
Numerical Simulation of Electronic Systems Based on Circuit-Block Partitioning Strategies

Jorge dos Santos Freitas de Oliveira

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.75490>

Abstract

Numerical simulation of complex and heterogeneous electronic systems can be a very challenging issue. Circuits composed of a combination of analog, mixed-signal and digital blocks or even radio frequency (RF) blocks, integrated in the same substrate, are very difficult to simulate as a whole at the circuit level. The main reason is because they contain a lot of state variables presenting very distinct properties and evolving in very widely separated time scales. Examples of practical interest are systems-on-a-chip (SoCs), very common in mobile electronics applications, as well as in many other embedded electronic systems. This chapter is intended to briefly review some advanced circuit-level numerical simulation techniques based on circuit-block partitioning schemes, which were especially designed to address the simulation challenges brought by this kind of circuits into the computer-aided-design (CAD) field.

Keywords: numerical simulation, electronic systems, multirate schemes, circuit-block partition

1. Introduction

Electronic circuit simulation has emerged in the 1970s, triggered by the necessity of engineers having a tool to help in design and analysis of integrated circuits (ICs). Since probing internal nodes of semiconductor chips is extremely difficult, or even prohibitive in almost all cases, manufacturing integrated circuits having no help of a simulation tool would lead to an unbearable set of successive physical prototypes until a final solution was achieved. A simulation package combining device modeling and numerical simulation will help engineers to verify correctness and debug circuits during their design, avoiding physical prototyping and reducing product development expenses. Over the years, the continuous scaling of semiconductor devices

and the increasing complexity of electronic architectures have been making CAD tools more and more important for circuits and systems designing. Demands for continuously providing new systems' functionalities, lower-power consumptions or higher transmission rates (e.g., RF and microwave communication systems) are typical requisites that have led to a complex scenario of highly heterogeneous electronic systems. Conventional algorithms are not capable of simulating such kind of electronic systems in an efficient way. The justification for such ineffectiveness relies on the fact that standard simulation techniques do not perform any distinction between nodes or blocks within the circuits, treating all the variables in the same manner. This causes all the blocks of the circuit (analog, mixed-signal, digital or RF blocks) to be computed with the same numerical scheme, without taking their nature into consideration. To cope with this scenario, some advanced numerical simulation algorithms based on circuit-block partition have been proposed in recent years in the scientific literature. The most important ones are briefly reviewed in this chapter.

2. Review of basic circuit simulation concepts

2.1. Mathematical modeling of electronic systems

Dynamic behavior of electronic systems is modeled as systems of differential algebraic equations (DAEs) involving voltages, currents, charges and fluxes. These systems are usually obtained via nodal analysis, or modified nodal analysis (MNA), which consists of applying the Kirchhoff's current law (KCL) to each electrical node and writing the branch currents in terms of the circuit node voltages using the corresponding constitutive relations to each circuit element. Such systems have, in general, the following form:

$$\mathbf{p}(\mathbf{y}(t)) + \frac{d\mathbf{q}(\mathbf{y}(t))}{dt} = \mathbf{x}(t), \quad (1)$$

in which $\mathbf{x}(t) \in \mathbb{R}^n$ is the vector of independent stimuli (voltage or current sources) to the circuit, $\mathbf{y}(t) \in \mathbb{R}^n$ is the vector of unknowns (voltages and currents waveforms) and n is the total number of unknowns. $\mathbf{p} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ describes the memoryless elements in the circuit (linear and nonlinear elements, as resistors, nonlinear voltage-controlled current sources, etc.), while $\mathbf{q} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ models all linear and nonlinear reactive circuit elements, as capacitors or inductors, represented as voltage-dependent electric charges or current-dependent magnetic fluxes, respectively.

Applying the chain differentiation rule to the reactive term of the DAE system of (1), we are able to obtain

$$\frac{d\mathbf{q}(\mathbf{y})}{d\mathbf{y}} \frac{d\mathbf{y}(t)}{dt} = \mathbf{x}(t) - \mathbf{p}(\mathbf{y}(t)), \quad (2)$$

or,

$$M[\mathbf{y}(t)] \frac{d\mathbf{y}(t)}{dt} = \mathbf{x}(t) - \mathbf{p}(\mathbf{y}(t)), \quad (3)$$

where $M[\mathbf{y}(t)]$ is habitually denoted as the *mass matrix*. If $M[\mathbf{y}(t)]$ is invertible then it is possible to convert (3) into the following *ordinary differential equations'* (ODE) system,

$$\frac{d\mathbf{y}(t)}{dt} = M[\mathbf{y}(t)]^{-1}(\mathbf{x}(t) - \mathbf{p}[\mathbf{y}(t)]), \quad (4)$$

which can be rewritten in the classical form

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{f}(t, \mathbf{y}(t)), \quad (5)$$

ordinarily utilized in the mathematical literature. When $M[\mathbf{y}(t)]$ is singular, the DAE system of (3) will not degenerate into a ODE system, but it is often possible to express it as a set of algebraic equations combined with a set of differential equations of the form of (5).

2.2. Classic SPICE-like simulation

The most natural way of simulating the dynamic behavior of an electronic circuit is to numerically time-step integrate, in time domain, the DAE system, or the ODE system, modeling its operation. This means that the solution of (1), or (5), has to be computed over a specified time interval $[t_0, t_{Final}]$ from a specific initial condition $\mathbf{y}(t_0) = \mathbf{y}_0$, leading to the so-called *initial value problems*

$$\mathbf{p}(\mathbf{y}(t)) + \frac{d\mathbf{q}(\mathbf{y}(t))}{dt} = \mathbf{x}(t), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \quad t_0 \leq t \leq t_{Final}, \quad \mathbf{y}(t) \in \mathbb{R}^n, \quad (6)$$

or

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{f}(t, \mathbf{y}(t)), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \quad t_0 \leq t \leq t_{Final}, \quad \mathbf{y}(t) \in \mathbb{R}^n. \quad (7)$$

Computing the solution of (6), or (7), is frequently referred to as *transient analysis* and can be done by using initial value solvers, such as linear multistep methods (LMM) [1] or one-step methods (the popular Runge-Kutta (RK) schemes) [2, 3]. Either LMM or RK methods offer a large variety of explicit and implicit schemes, with very distinct properties in terms of order (accuracy) and numerical stability.

This time-step integration technique was used in the first digital computer programs of circuit analysis initially developed at the Electronics Research Laboratory of the University of California, Berkeley, in the early 1970s, and is still today the most widely used technique for such purpose. It is the core of all SPICE (which means Simulation Program with Integrated Circuit Emphasis) or SPICE-like computer programs, available in many commercial simulators.

2.3. Periodic steady-state simulation

As described earlier, SPICE-like simulation tools are primarily focused on transient analysis. However, in some cases electronics designers are essentially interested in obtaining circuits' steady-state responses and not their transient regimes. The reason for that is because some specific properties of the circuits are better characterized, or simply only defined, in steady-state (e.g., impedance, voltage gain, current gain, harmonic distortion, signal to noise ratio, etc.).

SPICE tools are not adequate for computing steady-state responses of circuits presenting very different time constants, or high Q resonances, as is typically the case of RF and microwave circuits. This is so because they have to pass through the lengthy process of integrating all transients, and expecting them to vanish. Indeed, in such cases time-step integration can be extremely inefficient, since the number of discretization time steps used by the numerical integration scheme will be dramatically enormous. This is because the time interval over which the differential equations must be numerically integrated is set by the lowest frequency, or by how long the circuit takes to achieve steady-state, while the length of the time steps is constrained by the highest frequency component.

Periodic steady-state response of an electronic circuit is a regime where its unknowns are a set of generic waveforms (node voltages and branch currents) presenting a common period. Computing the periodic steady-state response, without having to first integrate all the transients, consists of finding the initial condition, $\mathbf{y}(t_0)$, for the DAE, or ODE, system describing the circuit's operation, such that the values of the unknowns at the end of one period T match their values at the beginning of that period, that is to say, $\mathbf{y}(t_0) = \mathbf{y}(t_0 + T)$. These problems (evaluating the solution to a differential system that satisfies constraints at two or more distinct time instants) are referred to as *boundary value problems*. In this case, we have a *periodic boundary value problem* that can be formulated as:

$$\mathbf{p}(\mathbf{y}(t)) + \frac{d\mathbf{q}(\mathbf{y}(t))}{dt} = \mathbf{x}(t), \quad \mathbf{y}(t_0) = \mathbf{y}(t_0 + T), \quad t_0 \leq t \leq t_0 + T, \quad \mathbf{y}(t) \in \mathbb{R}^n, \quad (8)$$

or, in the ODE form, as

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{f}(t, \mathbf{y}(t)), \quad \mathbf{y}(t_0) = \mathbf{y}(t_0 + T), \quad t_0 \leq t \leq t_0 + T, \quad \mathbf{y}(t) \in \mathbb{R}^n, \quad (9)$$

in which $\mathbf{y}(t_0) = \mathbf{y}(t_0 + T)$ is known as the *periodic boundary condition*.

The most widely used techniques for computing the periodic steady-state solution of electronic circuits are briefly reviewed in the following: the shooting-Newton method [4] and the harmonic balance method [5, 6].

2.3.1. The shooting-Newton method

The shooting-Newton method [4] is the time-domain technique most commonly used by the electronic design automation (EDA) community for computing periodic steady-state solutions of electronic circuits. Shooting-Newton is an iterative technique that: (1) starts by computing the solution of the circuit for one period T using a LMM or RK integrator, considering some

guessed initial condition (which is in general determined from a previous DC analysis); (2) then the computed solution at the end of the period is checked, and if it does not agree with the initial condition, the initial condition is wisely updated and (3) the circuit is then re-simulated for one period with the adjusted initial condition, and this process is repeated until the solution at the end of the period matches the initial condition.

Shooting is an iterative solver that uses an initial value technique to solve a boundary value problem. With the purpose of providing some technical details on the implementation of the shooting-Newton method, let us consider (8). Since with shooting, we perform numerical time-step integration of the DAE system from $t = t_0$ until $t = t_0 + T$, the main difficulty is that we do not know a priori for which initial condition $\mathbf{y}(t_0)$ has to be considered that will lead to the steady-state solution, that is, that will satisfy the periodic boundary condition $\mathbf{y}(t_0) = \mathbf{y}(t_0 + T)$. Thus, the key aspect of shooting-Newton relies on finding the solution of

$$\mathbf{y}(t_0) = \mathbf{y}(t_0 + T) \Leftrightarrow \mathbf{y}(t_0) - \mathbf{y}(t_0 + T) = 0. \quad (10)$$

Let us now define $\phi(\mathbf{y}(t_0), T) = \mathbf{y}(t_0 + T)$, where ϕ is known as the *state-transition function* [4, 7], and rewrite (10) as

$$\phi(\mathbf{y}(t_0), T) - \mathbf{y}(t_0) = 0. \quad (11)$$

Although electronic circuits may operate in strongly nonlinear regimes, their state-transition functions are often moderately nonlinear (or even quite linear). This means that slight perturbations on the initial condition (starting state) produce almost proportional perturbations in the subsequent time states. Taking this aspect into account, it is straightforward to conclude that (11) can be iteratively solved in an efficient way with the Newton's method, which will lead to the following iterative scheme:

$$\phi(\mathbf{y}^{[r]}(t_0), T) - \mathbf{y}^{[r]}(t_0) + \left[\frac{\partial \phi(\mathbf{y}^{[r]}(t_0), T)}{\partial \mathbf{y}(t_0)} - I \right] \Bigg|_{\mathbf{y}(t_0) = \mathbf{y}^{[r]}(t_0)} [\mathbf{y}^{[r+1]}(t_0) - \mathbf{y}^{[r]}(t_0)] = 0, \quad (12)$$

where I is the $n \times n$ identity matrix. The cost of the solution of the linear system of (12) is dominated by the computational effort required to evaluate the derivative of the state-transition function (usually referred to as the *sensitivity matrix*). This matrix is computed taking into account the chain differentiation rule, that is, taking into consideration that $\phi(\mathbf{y}(t_0), T)$ is, in fact, the numerical vector \mathbf{y}_K , with K being the total number of time steps in the interval $[t_0, t_0 + T]$, which depends on the previous step value \mathbf{y}_{K-1} , which, itself, depends on \mathbf{y}_{K-2} , and so forth. The sensitivity matrix is then given by

$$\frac{\partial \phi(\mathbf{y}(t_0), T)}{\partial \mathbf{y}(t_0)} = \frac{\partial \mathbf{y}_K}{\partial \mathbf{y}_{K-1}} \cdot \frac{\partial \mathbf{y}_{K-1}}{\partial \mathbf{y}_{K-2}} \cdot \dots \cdot \frac{\partial \mathbf{y}_1}{\partial \mathbf{y}_0}. \quad (13)$$

Although solving (12) and computing the sensitivity matrix (13) involve substantial computational cost, shooting-Newton converges to the steady-state solution much faster than classic time-step integration. This is the reason why it is the time-domain steady-state engine most widely used in the circuit simulation field.

2.3.2. The harmonic balance method

The harmonic balance (HB) method [4–6] is a frequency domain steady-state simulation technique widely used by the EDA community. In contrast to time domain tools, which represent waveforms as a set of time samples, frequency domain techniques represent periodic signals using coefficients in a sum of complex exponentials (or sines and cosines) harmonically related. The main advantage of HB over time-domain techniques (e.g., shooting-Newton) is that it can represent steady-state solutions (voltage and current waveforms) very accurately using a small number of coefficients. This is especially evident for moderately nonlinear circuits excited by smooth waveforms, in which significant reductions in the computational cost are achieved when HB is used as the simulation tool. However, it must be noted that HB is not suitable for dealing with strongly nonlinear regimes producing waveforms with sharp transitions. In such case, the large number of terms required in the Fourier series expansions will make HB very inefficient.

With the purpose of providing a brief and intuitive explanation of the HB method, let us consider (8). For achieving simplicity in our formulation, let us consider a very small circuit driven by a single periodic source $x(t) \in \mathbb{R}$ satisfying $x(t) = x(t + T)$ and whose dynamic behavior is described by a unique unknown, $y(t) \in \mathbb{R}$ (the generalization to $x(t) \in \mathbb{R}^n$, $y(t) \in \mathbb{R}^n$ is straightforward). Given that the steady-state regime of the circuit will be periodic with the same period T , both the stimulus and the steady-state solution can be represented as the Fourier series

$$x(t) = \sum_{k=-\infty}^{+\infty} X_k e^{jk\omega_0 t}, \quad y(t) = \sum_{k=-\infty}^{+\infty} Y_k e^{jk\omega_0 t}, \quad (14)$$

in which $\omega_0 = 2\pi/T$ is the fundamental frequency. If we substitute (14) into (8), and consider an appropriate harmonic truncation at some order $k = K$, we will obtain

$$p\left(\sum_{k=-K}^{+K} Y_k e^{jk\omega_0 t}\right) + \frac{d}{dt} \left[q\left(\sum_{k=-K}^{+K} Y_k e^{jk\omega_0 t}\right) \right] = \sum_{k=-K}^{+K} X_k e^{jk\omega_0 t}. \quad (15)$$

The HB method converts the differential problem of (8) into the frequency domain, obtaining the $(2K + 1)$ algebraic equations system

$$\mathbf{F}(\mathbf{Y}) = \mathbf{P}(\mathbf{Y}) + j\Omega\mathbf{Q}(\mathbf{Y}) - \mathbf{X} = 0 \quad (16)$$

where $\mathbf{Y} = [Y_{-K}, \dots, Y_0, \dots, Y_K]^T$, $\mathbf{X} = [X_{-K}, \dots, X_0, \dots, X_K]^T$, $j\Omega = \text{diag}(-jK\omega_0, \dots, 0, \dots, jK\omega_0)$.

and \mathbf{P} , \mathbf{Q} are vectors containing the Fourier coefficients of $p(y(t))$ and $q(y(t))$, respectively. The algebraic equations system of (16) is usually denoted as the *harmonic balance system*, which can be iteratively solved using the Newton method

$$\mathbf{F}(\mathbf{Y}^{[r]}) + \left. \frac{d\mathbf{F}(\mathbf{Y})}{d\mathbf{Y}} \right|_{\mathbf{Y}=\mathbf{Y}^{[r]}} [\mathbf{Y}^{[r+1]} - \mathbf{Y}^{[r]}] = 0, \quad (17)$$

in which

$$\frac{d\mathbf{F}(\mathbf{Y})}{d\mathbf{Y}} = \mathbf{J}(\mathbf{Y}) = \mathbf{G}(\mathbf{Y}) + j\Omega\mathbf{C}(\mathbf{Y}) \quad (18)$$

is the $(2K + 1) \times (2K + 1)$ composite conversion matrix, known as the *Jacobian matrix* of the error function $\mathbf{F}(\mathbf{Y})$. \mathbf{G} and \mathbf{C} denote the $(2K + 1) \times (2K + 1)$ conversion matrices (Toeplitz) [7] corresponding to $g(y(t)) = dp(y(t))/dy$ and $c(y(t)) = dq(y(t))/dy$.

3. Univariate time-domain partitioned simulation engines

3.1. Time-domain latency

As highlighted in the Introduction of this chapter, dynamic regimes of operation of some electronic systems may involve signals (voltages and currents) presenting widely distinct time evolution rates. Typical examples of that are coupled analog-digital systems or combined technologies of baseband analog, digital and RF blocks, in the same circuit. In these examples, very fast signals and slowly varying signals cohabit in the same framework. This property, the one of having in the same problem signals presenting rapid time rates of change, while others evolve in a very slow way (or remain approximately constant within a certain time window) is usually denoted as *time-domain latency*. It must be pointed out that time-domain latency is a phenomenon that is not restricted to heterogeneous circuits. For instance, regimes of operation of pure digital electronic systems typically present a set of variables remaining practically constant within a specific time interval, while others evidence quick variations (fast transitions) in that interval.

In Section 2, we have seen that SPICE-like simulation engines (which are based on time-step integration schemes) are widely used for computing the numerical solution of electronic systems. However, when dealing with circuits presenting time-domain latency, that is, containing node voltages and brunch currents evolving at very different rates, traditional SPICE simulators become inefficient because they expend unnecessary work on the computation of the slowly varying components. This is so because classic initial value solvers (as LMM or RK schemes) integrate all the DAE, or ODE, unknowns with the same step size.

3.2. Partitioned algorithms for time-step integration

To deal with the earlier described time-domain latency in an effective way, some modern partitioned algorithms for univariate time-step integration have been proposed in the recent years in the scientific literature [8–10]. These powerful techniques, denoted as multirate Runge-Kutta (MRK) schemes, split the ODE system of (5) into coupled fast and slow (latent) sub-systems, obtaining

$$\begin{aligned} \frac{d\mathbf{y}_F(t)}{dt} &= \mathbf{f}_F(t, \mathbf{y}_F, \mathbf{y}_L), & \mathbf{y}_F(t_0) &= \mathbf{y}_{F,0} \\ \frac{d\mathbf{y}_L(t)}{dt} &= \mathbf{f}_L(t, \mathbf{y}_F, \mathbf{y}_L), & \mathbf{y}_L(t_0) &= \mathbf{y}_{L,0} \end{aligned} \quad (19)$$

with

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_F \\ \mathbf{y}_L \end{bmatrix}, \quad \mathbf{y}_F \in \mathbb{R}^{n_F}, \quad \mathbf{y}_L \in \mathbb{R}^{n_L}, \quad n_F + n_L = n, \quad (20)$$

where \mathbf{y}_F is the vector containing the fast-varying signals and \mathbf{y}_L is the vector holding the slowly varying (latent) ones. The former will be integrated with a small step length h (*microstep*), whereas the latter will be integrated with a large step size H (*macrostep*). The number of microsteps within each macrostep is an integer that we will denote by m . Hence, $H = m \cdot h$. The generic formulation of a MRK scheme is given in the following [8, 9].

Consider two Runge-Kutta methods of s and \bar{s} stages represented by their Butcher tableaux [2] $(\mathbf{b}, \mathbf{A}, \mathbf{c})$ and $(\bar{\mathbf{b}}, \bar{\mathbf{A}}, \bar{\mathbf{c}})$,

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \quad \begin{array}{c|cccc} \bar{c}_1 & \bar{a}_{11} & \bar{a}_{12} & \cdots & \bar{a}_{1\bar{s}} \\ \bar{c}_2 & \bar{a}_{21} & \bar{a}_{22} & \cdots & \bar{a}_{2\bar{s}} \\ \vdots & \vdots & \vdots & & \vdots \\ \bar{c}_{\bar{s}} & \bar{a}_{\bar{s}1} & \bar{a}_{\bar{s}2} & \cdots & \bar{a}_{\bar{s}\bar{s}} \\ \hline & \bar{b}_1 & \bar{b}_2 & \cdots & \bar{b}_{\bar{s}} \end{array} \quad (21)$$

The MRK method conceived for efficiently computing the numerical solution of the partitioned differential system of (19), using a microstep h for integrating the fast unknowns and a macrostep H for integrating the latent unknowns, is defined as follows [8, 9].

The fast-varying vector is obtained by

$$\mathbf{y}_F(t_0 + \lambda h + h) \simeq \mathbf{y}_{F,\lambda+1} = \mathbf{y}_{F,\lambda} + h \sum_{i=1}^s b_i \mathbf{k}_{F,i}^\lambda \quad (22)$$

$$\lambda = 0, 1, \dots, m-1,$$

$$\mathbf{k}_{F,i}^\lambda = \mathbf{f}_F \left(t_0 + \lambda h + c_i h, \mathbf{y}_{F,\lambda} + h \sum_{j=1}^s a_{ij} \mathbf{k}_{F,j}^\lambda, \tilde{\mathbf{Y}}_{L,i}^\lambda \right), \quad (23)$$

$$i = 1, 2, \dots, s,$$

with $\tilde{\mathbf{Y}}_{L,i}^\lambda \simeq \mathbf{y}_L(t_0 + \lambda h + c_i h)$.

The slowly varying vector is given by

$$\mathbf{y}_L(t_0 + H) \simeq \mathbf{y}_{L,1} = \mathbf{y}_{L,0} + H \sum_{i=1}^{\bar{s}} \bar{b}_i \mathbf{k}_{L,i}, \quad (24)$$

$$\mathbf{k}_{L,i} = \mathbf{f}_L \left(t_0 + \bar{c}_i H, \tilde{\mathbf{Y}}_{F,i}, \mathbf{y}_{L,0} + H \sum_{j=1}^{\bar{s}} \bar{a}_{ij} \mathbf{k}_{L,j} \right), \quad (25)$$

$$i = 1, 2, \dots, \bar{s},$$

with $\tilde{\mathbf{Y}}_{F,i} \simeq \mathbf{y}_F(t_0 + \bar{c}_i H)$.

From this definition we can attest that numerical coupling between fast and slow differential subsystems is achieved by $\tilde{Y}_{L,i}^\lambda$ and $\tilde{Y}_{F,i}$. Effective stratagems for computing these intermediate stage values are proposed in [8, 9].

3.3. Circuit-block partitioning strategy

We must be aware that circuit-block partition into fast and slow subsystems can possibly change during the time-step integration process. Hence, it will be very helpful if the simulator is capable of automatically detect the slow and fast variables in the circuit. This automatic classification can be achieved using embedded RK methods [2] and error estimates usually evaluated for step size control and stiffness detection [3]. Fast-slow partitioning strategies, step size control tools, number of microsteps within a macrostep, stiffness detection stratagems and many other technical details of MRK code implementation are thoroughly addressed in [8–10] and also in [11].

Finally, it must be pointed out that significant efficiency gains in computation speed for the simulation of several illustrative examples have been reported in the scientific literature, which demonstrate an effective reduction on the MRK computational cost in comparison to traditional SPICE simulation engines.

4. Multitime partitioned simulation engines

This section is devoted to briefly review some advanced circuit-block partitioning numerical simulation techniques operating in a multivariate time-domain framework. Section 4.1 introduces the multivariate formulation theory, in which the 1-D time is converted into a set of artificial time variables. Section 4.2 addresses some fundamental aspects of the numerical simulation algorithms. Finally, Section 4.3 describes some techniques for automatic circuit-block partition.

4.1. Multivariate formulation

The multivariate formulation is a useful stratagem that plays an important role in the EDA scientific community, especially in the RF and microwave areas. It was initially introduced in 1996 [12] as a sophisticated derivation of quasi-periodic HB, and it has been adopted by other researchers (e.g., [13–19]), which have demonstrated that it can be an efficient strategy when dealing with electronic circuits operating on widely distinct time scales. The success of multivariate formulation relies on the fact that voltages and currents containing components that evolve themselves at two, or more, widely separated rates of variation can be represented much more efficiently if we define them as functions of two, or more, time variables (artificial time scales). With this stratagem all signals (stimuli and responses) will be represented as multivariate functions, which will imply that dynamic behavior of the circuits will no longer be modeled by DAE, or ODE, systems formulated in the 1-D time. It will be described by partial differential systems.

In order to provide a simple and illustrative mathematical description of the multivariate formulation let us consider the classical example of a generic nonlinear RF circuit driven by an envelope modulated signal of the form

$$x(t) = e(t) \cos(\omega_C t + \phi(t)), \quad (26)$$

where $e(t)$ and $\phi(t)$ are, respectively, the amplitude and phase slowly varying baseband signals (envelope), modulating the $\cos(\omega_C t)$ fast-varying carrier. Computing the numerical solution of such circuit using conventional 1-D time-step integrators (RK methods or LMM methods) is very expensive. This is so because the response of the circuit has to be evaluated during a long time window defined by the slowly varying envelope, wherein the time-step size is severely restricted by the high frequency carrier. Taking into consideration the disparity between the baseband and the carrier time scales, and assuming that they are also uncorrelated, which is typically true, we are able to rewrite (26) as a bivariate function

$$\hat{x}(t_1, t_2) = e(t_1) \cos(\omega_C t_2 + \phi(t_1)), \quad (27)$$

where t_1 is the slow baseband artificial time scale and t_2 is the fast carrier artificial time scale. It must be noted that $\hat{x}(t_1, t_2)$ is a periodic function with respect to t_2 but not to t_1 , that is,

$$\hat{x}(t_1, t_2) = \hat{x}(t_1, t_2 + T_2), \quad T_2 = 2\pi/\omega_C. \quad (28)$$

This means that a generic 1-D $[0, t_{Final}]$ time interval will be mapped into a 2-D $[0, t_{Final}] \times [0, T_2]$ rectangular domain. It is easy to attest that, in general, the number of points required to represent $\hat{x}(t_1, t_2)$ in $[0, t_{Final}] \times [0, T_2]$ is much less than the number of points needed to represent $x(t)$ in $[0, t_{Final}]$. This is especially evident when the t_1 and t_2 time scales are widely separated [13, 14].

Let us now consider the DAE system of (1) modeling the dynamic behavior of a generic RF circuit excited by the envelope modulated signal of (26). Taking the abovementioned into account, we will adopt the following procedure for the vector-valued functions $\mathbf{x}(t)$ and $\mathbf{y}(t)$: t is replaced by t_1 in the slowly varying entities (envelope time scale) and t is replaced by t_2 in the fast-varying entities (RF carrier time scale). The application of this stratagem will turn the DAE system of (1) into the following partial differential system

$$\mathbf{p}(\hat{\mathbf{y}}(t_1, t_2)) + \frac{\partial \mathbf{q}(\hat{\mathbf{y}}(t_1, t_2))}{\partial t_1} + \frac{\partial \mathbf{q}(\hat{\mathbf{y}}(t_1, t_2))}{\partial t_2} = \hat{\mathbf{x}}(t_1, t_2), \quad (29)$$

usually denoted as *multirate partial differential algebraic equations'* (MPDAE) system [13, 14]. It is easy to demonstrate that, if $\hat{\mathbf{x}}(t_1, t_2)$ and $\hat{\mathbf{y}}(t_1, t_2)$ satisfy (29), then the univariate forms $\mathbf{x}(t) = \hat{\mathbf{x}}(t, t)$ and $\mathbf{y}(t) = \hat{\mathbf{y}}(t, t)$ satisfy (1) [13]. Thus, $\mathbf{y}(t)$ may be retrieved from its bivariate form $\hat{\mathbf{y}}(t_1, t_2)$ by simply setting $t_1 = t_2 = t$, meaning that univariate solutions of (1) are available on diagonal lines $t_1 = t, t_2 = t$, along the bivariate solutions of (29). In truth, attending to the periodicity of the problem in the t_2 time dimension, the univariate version of the vector containing the circuit responses is obtained from its bivariate representation on the rectangular domain $[0, t_{Final}] \times [0, T_2]$ as

$$\mathbf{y}(t) = \hat{\mathbf{y}}(t, t \bmod T_2), \quad (30)$$

where $t \bmod T_2$ represents the remainder of division of t by T_2 .

The generalization of the bivariate strategy to a multidimensional problem with more than two time scales is straightforward. In fact, if the signals in the circuit present m separate rates of change, then m time scales will be used. In that case (29) assumes the generic form

$$p(\hat{\mathbf{y}}(t_1, t_2, \dots, t_m)) + \frac{\partial q(\hat{\mathbf{y}}(t_1, t_2, \dots, t_m))}{\partial t_1} + \dots + \frac{\partial q(\hat{\mathbf{y}}(t_1, t_2, \dots, t_m))}{\partial t_m} = \hat{\mathbf{x}}(t_1, t_2, \dots, t_m), \quad (31)$$

and the univariate solution, $\mathbf{y}(t)$, may be recovered from its multivariate form, $\hat{\mathbf{y}}(t_1, t_2, \dots, t_m)$, by setting $t_1 = t_2 = \dots = t_m = t$.

4.2. Partitioned algorithms for envelope following computation

The main advantage of the earlier described MPDAE approach is that it can result in significant improvements in simulation speed when compared to DAE-based alternatives [13–15]. However, by itself this approach does not perform any distinction between nodes or blocks in the circuit under analysis. In fact, in the first multivariate circuit simulation schemes initially introduced in [12], and then in [13], the same numerical algorithm was used to compute all the unknowns of the circuit. Only a few years later other advanced multivariate algorithms were proposed (e.g., [16–19]) in way to take into account possible circuit’s heterogeneities. These algorithms are based on circuit-block partitioning stratagems defined within the multivariate frameworks. The most important ones regarding pure time-domain operations are briefly reviewed in the following.

As stated earlier, envelope modulated regimes are typical cases of practical interest. Computing responses to excitations of the form of (26) is a technique generally referred to as *envelope following*, which correspond to a combination of initial and periodic boundary conditions in the bivariate framework, leading to the following initial-boundary value problem

$$\begin{aligned} p(\hat{\mathbf{y}}(t_1, t_2)) + \frac{\partial q(\hat{\mathbf{y}}(t_1, t_2))}{\partial t_1} + \frac{\partial q(\hat{\mathbf{y}}(t_1, t_2))}{\partial t_2} &= \hat{\mathbf{x}}(t_1, t_2) \\ \hat{\mathbf{y}}(0, t_2) &= \mathbf{g}(t_2) \\ \hat{\mathbf{y}}(t_1, 0) &= \hat{\mathbf{y}}(t_1, T_2) \end{aligned} \quad (32)$$

defined on the rectangular domain $[0, t_{Final}] \times [0, T_2]$. $\mathbf{g}(\cdot)$ is a vector-valued initial-condition satisfying $\mathbf{g}(0) = \mathbf{g}(T_2) = \mathbf{y}(0)$, and $\hat{\mathbf{y}}(t_1, 0) = \hat{\mathbf{y}}(t_1, T_2)$ is the periodic boundary condition due to the periodicity of the problem in the t_2 fast time scale. In order to solve this initial-boundary value problem let us begin to consider the following semi-discretization of $[0, t_{Final}] \times [0, T_2]$ in the t_1 slow time dimension defined by

$$0 = t_{1,0} < t_{1,1} < \dots < t_{1,i-1} < t_{1,i} < \dots < t_{1,K_1} = t_{Final}, \quad h_{1,i} = t_{1,i} - t_{1,i-1}, \quad (33)$$

in which K_1 is the total number of steps in t_1 . Now, using a backward differentiation formula (BDF) [1] to approximate the t_1 derivatives of the MPDAE (for simplicity let us consider here the Gear-2 rule [1]), we obtain for each slow time instant $t_{1,i}$, from $i = 1$ to $i = K_1$, the periodic boundary value problem defined by

$$p(\hat{\mathbf{y}}_i(t_2)) + \frac{3q(\hat{\mathbf{y}}_i(t_2)) - 4q(\hat{\mathbf{y}}_{i-1}(t_2)) + q(\hat{\mathbf{y}}_{i-2}(t_2))}{2h_{1,i}} + \frac{dq(\hat{\mathbf{y}}_i(t_2))}{dt_2} = \hat{\mathbf{x}}(t_{1,i}, t_2), \quad (34)$$

$$\hat{\mathbf{y}}_i(0) = \hat{\mathbf{y}}_i(T_2),$$

where $\hat{\mathbf{y}}_i(t_2) \simeq \hat{\mathbf{y}}(t_{1,i}, t_2)$. This means that, once $\hat{\mathbf{y}}_{i-1}(t_2)$ is known, the solution on the next slow time instant, $\hat{\mathbf{y}}_i(t_2)$, is obtained by solving (34). Hence, a set of K_1 boundary value problems have to be solved for obtaining the whole bivariate solution in the entire domain $[0, t_{Final}] \times [0, T_2]$. A very powerful technique has been proposed in the literature for solving each of the periodic boundary problems defined by (34) in an efficient way [16, 17]. This technique uses shooting-Newton based on MRK schemes. It splits the circuits into two distinct subsets according to the time evolution rates of their voltages and currents and performs time-step integration with different step lengths in each of the consecutive shooting iterations needed to solve (34). For that (34) is firstly divided into the following coupled fast-slow subsystems

$$p_F(\hat{\mathbf{y}}_{F,i}(t_2), \hat{\mathbf{y}}_{L,i}(t_2)) + \frac{3q_F(\hat{\mathbf{y}}_{F,i}(t_2), \hat{\mathbf{y}}_{L,i}(t_2)) - 4q_F(\hat{\mathbf{y}}_{F,i-1}(t_2), \hat{\mathbf{y}}_{L,i-1}(t_2)) + q_F(\hat{\mathbf{y}}_{F,i-2}(t_2), \hat{\mathbf{y}}_{L,i-2}(t_2))}{2h_{1,i}} + \frac{dq_A(\hat{\mathbf{y}}_{F,i}(t_2), \hat{\mathbf{y}}_{L,i}(t_2))}{dt_2} = \hat{\mathbf{x}}(t_{1,i}, t_2), \quad (35)$$

$$p_L(\hat{\mathbf{y}}_{F,i}(t_2), \hat{\mathbf{y}}_{L,i}(t_2)) + \frac{3q_L(\hat{\mathbf{y}}_{F,i}(t_2), \hat{\mathbf{y}}_{L,i}(t_2)) - 4q_L(\hat{\mathbf{y}}_{F,i-1}(t_2), \hat{\mathbf{y}}_{L,i-1}(t_2)) + q_L(\hat{\mathbf{y}}_{F,i-2}(t_2), \hat{\mathbf{y}}_{L,i-2}(t_2))}{2h_{1,i}} + \frac{dq_L(\hat{\mathbf{y}}_{F,i}(t_2), \hat{\mathbf{y}}_{L,i}(t_2))}{dt_2} = \hat{\mathbf{x}}(t_{1,i}, t_2),$$

with

$$\hat{\mathbf{y}}_i(t_2) = \begin{bmatrix} \hat{\mathbf{y}}_{F,i}(t_2) \\ \hat{\mathbf{y}}_{L,i}(t_2) \end{bmatrix}, \quad \hat{\mathbf{y}}_{F,i}(t_2) \in \mathbb{R}^{n_F}, \quad \hat{\mathbf{y}}_{L,i}(t_2) \in \mathbb{R}^{n_L}, \quad n_F + n_L = n. \quad (36)$$

$\hat{\mathbf{y}}_{F,i}(t_2)$ is the vector containing the fast-varying circuit variables at the slow time instant $t_{1,i}$, and $\hat{\mathbf{y}}_{L,i}(t_2)$ is the vector holding the slowly varying ones at the same slow time instant. The former will be integrated in t_2 with a small step size h_2 , while the later will be integrated with a much larger step size H_2 .

4.3. Circuit-block partitioning strategy

Since the partition of the electronic system in fast and slow subsystems may dynamically vary with time, it will be of great usefulness if the simulation algorithm is capable of automatically

detecting the fast-varying and the slowly varying signals. The approach adopted in [16, 17] for that purpose is briefly described in the following.

As mentioned earlier, each of the periodic boundary value problems defined by (34) is solved using shooting-Newton based on multirate time-step integrators (MRK schemes). Now, taking into account that shooting is an iterative technique, the key idea of this partitioning strategy is to use a uni-rate scheme on the first shooting iterations needed for each time line $t_{1,i} \times [0, T_2]$ of the $[0, t_{Final}] \times [0, T_2]$ rectangular domain. For that, it starts by considering $m = 1$ in the adopted MRK scheme, so that it degenerates into a standard uni-rate RK scheme. This means that all the unknowns (voltages or currents) will be integrated in t_2 with the same microstep h_2 . After that, the differential system of (34) is partitioned into (35) according to the variations in the t_2 derivatives of the unknowns. Each unknown that practically evidences no variations in its t_2 time derivatives for the entire time line $t_{1,i} \times [0, T_2]$, that is, each unknown that satisfies the condition

$$\left| \max(\text{slope}_j) - \min(\text{slope}_j) \right| < Tol, \tag{37}$$

where Tol is a small specified tolerance, will be treated as slow on the next shooting iterations. The remaining unknowns will be treated as fast. In a cheap scheme within a uniform grid each slope can be simply given by

$$\text{slope}_j = [\widehat{y}_i(t_{2,j} + h_2) - \widehat{y}_i(t_{2,j})]/h_2, \quad t_{2,j} \in \{0, h_2, 2h_2, \dots, T_2 - h_2\}. \tag{38}$$

All the subsequent shooting iterations on the time line $t_{1,i} \times [0, T_2]$ will be conducted in a multirate way using different step sizes. In a nonuniform grid, step sizes h_2 and $H_2 = m \cdot h_2$ may be chosen using any step-size control tool. In a uniform grid, they can be predefined or successively refined to achieve a desired accuracy.

The robustness of this partitioning strategy can be improved if more than one shooting iteration with the uni-rate scheme is considered for each time line $t_{1,i} \times [0, T_2]$. However, such tactic will obviously conduct to some extra computational cost, turning the overall algorithm to be a little less efficient.

Finally, to conclude this section, it must be highlighted that although efficiency gains are essentially dependent on the ratio between the number of slow and fast signals, significant reductions on the computational effort were reported in the scientific literature (e.g., [16, 17]) for the simulation of several illustrative examples of practical interest using the envelope following bivariate partitioned techniques.

5. Hybrid time-frequency partitioned techniques

This section is intended to provide a brief description of powerful circuit-block partitioning numerical simulation techniques operating in multivariate hybrid time-frequency frameworks. Section 5.1 introduces the fundamentals of multivariate hybrid envelope following. Section 5.2

addresses some basic details of the simulation algorithms and finally, Section 5.3 describes the approach for automatic circuit-block partition.

5.1. Multivariate hybrid envelope following

Let us once again consider the initial-boundary value problem of (32) characterizing the bivariate nature of an electronic circuit operating at two widely separated time scales. Let us also consider the semi-discretization of the $[0, t_{Final}] \times [0, T_2]$ rectangular domain, which has led to the set of periodic boundary value problems defined by (34). When waveforms are not excessively demanding on the quantity of harmonic components needed for an accurate frequency domain representation, we are able to use the HB method for efficiently computing the solution of (34). In such case we will obtain for each time line $t_{1,i} \times [0, T_2]$ the following HB system

$$\mathbf{P}(\widehat{\mathbf{Y}}(t_{1,i})) + \frac{3\mathbf{Q}(\widehat{\mathbf{Y}}(t_{1,i})) - 4\mathbf{Q}(\widehat{\mathbf{Y}}(t_{1,i-1})) + \mathbf{Q}(\widehat{\mathbf{Y}}(t_{1,i-2}))}{2h_{1,i}} + j\Omega\mathbf{Q}(\widehat{\mathbf{Y}}(t_{1,i})) = \widehat{\mathbf{X}}(t_{1,i}), \quad (39)$$

in which $\widehat{\mathbf{X}}(t_{1,i})$ is a $(2K + 1) \times n$ vector containing the Fourier coefficients of the stimuli (independent sources) and $\widehat{\mathbf{Y}}(t_{1,i})$ a $(2K + 1) \times n$ vector containing the Fourier coefficients of the unknowns (node voltages and brunch currents' waveforms), at $t_1 = t_{1,i}$. K is the maximum harmonic order and n is the number of unknowns. In order to obtain the solution in the entire $[0, t_{Final}] \times [0, T_2]$ rectangular domain a total of K_1 HB systems have to be solved.

With this hybrid time-frequency approach we have an envelope following technique that handles the slow variations of the unknowns in the time domain and their fast variations in the frequency domain. This technique, usually denoted as multivariate *envelope transient harmonic balance* [20, 21], is able to exploit the existence of time-rate disparities in circuits operating under moderately nonlinear regimes of operation.

5.2. Partitioned algorithms for hybrid envelope following computation

The hybrid time-frequency envelope following technique presented earlier does not perform any distinction between nodes or blocks in the circuit. Although it has been widely used, especially by the RF and microwave community, only a few years ago other versions regarding circuit-block partition were proposed [18, 19] in way to take into account possible circuit's heterogeneities. The most important aspects of these partitioning techniques are briefly reviewed in the following.

Let us rewrite (39) as

$$\begin{aligned} \mathbf{F}(\widehat{\mathbf{Y}}(t_{1,i})) = \mathbf{P}(\widehat{\mathbf{Y}}(t_{1,i})) + \frac{3\mathbf{Q}(\widehat{\mathbf{Y}}(t_{1,i})) - 4\mathbf{Q}(\widehat{\mathbf{Y}}(t_{1,i-1})) + \mathbf{Q}(\widehat{\mathbf{Y}}(t_{1,i-2}))}{2h_{1,i}} \\ + j\Omega\mathbf{Q}(\widehat{\mathbf{Y}}(t_{1,i})) - \widehat{\mathbf{X}}(t_{1,i}) = 0. \end{aligned} \quad (40)$$

The Newton iterative solver is usually utilized to solve (40), leading to

$$\mathbf{F}\left(\widehat{\mathbf{Y}}^{[r]}(t_{1,i})\right) + \left. \frac{d\mathbf{F}\left(\widehat{\mathbf{Y}}(t_{1,i})\right)}{d\widehat{\mathbf{Y}}(t_{1,i})} \right|_{\widehat{\mathbf{Y}}(t_{1,i})=\widehat{\mathbf{Y}}^{[r]}(t_{1,i})} \left[\widehat{\mathbf{Y}}^{[r+1]}(t_{1,i}) - \widehat{\mathbf{Y}}^{[r]}(t_{1,i}) \right] = 0, \quad (41)$$

which implies that for computing $\widehat{\mathbf{Y}}^{[r+1]}(t_{1,i})$ from the previous estimate $\widehat{\mathbf{Y}}^{[r]}(t_{1,i})$ we have to solve a linear system composed of $n \times (2K + 1)$ equations. The system of (41) involves the so-called Jacobian matrix of $\mathbf{F}\left(\widehat{\mathbf{Y}}(t_{1,i})\right)$, which has a block structure, consisting of an $n \times n$ matrix of square submatrices (blocks), each one with dimensions $(2K + 1) \times (2K + 1)$. The general block of row m and column l can be expressed as

$$\begin{aligned} \frac{d\mathbf{F}_m\left(\widehat{\mathbf{Y}}(t_{1,i})\right)}{d\widehat{\mathbf{Y}}_l(t_{1,i})} &= \frac{d\mathbf{P}_m\left(\widehat{\mathbf{Y}}(t_{1,i})\right)}{d\widehat{\mathbf{Y}}_l(t_{1,i})} + \frac{1}{h_{1,i}} \frac{d\mathbf{Q}_m\left(\widehat{\mathbf{Y}}(t_{1,i})\right)}{d\widehat{\mathbf{Y}}_l(t_{1,i})} \\ &+ j\Omega \frac{d\mathbf{Q}_m\left(\widehat{\mathbf{Y}}(t_{1,i})\right)}{d\widehat{\mathbf{Y}}_l(t_{1,i})}, \quad l, m = 1, 2, \dots, n, \end{aligned} \quad (42)$$

where $d\mathbf{P}_m\left(\widehat{\mathbf{Y}}(t_{1,i})\right)/d\widehat{\mathbf{Y}}_l(t_{1,i})$ and $d\mathbf{Q}_m\left(\widehat{\mathbf{Y}}(t_{1,i})\right)/d\widehat{\mathbf{Y}}_l(t_{1,i})$ denote the Toeplitz matrices [7] of the vectors containing the Fourier coefficients of $dp_m(\widehat{y}(t_{1,i}, t_2))/d\widehat{y}_l(t_{1,i}, t_2)$ and $dq_m(\widehat{y}(t_{1,i}, t_2))/d\widehat{y}_l(t_{1,i}, t_2)$, respectively.

A very powerful technique has been proposed in the literature [18, 19] for solving (41) in an efficient way. This technique takes into account that, in some cases (e.g., RF heterogeneous systems), there are parts of the circuits in which there are no fluctuations dictated by the fast carrier. As a consequence, bidimensional forms $\widehat{y}(t_1, t_2)$ of voltages and currents in those parts have no dependence on t_2 . This means that for each time line $t_{1,i} \times [0, T_2]$ each signal $\widehat{y}(t_{1,i}, t_2)$ has a constant value that can be represented as a Fourier series with a unique $k = 0$ coefficient. Thus, while quickly varying signals are represented in the frequency domain by a set of $(2K + 1)$ Fourier coefficients, signals in latent (slowly varying) blocks are represented by a single coefficient. Having this in mind it is straightforward to conclude that the size of vector $\widehat{\mathbf{Y}}(t_{1,i})$ will be significantly reduced, as well as the size (number of equations) of the HB system. Significant Jacobian matrix size reductions will also be achieved, since some of the submatrices will no longer be $(2K + 1) \times (2K + 1)$ matrices, but, instead, simple 1×1 scalar elements. For instance, let us suppose that the m th component of (40) is a slowly varying signal exclusively dependent on other slowly varying signals. Let us also suppose that the l th component is also a slowly varying signal. Then, the Jacobian matrix block of row m and column l will be a 1×1 block given by

$$\frac{d\mathbf{F}_m\left(\widehat{\mathbf{Y}}(t_{1,i})\right)}{d\widehat{\mathbf{Y}}_l(t_{1,i})} = \frac{dp_m(\widehat{y}(t_{1,i}))}{d\widehat{y}_l(t_{1,i})} + \frac{1}{h_{1,i}} \frac{dq_m(\widehat{y}(t_{1,i}))}{d\widehat{y}_l(t_{1,i})}. \quad (43)$$

It may be noted that this 1×1 scalar block can be seen as a particular case of the general $(2K + 1) \times (2K + 1)$ block of (42), if $K = 0$ is assumed as the maximum harmonic order. Actually, given that $df_m(\hat{y}(t_{1,i}, t_2))/d\hat{y}_l(t_{1,i}, t_2)$ is a constant function evidencing no fluctuations in the t_2 fast time scale, the last term of (42) will vanish and there will be no more necessity of converting the right-hand side terms of (43) into the frequency domain.

Since only fast-varying signals are converted into the frequency domain, this partitioned technique can be seen as a combination of multivariate envelope transient harmonic balance (by treating the fast-varying signals in a hybrid time-frequency framework) with a pure time marching simulation engine (by treating some of the signals in a pure time domain scheme). Thus, beyond the notorious significant vector and matrix size reductions above mentioned, there is another important advantage brought by this partitioned technique. For example, in complex heterogeneous RF systems strongly nonlinear regimes of operation are in general associated to digital or baseband blocks, whereas moderately nonlinear regimes are typical of RF blocks. With this partitioned technique signals in digital and baseband blocks are appropriately computed in the time domain, while signals in RF blocks are treated in the hybrid time-frequency framework.

5.3. Circuit-block partitioning strategy

Similar to what we have mentioned for the methods discussed in the previous sections, it will be of great utility if the simulator is able to automatically distinguish the fast-varying variables from the slowly varying ones. We now briefly review the approach addressed in [19] for that purpose, which splits the circuit into distinct blocks according to the time rates of change of their voltages and currents.

Let us consider the HB system of (39). As stated earlier, this system has to be solved with the iterative scheme of (41) for each artificial time line $t_{1,i} \times [0, T_2]$. The automatic partitioning strategy proposed in [19] consists in considering all the circuit's variables as fast on the first iteration of (41), that is, it consists in initially representing all the unknowns in the frequency domain as a set of $(2K + 1)$ Fourier coefficients. This way, the algorithm starts by computing a single iteration in (41) to evaluate $\hat{\mathbf{Y}}^{[1]}(t_{1,i})$

$$\hat{\mathbf{Y}}^{[1]}(t_{1,i}) = \left[\hat{\mathbf{Y}}_1^{[1]}(t_{1,i})^T, \hat{\mathbf{Y}}_2^{[1]}(t_{1,i})^T, \dots, \hat{\mathbf{Y}}_n^{[1]}(t_{1,i})^T \right]^T, \quad (44)$$

where each $\hat{\mathbf{Y}}_v(t_{1,i})$, $v = 1, \dots, n$, is a $(2K + 1) \times 1$ vector defined as

$$\hat{\mathbf{Y}}_v^{[1]}(t_{1,i}) = \left[Y_{v,-K}^{[1]}(t_{1,i}), \dots, Y_{v,0}^{[1]}(t_{1,i}), \dots, Y_{v,K}^{[1]}(t_{1,i}) \right]^T. \quad (45)$$

Each of the $\hat{\mathbf{Y}}_v^{[1]}(t_{1,i})$ is then inspected. If its Fourier coefficients of order $k \neq 0$ are practically null (their absolute values stay under a very small prescribed tolerance) it will be classified as slow. Otherwise it will be classified as fast. After this classification, which will temporarily split the system into fast and slow subsystems, the simulator considers that signals in slow subsystems can be represented by a single Fourier coefficient for the remaining iterations needed to compute the solution of (41).

Similar to what we have discussed for the method described in Section 4, the robustness of this partitioning strategy may be improved if more than one iteration in (41) is computed before the simulator decides which signals are slow and which signals are fast. The main drawback of such approach is the loss of some efficiency due to the extra computational effort required.

As a final point of this section, we would like to point out that significant gains in computation speed have been reported in the scientific literature (e.g., [18, 19]) for the simulation of several illustrative examples of practical interest using these hybrid time-frequency envelope following partitioned algorithms.

6. Conclusions

In this chapter, we have briefly reviewed some powerful numerical simulation techniques based on partitioned stratagems. Such techniques were especially designed to cope with the simulation challenges brought by emerging electronic technologies to the EDA community, as is the case of complex heterogeneous electronic systems composed of a combination of different kinds of circuit blocks (analog, mixed-signal and digital blocks, or even radio frequency blocks) containing node voltages and branch currents of very distinct formats and running on widely separated time scales. With these partitioned techniques signals within different blocks of the circuits are computed with distinct algorithms and/or step sizes. Considerable reductions on the computational cost have been registered in several experiments published in the scientific literature (in comparison to previously recognized techniques) without compromising the accuracy of the results.

Acknowledgements

This work is funded by National Funds through FCT—Fundação para a Ciência e Tecnologia, under the project UID/EEA/50008/2013.

Author details

Jorge dos Santos Freitas de Oliveira^{1,2*}

*Address all correspondence to: jorge.oliveira@ipleiria.pt

1 School of Technology and Management, Polytechnic Institute of Leiria, Leiria, Portugal

2 Institute of Telecommunications, University of Aveiro, Aveiro, Portugal

References

- [1] Lambert J. Numerical Methods for Ordinary Differential Systems: The Initial Value Problem. 1st ed. West Sussex: Wiley; 1991

- [2] Hairer E, Nørsett S, Wanner G. Solving Ordinary Differential Equations I: Nonstiff Problems. 1st ed. Berlin: Springer-Verlag; 1987. DOI: 10.1007/978-3-662-12607-3
- [3] Hairer E, Wanner G. Solving Ordinary Differential Equations II: Stiff and Differential Algebraic Problems. 2nd ed. Berlin: Springer-Verlag; 1996. DOI: 10.1007/978-3-642-05221-7
- [4] Kundert K, White J, Sangiovanni-Vincentelli A. Steady-State Methods for Simulating Analog and Microwave Circuits. Norwell, MA: Springer; 1990
- [5] Rodrigues P. Computer-Aided Analysis of Nonlinear Microwave Circuits. Norwood, MA: Artech House; 1998
- [6] Maas S. Nonlinear Microwave and RF Circuits. 2nd ed. Norwood, MA: Artech House; 2003
- [7] Pedro J, Carvalho N. Intermodulation Distortion in Microwave and Wireless Circuits. 1st ed. Norwood MA: Artech House; 2003
- [8] Kværnø A. Stability of multirate Runge-Kutta schemes. International Journal of Differential Equations and Applications. 2000;1A:97-105
- [9] Günther M, Kværnø A, Rentrop P. Multirate partitioned Runge-Kutta methods. BIT. 2001; 41(3):504-514. DOI: 10.1023/A:1021967112503
- [10] Bartel A, Günther M, Kværnø A. Multirate methods in electrical circuit simulation. Progress in Industrial Mathematics, Springer. 2002;1:258-265. DOI: 10.1007/978-3-662-04784-2_35
- [11] Oliveira J, Araújo A. Envelope transient simulation of nonlinear electronic circuits using multirate Runge-Kutta algorithms. WSEAS Transactions on Electronics. 2006;3(2):77-84
- [12] Brachtendorf H, Welsch G, Laur R, Bunse-Gerstner A. Numerical steady-state analysis of electronic circuits driven by multi-tone signals. Electrical Engineering (Springer-Verlag). 1996;79(2):103-112. DOI: 10.1007/BF01232919
- [13] Roychowdhury J. Analyzing circuits with widely separated time scales using numerical PDE methods. IEEE Transactions on Circuits and Systems. 2001;48(5):578-594. DOI: 10.1109/81.922460
- [14] Mei T, Roychowdhury J, Coffey T, Hutchinson S, Day D. Robust stable time-domain methods for solving MPDEs of fast/slow systems. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems. 2005;24(2):226-239. DOI: 10.1109/TCAD.2004.841073
- [15] Zhu L, Christoffersen C. Adaptive harmonic balance analysis of oscillators using multiple time scales. In: 3rd International IEEE Northeast Workshop on Circuits and Systems; Québec City. 2005. pp. 187-190
- [16] Oliveira J, Pedro J. An efficient time-domain simulation method for multirate RF nonlinear circuits. IEEE Transactions on Microwave Theory and Techniques. 2007;55(11): 2384-2392. DOI: 10.1109/TMTT.2007.908679
- [17] Oliveira J, Pedro J. A multiple-line double multirate shooting technique for the simulation of heterogeneous RF circuits. IEEE Transactions on Microwave Theory and Techniques. 2009;57(2):421-429. DOI: 10.1109/TMTT.2008.2011228

- [18] Oliveira J, Pedro J. A new mixed time-frequency simulation method for nonlinear heterogeneous multirate RF circuits. In: Proceeding of the IEEE MTT-S International Microwave Symposium (MTT '10); May 2010; Anaheim, CA. pp. 548-551
- [19] Oliveira J, Pedro J. Efficient RF circuit simulation using an innovative mixed time-frequency method. IEEE Transactions on Microwave Theory and Techniques. 2011;59(4): 827-836. DOI: 10.1109/TMTT.2010.2095035
- [20] Rizzoli V, Neri A, Mastri F. A modulation-oriented piecewise harmonic-balance technique suitable for transient analysis and digitally modulated analysis. In: Proc. 26th European Microwave Conference; Jun. 1996; Prague. 1996. pp. 546-550
- [21] Sharrit D. Method for simulating a circuit. U.S. Patent 5588142, December 24, 1996

