}^k - \left.\frac{\partial q}{\partial u}\right|_{u_{i,j-1}^k} u_{i,j-1}^k \right] \\ &\quad - \frac{1}{h_1} \left[q_{i-1,j}^k - \left.\frac{\partial q}{\partial u}\right|_{u_{i-1,j}^k} u_{i-1,j}^k \right]. \end{aligned}$$

The equation can be presented with current sources and VCCSes as shown in Fig. 1. Sources $g_{i-1,j}^k u_{i-1,j}^{k+1}$ and $g_{i,j-1}^k u_{i,j-1}^{k+1}$ are VCCSes having controlling voltages $u_{i-1,j}^{k+1}$ and $u_{i,j-1}^{k+1}$. The value of the current source J^k depends on the voltages in the previous iteration; $J^k = 0$ if the circuit is linear.

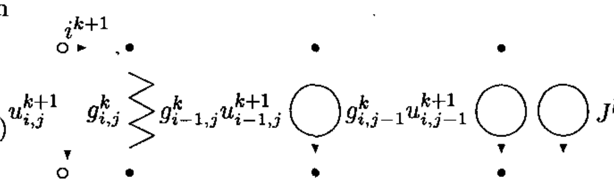

Fig. 1. Circuit model for dynamic component

The operation of a static component does not depend on voltages at other time points and the current is

$$i_{i,j}^{k+1} = \underbrace{\left.\frac{\partial i}{\partial u}\right|_{u=u_{i,j}^k}}_{g_{i,j}^k} u_{i,j}^{k+1} + \underbrace{i_{i,j}^k - \left.\frac{\partial i}{\partial u}\right|_{u=u_{i,j}^k} u_{i,j}^k}_{J^k} \tag{7}$$

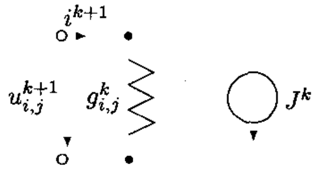

Fig. 2. Circuit model for static component

corresponding to the circuit in Fig. 2.

Eq. (6) is applied to all dynamic components and static components are modeled as in Eq. (7). After that, the equations are collected in a matrix equation $yu = j$, where the unknowns, $u(t)$, are the node voltages of the circuit at all time points in the (T_1, T_2)-grid. The matrix equation can be solved directly by utilizing the matrix solving algorithms used in DC analysis.

III. Solving MSSTD Equations

In APLAC, both SPARSE and GMRES matrix-solving algorithms are implemented in MSSTD analysis. For the sparse-matrix solver, the original circuit has to be duplicated at all desired time points. The problem size grows by the factor $m_1 \times m_2$. The size of the matrix expands rapidly and sparse matrix solving becomes inefficient.

The iterative matrix-solving algorithms like GMRES need a good preconditioner to make them computationally more efficient. One commonly used preconditioner is Jacobi preconditioning, where the preconditioner matrix is equal to the inverse of the diagonal part. Therefore, we have tried to create a diagonal dominant matrix and use the diagonal blocks as preconditioner. This type of preconditioner is generally used in harmonic-balance methods because of quite a simple implementation.

Due to the importance of a good preconditioner, the structure of the matrix is crucial. It depends directly on the order of the unknown variables. The equations have to be ordered such that, first, the circuit at one time point is represented, then at another time point, etc. Then the dimension of the diagonal block is the number of nodes, and the elements outside the diagonal blocks are due to the time-dependency between the different time points.

Two different preconditioners have been implemented and tested. In the first case, only the separate diagonal blocks are included in the preconditioner, i.e., it contains the circuit at different time points without any time dependency as shown in Fig. 3. d denotes the circuit matrix at one time point and matrices c_1 and c_2 of the same size include the dynamic terms corresponding to h_1 and h_2, respectively. In this example, three samples are used with respect to both frequencies.

$$\begin{matrix} d & c_2 & & & & & c_1 & & \\ c_2 & d & & & & & & c_1 & \\ & c_2 & d & & & & & & c_1 \\ c_1 & & & d & c_2 & & & & \\ & c_1 & & c_2 & d & & & & \\ & & c_1 & & c_2 & d & & & \\ & & & c_1 & & & d & c_2 & \\ & & & & c_1 & & c_2 & d & \\ & & & & & c_1 & & c_2 & d \end{matrix}$$

Fig. 3. Preconditioner with small diagonal blocks, PRE1

Another alternative is to use the solution of the one dimensional problem as a preconditioner for the two-dimensional problem (Fig. 4). Then only the time dependency with respect to one time variable present. In this case, the preconditioner blocks are naturally much larger. This corresponds to solving the problem as many one-dimensional problems.

$$\begin{matrix} d & c_2 & & & & & c_1 & & \\ c_2 & d & & & & & & c_1 & \\ & c_2 & d & & & & & & c_1 \\ c_1 & & & d & c_2 & & & & \\ & c_1 & & c_2 & d & & & & \\ & & c_1 & & c_2 & d & & & \\ & & & c_1 & & & d & c_2 & \\ & & & & c_1 & & c_2 & d & \\ & & & & & c_1 & & c_2 & d \end{matrix}$$

Fig. 4. Preconditioner with larger diagonal blocks, PRE2

IV. Simulation results

To demonstrate the effectiveness of the MSSTD method in cases where the shape of the signal is rectangular, a comparison with the HB analysis results is shown in Fig. 5. In this example, the MSSTD analysis took about half of the simulation time needed for HB analysis and the result is more accurate.

Other properties of the MSSTD method are demonstrated next: CPU time consumption and the number of iteration cycles are focused on.

A. Comparison between different preconditioners

Two different preconditioners were implemented so as to be available with the GMRES matrix solving algorithm. CPU times of these two alternatives are compared in Table I, where PRE1 and PRE2 refer to the preconditioners specified in Figs. 3 and 4. The dimension of the problem is in column `dim`. Column `iter` in

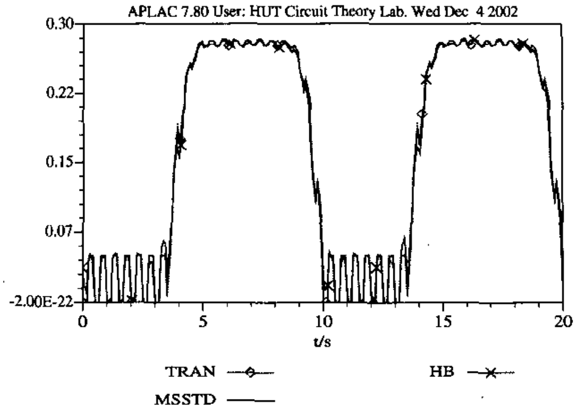

Fig. 5. Voltage waveform compared to HB analysis

Table I shows the numbers of GMRES iteration cycles and the total number of inner iterations.

TABLE I
CPU TIMES WITH DIFFERENT PRECONDITIONERS

file	dim	PRE1 CPU	iter	PRE2 CPU	iter
ex1	2500	1.03 s	11/126	719 ms	11/29
ex2	4050	3.52 s	30/444	1.92 s	21/242
ex3	12400	44.07 s	38/514	40.78 s	32/47

The simulation times with preconditioner PRE2 seem to be smaller. Also the numbers of inner iterations in GMRES are quite different with different preconditioners. It is clear that larger preconditioner blocks improve the convergence in inner iterations. However, the number of GMRES iterations has been decreased, too.

B. Increasing the number of samples

[5] states that for applications to partial differential equations, Jacobian preconditioners may be useful, but should not be expected to have dramatic effects. For this type of problems, the multigrid approach [2], [3] can give more promising results. The multigrid idea corresponds, in practice, to solving a problem with a smaller number of time points and using interpolation to obtain an initial solution at all time points of the fine grid. As an introductory study, we solved the problem first with a smaller number of samples and used the result as an initializer for the final simulation. The simulation times obtained for ex3 are presented in Table II, where the column guess tells the number of samples used in previous simulation and as the basis of the new one.

TABLE II
INCREASING THE NUMBER OF SAMPLES GRADUALLY

points	guess	CPU	total CPU
10×10	-	40.78 s	40.78 s
15×15	-	3 min 41 s	3 min 41 s
15×15	10×10	7.65 s	48.43 s
20×20	10×10	15.74 s	56.52 s
40×40	20×20	46.95 s	1 min 43 s

The simulation time with "15,15" points, doing with "10,10" guess, decreases by over 90 %. Another benefit of increasing the number of samples gradually is the possibility to simulate cases which do not converge otherwise or converge very slowly. Preconditioner PRE1 has been used in these simulations, because it gives more freedom to change the number of samples without changing the size of preconditioner blocks.

V. Conclusions

A multivariate steady-state time-domain analysis has been implemented in APLAC [1]. Both the SPARSE and GMRES matrix solving algorithms are used, GMRES with two alternative preconditioners.

The MSSTD method solves the steady-state of a circuit especially in the case when signals at two very different frequencies are present, and HB is inefficient due to the shape of the signals.

The method has still to be developed to ensure convergence in all situations. The multigrid approach appears promising and is worthy of close study.

References

[1] www.aplac.com

[2] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, 1997.

[3] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer Series in Computational Mathematics 4, Springer-Verlag, 1985.

[4] H. Jokinen, *Computation of the Steady-State Solution of Nonlinear Circuits with Time-Domain and Large-Signal-Small-Signal Analysis Methods*, Doctor's thesis, Helsinki University of Technology, 1997.

[5] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, 1995.

[6] K. Kundert and A. Sangiovanni-Vincentelli, Finding the Steady-state response of Analog and Microwave Circuits, *IEEE Custom Integrated Circuits Conference*, 1988.

[7] J. Roychowdhury, Analyzing Circuits with Widely Separated Time Scales Using Numerical PDE Methods, *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, Vol. 48, No. 5, pp. 578-594, 2001.

[8] M. Valtonen, P. Heikkilä, H. Jokinen, and T. Veijola, APLAC - object-oriented circuit simulator and design tool, *Low-Power HF Microelectronics - a unified approach* (G. A. S. Machado ed.), pp. 333-372, IEE Circuits and Systems Series 8, 1996.