Fast and Stable Transient Simulation of Nonlinear Circuits Using the Numerical Inversion of the Laplace Transform

Bardia Bandali, *Graduate Student Member, IEEE*, Emad Gad, *Senior Member, IEEE*, and Michel Nakhla, *Life Fellow, IEEE*

***Abstract*— This article outlines a novel approach for simulating general nonlinear circuits in the time-domain. The proposed approach can be considered as the generalization of the numerical inversion Laplace transform (NILT). NILT has been used for circuits with only linear elements. The new approach enables the well-known advantages of NILT such as the guaranteed numerical stability and the high-order approximation, to be carried over to the domain of nonlinear circuit simulations. Numerical examples are given to demonstrate the validity and efficiency of the proposed method.**

***Index Terms*— *A*-stability, general circuit simulation, high-order integration methods, *L*-stability, numerical solution of differential equations.**

NOMENCLATURE

$\mathbf{E}_N$	Identity matrix $N \times N$ and its ith column.
n	Number of terminals in a nonlinear device.
$\mathbf{u}(t) \in \mathbb{R}^n$	auxiliary voltage sources.
$\mathbf{i}(t), \mathbf{j}(t) \in \mathbb{R}^n$	Currents in auxiliary sources in the linear and nonlinear partitions, respectively.
$\hat{t}$, $(\hat{s})$	Scaled-shifted t (the Laplace counterpart).
$\mathscr{L}^{-1}\{\cdot\}(t)$	Inverse Laplace transform.
$\mathscr{L}^{-1}_{\text{NILT}}\{\cdot\}(t)$	NILT-based inverse Laplace transform.
$\hat{\mathbf{i}}_l^{(m)} \in \mathbb{R}^n$	NILT-based computation of $h^m((d^m\mathbf{i}(t))/dt^m)\vert_{t=lh}$.
$\hat{\mathbf{j}}_l^{(m)} \in \mathbb{R}^n$	RT computation of $h^m((d^m\mathbf{j}(t))/dt^m)$ at $t = lh$.
$\hat{\mathscr{L}}^{-1}_{\text{NILT}}\{\cdot\}(\hat{t})$	NILT-based inverse Laplace $\hat{s}$ to $\hat{t}$.
$R_{N,M}(z)$	Padé approximation to e^z (order N/M).

I. INTRODUCTION

THE current drive in integrated circuit technology for smaller structure, higher density, and higher operating frequencies have made the modeling and simulation approach a critical stage in the design process. Complexity of the circuits used to model the electronic packages and electrical interconnects structure have increased steadily, and has become a significant challenge to circuit simulation. The circuit models are typically developed from a variety of approaches. For example, the partial equivalent element circuit (PEEC) [1], [2] is one such approach where 3-D full-wave electromagnetic (EM) wave propagation in arbitrarily-shaped conductors and media, is captured using lumped circuit components.

Time-domain simulation is a task commonly invoked in the design cycle of those circuits. This task is typically carried out using differential equations solvers, which approximates the solution of the differential equations that model the circuits, numerically, at a set of discrete time points [3], [4]. Traditional solvers employed in time-domain circuit simulation belong to class of methods known as the linear multistep methods (LMS), with the Gear's method, trapezoidal rule (TR) and backward Euler (BE) being among the most commonly used in the context of circuit simulations [5], mainly due to the ease with which they can handle the nonlinear differential equations that model large integrated circuits, [6], [7]. Notwithstanding the popularity of these methods, it has been a well-established fact that the LMS family inherently suffers from a conflict between the approximation order and the numerical stability [8]. This conflict essentially forces the simulator to use the low-order, and less efficient, version of LMS method, in order to preserve the numerical stability in the time marching process [9].

One approach that is immune to the above conflict and that has also shown a great potential in handling time-domain simulation of such complex circuits is based on the idea of numerical inversion of Laplace transform (NILT) [10, Ch. 10]. NILT is essentially a time-domain approach that is fundamentally grounded in the frequency-domain. As a result of this feature, it offers the rare advantage in being more suitable to handle those circuits whose physics are captured more easily in the frequency-domain than in the time-domain. Coupled transmission lines (TLs) modeled as distributed interconnect serve as an example of those circuits.

Although NILT was first introduced several decades ago, the recent developments in model-order reduction techniques [11], [12] and the complex full-wave circuit modeling

Manuscript received 22 February 2022; revised 28 April 2022 and 3 June 2022; accepted 7 June 2022. Date of publication 10 June 2022; date of current version 26 July 2022. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. Recommended for publication by Associate Editor P. Triverio upon evaluation of reviewers' comments. *(Corresponding author: Emad Gad.)*

Bardia Bandali and Emad Gad are with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: egad@eecs.uottawa.ca).

Michel Nakhla is with the Department of Electronics, Carleton University, Ottawa, ON K1S 5B6, Canada.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCPMT.2022.3182136.

Digital Object Identifier 10.1109/TCPMT.2022.3182136

approaches developed using PEEC [2], [13] have shed new lights on its efficiency, usefulness and renewed interest in employing it as an engine in general circuit simulation. In fact, for PEEC circuits, it was shown that the NILT approach proved to be more reliable compared to the time stepping of the conventional methods [14].

Nonetheless, the well-known significant potentials of NILT remain mainly limited to simulating circuits with only linear elements. The main reason for this is that NILT, being founded on the Laplace integral transformation, does not offer an easy representation of nonlinear circuit elements in the Laplace or complex domain.

The main goal of this article is to address this issue through introducing a new approach that allows incorporating arbitrarily complex nonlinear elements within the NILT basic approach. The proposed approach brings all the desirable features of NILT, such as the stability and high-order approximations to bear on simulation of general nonlinear circuits. The proposed approach is mainly motivated by the stability and efficiency of NILT in handling delay differential equations, such as those modeling the PEEC circuits, and the desire to extend this performance to circuits with nonlinear terminations.

The proposed approach works by splitting the circuit along the ports of the nonlinear devices, treating the linear partition using NILT and the nonlinear partition using the notion of rooted trees. The splitting concept was outlined in [15]. This article describes the full approach by developing the techniques of computing the derivatives of the port voltages/currents in both partitions. It also derives and demonstrates the underlying theoretical basis for its performance characteristics.

The article is organized as follows. Section II provides the background and mathematical formulation highlighting the scope and motivation of the work. Section III describes a high-level view that outlines the development of the proposed approach. Section IV follows the development with a fully detailed description of the various computational blocks outlined in Section III. Section V presents the theoretical characterization for the performance of the proposed approach. Section VI describes the application of the proposed approach in settings more general than those used to present the development in the previous sections. Finally, Sections VII and VIII report the numerical simulations and the concluding remarks, respectively.

II. Background and Scope of the Contribution

The mathematical formulation of a general linear circuit is often done through the modified nodal approach (MNA) resulting in a system of mixed differential-algebraic equations taking the following form

$$\mathbf{C}\frac{d\mathbf{x}(t)}{dt} + \mathbf{G}\mathbf{x}(t) = \mathbf{b}(t) \tag{1}$$

where $\mathbf{C}, \mathbf{G} \in \mathbb{R}^K$ are sparse matrices that describe the memory and memoryless elements in the circuit, $\mathbf{x}(t) \in \mathbb{R}^K$ is a vector of circuit nodes voltages and currents in inductors and voltage sources, and $\mathbf{b}(t) \in \mathbb{R}^K$ is the vector representing the independent (voltages and current) stimuli, and K is the size of the MNA formulation.

The task of simulating the circuit transient response is performed through computing the vector $\mathbf{x}(t)$ at a set of discrete time points, e.g. $t = lh$, where $l = 0, 1, 2, \ldots$ and h is a given time step. Numerous methods for solving differential equations have been proposed to carry out this task numerically. Methods typically used in the domain of circuit simulation are based on a general class known as the linear multistep (LMS), where methods such as the BE, TR, backward differentiation formulae (BDF), and the Gear's method count among the most notable approaches.

In spite of the popularity of those methods within the community of circuit simulation, their limitations have been known for quite some time. One of the major limitations in these methods is the so-called conflict between order and stability. This conflict forces the method to be executed at a low order in the interest of guaranteeing the numerical stability. However, using low-order negatively impacts the efficiency or speed of the method as it forces reducing the step of time marching (h) to maintain the accuracy required by the user. It should be added here that a method free from this conflict was presented in [16]. The stability of this method emanates from utilizing the so-called Obreshkov formula as a constraint on the time stepping.

One approach, which was developed independently of LMS, is based on the idea of numerical inversion of the Laplace transform (NILT). NILT was proven to be free of the order-stability conflict. A brief review of NILT is presented next to establish the necessary background to the work proposed in this article.

A. Review of NILT in Circuit Simulation

The NILT approach is used to compute $\mathbf{x}(t)$ at $t = lh$. It starts by putting the MNA formulation (1) in the Laplace-domain

$$(s\mathbf{C} + \mathbf{G})\mathbf{X}(s) = \mathbf{B}(s) + \mathbf{C}\mathbf{x}(0) \tag{2}$$

where $\mathbf{X}(s) = \mathscr{L}\{\mathbf{x}(t)\}$, $\mathbf{B}(s) = \mathscr{L}\{\mathbf{b}(t)\}$, $\mathbf{x}(0)$ is the initial conditions ($t = 0$) for $\mathbf{x}(t)$, and $\mathscr{L}$ and $s \in \mathbb{C}$ denote the Laplace operation and complex variable, respectively. The *exact* $\mathbf{x}(t)$ is recovered using the inverse Laplace-domain operation given by the infinite line integral

$$\mathbf{x}(t) = \mathscr{L}^{-1}\{\mathbf{X}(s)\} = \frac{1}{2\pi \mathrm{J}} \int_{c-\mathrm{J}\infty}^{c+\mathrm{J}\infty} \mathbf{X}(s)\mathrm{e}^{st} ds \tag{3}$$

where $\mathrm{J} = \sqrt{-1}$ and c is a constant chosen such that all singularities in $\mathbf{X}(s)$ lie on the left of the vertical line $\Re(s) = c$, with $\Re(\cdot)$ denoting the real part of a complex number. The integral in (3) is a line integration performed in the complex domain along infinite vertical line $\Re(s) = c$.

The NILT approach proceeds by replacing st with z, e^z with its $[N/M]$ Padé approximant $\xi_{N,M}(z)$ [17], [18] and then, using Cauchy theorem of residues, exchanges the line integral with a finite summation of the residues of $\mathbf{X}(z/t)\xi_{N,M}(z)$ at the poles of the $\xi_{N,M}(z)$. Those steps, combined, and denoted

as $\mathscr{L}^{-1}_{\text{NILT}}\{\mathbf{X}(s)\}(t)$, provide an approximation, $\tilde{\mathbf{x}}(t)$, to $\mathbf{x}(t)$ defined through

$$\tilde{\mathbf{x}}(t) = \mathscr{L}^{-1}_{\text{NILT}}\{\mathbf{X}(s)\}(t) := -\frac{1}{t}\sum_{i=1}^{M/2} 2\Re\left\{k_i \mathbf{X}\left(\frac{z_i}{t}\right)\right\} \tag{4}$$

for even M, with k_i and z_i, $i = 1, \ldots, M$ being, respectively, the residues and poles of the partial fraction expression of $\xi_{N,M}(z)$. Note that for even values of M, poles and residues appear in complex-conjugate pairs, and for odd M, only one residue-pole duo (k_0, z_0) is real, with the rest being in complex-conjugate pairs. In the rest of the article, M will be assumed even, knowing that the case of odd M can be treated analogously. The theory underlying the formulation of (4) is detailed in [10, Ch. 10].

Implementation of NILT in (4) is known as the *initial time mode*, where the initial condition of the circuit at $t = 0$, $\mathbf{x}(0)$, is used in computing $\mathbf{X}(s)$ from (2) at $s = (z_i/t)$. The NILT approach can also be used in a *marching* mode, by allowing the value computed for $\tilde{\mathbf{x}}(t)$ at certain time, say $t = lh$, to be used *as the initial condition* in advancing to a new time point at $t = lh$ with an appropriate step size h, with $l = 1, 2, \ldots$ This is done by shifting the time origin to $(l-1)h$, and scaling the time t by dividing by h leading to the definition of the new time variable $\hat{t}$ given by $\hat{t} = (t/h) - (l-1)$.

If the Laplace transform is applied with respect to the $\hat{t}$ variable, the Laplace operator and its inverse will be denoted by $\widehat{\mathscr{L}}\{\cdot\}(\hat{s})$ and $\widehat{\mathscr{L}}^{-1}\{\cdot\}(\hat{t})$, respectively, where $\hat{s} = hs$. Likewise, the NILT-based realization to the latter will be given the notation $\widehat{\mathscr{L}}^{-1}_{\text{NILT}}\{\cdot\}(\hat{t})$ and the value obtained from it will be denoted $\hat{\mathbf{x}}(\hat{t})$. Hence, by this notation we have

$$\tilde{\mathbf{x}}(t = (l-1)h) \equiv \hat{\mathbf{x}}(\hat{t} = 0), \quad \tilde{\mathbf{x}}(t = lh) \equiv \hat{\mathbf{x}}(\hat{t} = 1) \tag{5}$$

$$\hat{\mathbf{x}}(1) = \widehat{\mathscr{L}}^{-1}_{\text{NILT}}\{\mathbf{X}(\hat{s})\}(\hat{t})\big|_{\hat{t}=1} = -\sum_{i=1}^{M/2} 2\Re\left\{k_i \hat{\mathbf{X}}(z_i)\right\} \tag{6}$$

where $\hat{\mathbf{X}}(z_i)$ is obtained from

$$\hat{\mathbf{X}}(z_i) = \left(\frac{z_i}{h}\mathbf{C} + \mathbf{G}\right)^{-1}\left(\hat{\mathbf{B}}(z_i) + \frac{1}{h}\mathbf{C}\tilde{\mathbf{x}}(0)\right) \tag{7}$$

and $\hat{\mathbf{B}}(\hat{s})$ is the Laplace transform of $\mathbf{b}(h\hat{t} + (l-1)h)$ taken with respect to $\hat{t}$, and computed at $\hat{s} = z_i$, while $\tilde{\mathbf{x}}_{l-1}$ is a short for $\tilde{\mathbf{x}}(t)$ at $t = (l-1)h$.

The time marching process of computing $\hat{\mathbf{x}}(1)$ from $\hat{\mathbf{x}}(0)$, (respectively, $\tilde{\mathbf{x}}_l$ from $\tilde{\mathbf{x}}_{l-1}$) using (6) and (7), is also known as reinitializing NILT. In the reinitialization framework, the solution from the previous step serves as the initial condition for the next step.

B. Main NILT Characteristics

The approximation $\tilde{\mathbf{x}}(t)$ is related to the exact $\mathbf{x}(t)$ through

$$\tilde{\mathbf{x}}(t) - \mathbf{x}(t) = \Psi_{N,M}\frac{d^{N+M+1}}{dt^{N+M+1}}\mathbf{x}(t)\bigg|_{t=0} t^{N+M+1} + \mathcal{O}\left(t^{N+M+2}\right) \tag{8}$$

where $\Psi_{N,M} = (-1)^M M!N!/((M+N)! \cdot (N+M+1)!)$ [19]. The choice of the integers N and M is done based on satisfying two main requirements. The first requirement is that the $\mathbf{X}(s)\xi_{N,M}(s)$ should tend to 0 as $s \to \infty$ to allow for the use of the Cauchy theorem of residues, while the second requirement is pertinent to guaranteeing the numerical stability in marching through time. It has been shown that both requirements are satisfied if, and only if, $N = M - 2$ [10, Ch. 10].

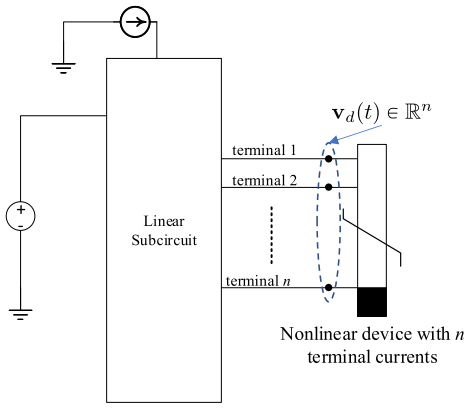

Fig. 1. Example circuit used to illustrate the proposed approach.

C. Scope of the Proposed Contribution

General circuits involve nonlinearity to model devices such as logical gates, output buffers and driver circuits. In this case the MNA needs to be modified to account those nonlinearity by adding a term $\mathbf{f}(\mathbf{x}(t))$

$$\mathbf{C}\frac{d\mathbf{x}(t)}{dt} + \mathbf{G}\mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t)) = \mathbf{b}(t). \tag{9}$$

In the above MNA formulation $\mathbf{f}(\mathbf{x}(t)) : \mathbb{R}^K \to \mathbb{R}^K$ is a mapping that describes an algebraic (memoryless) nonlinearity. A memory (capacitive/inductive) nonlinearity can be accommodated in a similar manner through using the charge/flux oriented MNA formulation which appends the charge/flux in the nonlinear capacitor/inductors to the list of the MNA variables $\mathbf{x}(t)$ [5]. Therefore, without any loss of generality, we treat in this work the algebraic nonlinearity.

It can be seen from (9) that in the presence of nonlinearity in the circuit the NILT framework cannot be applied in this situation since the nonlinear term precludes a Laplace-domain representation of the MNA formulation. The goal of this work is to address this difficulty by presenting an approach that relies on the core ideas of NILT while preserving its desirable characteristics.

III. Overview of the Proposed Approach

To facilitate the description of the proposed approach, we first consider its application to a relatively simple circuit which is depicted by Fig. 1. Key ideas and notations will be established through this example, paving the way to present the treatment of the general circuits in Section VI.

The schematic in Fig. 1 shows a circuit that consists of a large general linear circuit driven by multiple independent sources and is attached to a single nonlinear device.

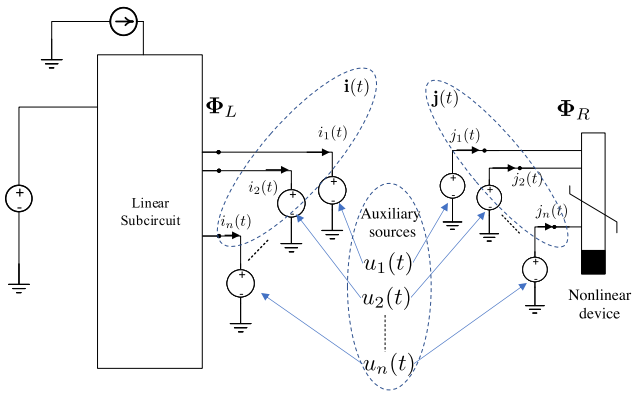

Fig. 2. Splitting the circuit of Fig. 2 and inserting the auxiliary sources.

The representation of the nonlinear device in this circuit serves as vehicle to illustrate the handling of an arbitrary nonlinear element. It does so by assuming that the device is controlled by n voltages at its terminals and that the currents in those terminals are functions of those voltages captured by a mapping $\Phi_R\colon \mathbb{R}^n \to \mathbb{R}^n$. The voltages at the device terminals, denoted $\mathbf{v}_d(t) \in \mathbb{R}^n$ constitute a subset of the MNA variables $\mathbf{x}(t)$ and are expressed in terms of $\mathbf{x}(t)$ through the use of an incidence matrix $\mathbf{d} \in \{0,1\}^{K\times n}$ that selects the specific n nodes representing the terminals of the device, that is,

$$\mathbf{v}_d(t) = \mathbf{d}^\top \mathbf{x}(t). \tag{10}$$

The proposed approach starts by splitting the circuit in Fig. 1 into two circuits along the n terminals and inserting a set of $2n$ voltage sources (henceforth referred to as *auxiliary sources*) to obtain the two circuits shown in Fig. 2. The two sets of auxiliary sources are assumed to be identical, that is, the n sources on the left- and those n sources on right-side of the split are assumed to have the same temporal waveforms which are denoted by $u_r(t), r = 1,\ldots,n$. For arbitrary $u_r(t)$, the currents in those (auxiliary) sources are, in general, not equal. Thus, the currents in the left-side of the split are denoted by $i_r(t)$ while those on the right-side are denoted by $j_r(t)$, with $r = 1,\ldots,n$. Grouping the set of voltages $u_r(t)$ and currents $j_r(t)$ and $i_r(t)$ in vectors denoted by $\mathbf{u}(t)$, $\mathbf{j}(t)$ and $\mathbf{i}(t) \in \mathbb{R}^n$, respectively, we write

$$\mathbf{i}(t) = \mathbf{\Phi}_L(\mathbf{u}(t)), \quad \mathbf{j}(t) = \mathbf{\Phi}_R(\mathbf{u}(t)) \tag{11}$$

where $\mathbf{\Phi}_R$ is a the nonlinear mapping $\mathbb{R}^n \to \mathbb{R}^n$, and $\mathbf{\Phi}_L$ is a linear operator, as a result of the linearity of the circuit on the left-side.

The objective of the proposed approach is to carry out the simulation of the entire circuit (without splitting the nonlinear device) in the time-domain using the NILT approach. Given the difficulty of incorporating the nonlinear device in a Laplace-domain-based framework, the idea of splitting circuit as described above will be the principal tool to achieve this objective, as shown next.

The key idea in the proposed approach seeks to compute the auxiliary sources $\mathbf{u}(t)$ that enforce the equality $\mathbf{i}(t) = \mathbf{j}(t)$. Naturally, in the context of computer simulation, this condition can be enforced at discrete time points, t_l, $l = 1, 2, \ldots$, with variable step size $h_l = t_l - t_{l-1}$. To simplify the notation, we shall assume a constant step size, i.e., that $t_l = lh$, h being an appropriate time step. Nonetheless, enforcing a simple equality at discrete time points, i.e., $\mathbf{i}(lh) = \mathbf{j}(lh)$ *is not* sufficient to bring over the numerical advantages of NILT approach to the problem at hand. Instead of enforcing simple equality in the currents, the proposed approach does enforce the equality in the high-order derivatives (up to a given order $p-1$) of the currents at those discrete time points. In other words, the proposed approach casts the problem as that of computing the auxiliary sources $\mathbf{u}(t)$ such that

$$\text{Compute } \mathbf{u}(t) \quad \text{s.t. } \hat{\mathbf{i}}_l^{(m)} = \hat{\mathbf{j}}_l^{(m)}, \quad \begin{cases} l = 0, 1, 2, 3, \ldots \\ m = 0, 1, 2, \ldots, p-1 \end{cases} \tag{12}$$

where[1]

$$\hat{\mathbf{i}}_l^{(m)} = h^m \left.\frac{d^m \mathbf{i}(t)}{dt^m}\right|_{t=lh}, \quad \hat{\mathbf{j}}_l^{(m)} = h^m \left.\frac{d^m \mathbf{j}(t)}{dt^m}\right|_{t=lh}. \tag{13}$$

As will be shown in Section V, enforcing a simple equality of the currents at discrete points, i.e., by setting $p = 1$, in (12), will *effectively* reduce M to 1, even though the actual implementation may have been carried out with a value $M > 1$. It should also be mentioned here that the condition enforced by (12) implies that the interface voltage waveforms need to satisfy a certain smoothness requirement, with derivatives up to order p being defined and computable. This requirement was found to be satisfied for all circuits used in the experimental testing in this work.

The system in (12) represents a system of np nonlinear equations whose solution, using the Newton-Raphson (NR) method at $l = 0, 1, 2, 3, \ldots$ is used in the proposed approach to steer, so to speak, the simulation of the entire (non-split) circuit in the time domain. The computation of the currents in the linear subnetwork and their derivatives $\hat{\mathbf{i}}_l^{(m)}$ will be carried out using the NILT approach, while the currents in the nonlinear elements and their derivatives $\hat{\mathbf{j}}_l^{(m)}$ will be addressed through the idea of rooted trees.

As quick illustration of the splitting technique, we use the simple diode circuit on the left in Fig. 3 to show how the splitting is carried out producing two separate circuits, on the right, coupled through a single auxiliary source.

Before delving into the details of solving (12) using NR, a brief note on the relation between the proposed technique and earlier approaches related to ideas based on circuit splitting is in order.

A. Relation to Other Circuit Splitting Approaches

The idea of partitioning a circuit into smaller subcircuits is well-known in circuit [20]–[22] and EM simulations [23]. It has been used under different names such as domain decomposition, waveform relaxation [24], [25], tearing and diakoptic analysis. Those methods are motivated by the fact that in many situations the simulations of the subcircuits individually

[1]Scaling the derivatives by powers of h in (13) is meant to improve the numerical conditioning of the resulting equations and prevent the exponential growth in the numerical analysis.

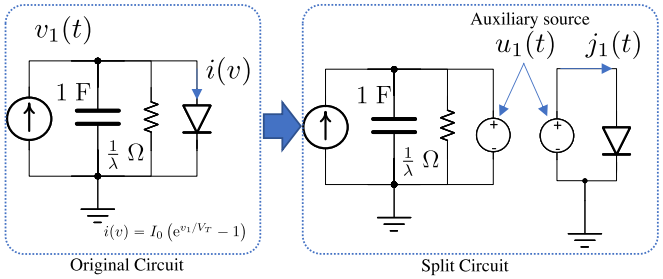

Fig. 3. Simple circuit and its split parts.

is easier than the simulation of the whole, or can be run in parallel to achieve speedups. This idea was implemented in several approaches, such as branch tearing, node tearing, and hierarchical decomposition [26]–[30]. Nonetheless, and in spite of the wide variety of those techniques, they all shared a common approach to simulate the response of the whole circuit based on the simulation of the subcircuits, namely, through enforcing a simple (zero-order) equality on the boundary variables.

Seen from that perspective, those approaches can be considered as a special case of the proposed approach, that is, when implemented with $p = 1$ in (12). Indeed, as will be shown in Section V, the ability to explicitly compute and enforce the equality of higher-order derivatives for the port voltages represents the key feature that enables the proposed approach to inherit the main NILT characteristics (high order and numerical stability).

IV. Detailed Development

An outline for the steps involved in implementing the NR can be summarized in the following points.

1) The proposed approach executes the time stepping used in conventional simulators through enforcing the set of constraints (12) at each time step. This requires defining a set of independent variables that can be iteratively updated, in the NR framework, to satisfy those constraints. Moreover, to formulate a well-defined mathematical problem, the number of those variables should be equal to the number of constraints in (12). Those variables, denoted by $\boldsymbol{\xi}_l^{[i]}$, i being an index for the NR iteration, will be associated with a set of electrical variables that will be identified in Section IV-A. As will be shown below, $\boldsymbol{\xi}_l^{[i]}$ are used to derive the auxiliary sources $\mathbf{u}(t)$.
2) Deriving waveforms of the auxiliary sources $\mathbf{u}(t)$ from the independent variables $\boldsymbol{\xi}_l^{[i]}$ as will be illustrated in Section IV-B.
3) Establishing the process of computing $\hat{\mathbf{i}}_l^{(m)}$ and $\hat{\mathbf{j}}_l^{(m)}$ from the auxiliary sources $\mathbf{u}(t)$. This process is expanded in Section IV-C using the basic NILT paradigm and in Section IV-D and Appendix A using the notion of rooted trees, respectively.
4) Enforcing a set of constraints, in Section IV-E, on the search for independent variables which will create the sufficient conditions guaranteeing that the stability and order of the proposed method will match the NILT stability and order.
5) Defining the updating process of the independent variables $\boldsymbol{\xi}_l^{[i]}$. This process requires computing the Jacobian matrix of both $\hat{\mathbf{i}}_l^{(m)}$ and $\hat{\mathbf{j}}_l^{(m)}$ with respect to those variables and will be addressed in Section IV-F.

To proceed with the above development, and show how to use NR iterative method to find $\mathbf{v}_d(lh)$, it will be assumed that all the NR processes launched in the past $l-1$ time steps have converged producing the values of all node voltages including the node voltages of the device, as well as its derivatives, up to the qth order derivative. In other words, $\hat{\mathbf{v}}_l^{(j)}$, $j = 0, \ldots, q$ are assumed readily available.[2] Similar to notations used earlier, the derivatives of the device voltages at $t = (l-1)h$ and $t = lh$ are called $\hat{\mathbf{v}}_{l-1}^{(m)}$ and $\hat{\mathbf{v}}_l^{(m)}$, respectively, that is,

$$\hat{\mathbf{v}}_{l-1}^{(m)} = h^m \left.\frac{d^m \mathbf{v}_d(t)}{dt^m}\right|_{t=(l-1)h}, \quad \hat{\mathbf{v}}_l^{(m)} = h^m \left.\frac{d^m \mathbf{v}_d(t)}{dt^m}\right|_{t=lh}.$$

A. Defining the Independent Variables $\boldsymbol{\xi}_l^{[i]}$

While $\hat{\mathbf{v}}_{l-1}^{(m)}$, $m = 0, 1, \ldots, q$ are assumed to be available upon the convergence of the NR method at the previous time step $t = (l-1)h$, the values of $\hat{\mathbf{v}}_l^{(m)}$, for $m = 0, 1, \ldots, p$ remain unknowns that need to be determined such that (12) are satisfied.

In fact, the independent variables $\boldsymbol{\xi}_l^{[i]}$ are taken to represent $\hat{\mathbf{v}}_l^{(m)}$, for $m = 0, 1, \ldots, p$. For now, it will be assumed that $p \geq q$. Section V will derive the sufficient condition on the relation between q and p that guarantees the numerical stability and high-order of the proposed method.

As it is typical in the normal NR iterative process, $\boldsymbol{\xi}_l^{[0]}$ is given an initial "guess." This initial guess is based on the previous time point, that is,

$$\boldsymbol{\xi}_l^{[0]} \leftarrow \left[\hat{\mathbf{v}}_{l-1}^{(0)\top}, \hat{\mathbf{v}}_{l-1}^{(1)\top}, \ldots, \hat{\mathbf{v}}_{l-1}^{(q)\top}, \mathbf{0}, \ldots, \mathbf{0}\right]^\top. \tag{14}$$

Upon convergence of the NR iterations, say after κ iterations, the solution, $\boldsymbol{\xi}_l^{[\kappa]}$, arrived at the κth iteration will be used for $\hat{\mathbf{v}}_l^{(m)}$, $m = 1, 2, \ldots, p$, that is,

$$\left[\hat{\mathbf{v}}_l^{(0)\top}, \hat{\mathbf{v}}_l^{(1)\top}, \ldots, \hat{\mathbf{v}}_l^{(p)\top}\right]^\top \leftarrow \boldsymbol{\xi}_l^{[\kappa]}. \tag{15}$$

B. Deriving the Auxiliary Sources $\mathbf{u}(t)$

The auxiliary sources $\mathbf{u}(t)$ are defined as a piecewise polynomial, that is,

$$\mathbf{u}(t) = \sum_{l=0,1,2,\ldots} \mathbf{u}_l(t), \quad (l-1)h \leq t \leq lh \tag{16}$$

where the coefficients of the polynomial $\mathbf{u}_l(t)$ are selected such that $\mathbf{u}_l(t)$ interpolates $\mathbf{v}_d(t)$ at $t = (l-1)h$ and $t = lh$, i.e., matches $\hat{\mathbf{v}}_{l-1}^{(0)}$ and $\hat{\mathbf{v}}_l^{(0)}$ at these two points. In addition, the polynomial coefficients are also selected so that the first q and p derivatives of $\mathbf{u}_l(t)$, at $t = (l-1)h$ and $t = lh$, respectively, be equal to $\hat{\mathbf{v}}_{l-1}^{(n)}$ and $\hat{\mathbf{v}}_l^{(m)}$, for $n = 1, 2, \ldots, q$

[2]The initial derivatives of the voltages at the terminals of the device at $t = 0$ or $\hat{\mathbf{v}}_0^{(m)}$ are obtained from the dc operating point and does not require NR iterations [16].

and $m = 1, 2, \ldots, p$. This idea is known in the literature as polynomial Hermite interpolation and was shown in [31] that a unique polynomial of degree $p + q + 1$ can be found to satisfy the previous requirements. In addition, this polynomial can be expressed explicitly in the following form

$$\mathbf{u}_l(\hat{t}) = \sum_{j=0}^{q} \frac{\hat{\mathbf{v}}_{l-1}^{(j)}}{j!}(1-\hat{t})^{p+1} \sum_{k=j}^{q} \hat{t}^k \frac{(p+1)_{[k-j]}}{(k-j)!} + \sum_{j=0}^{p} \frac{\hat{\mathbf{v}}_{l}^{(j)}}{j!} \times \hat{t}^{q+1} \sum_{k=j}^{p} (-1)^{k-j} \big(\hat{t}-1\big)^k \frac{(q+1)_{[k-j]}}{(k-j)!} \quad (17)$$

where, for convenience, we used the transformed time variable $\hat{t}$ in place of t and utilized $m_{[n]}$ to denote the Pochhammer symbol defined by by $(((m+n-1)!)/((m-1)!))$.

With (17) showing the computational path from the independent variables $\hat{\mathbf{v}}_l^{(j)}$ to the auxiliary sources $\mathbf{u}_l(\hat{t})$, in the interval $(l-1)h \le t \le lh$, we now move to illustrating the process that takes $\mathbf{u}_l(\hat{t})$ and yields $\hat{\mathbf{i}}_l^{(m)}$ and $\hat{\mathbf{j}}_l^{(m)}$.

C. Computing $\hat{\mathbf{i}}_l^{(m)}$ Using NILT

The currents $\mathbf{i}(t)$ arise in response to the stimulus of the external sources as well as the added auxiliary sources in a linear circuit. Hence, one can utilize the concept of NILT along with the principle of superposition to compute $\hat{\mathbf{i}}_l^{(m)}$, by first applying NILT to compute the portion of $\hat{\mathbf{i}}_l^{(m)}$ arising from the external sources, and then applying NILT to compute the other portion in $\hat{\mathbf{i}}_l^{(m)}$ arising from the auxiliary sources, and then adding the two portions to arrive at the required $\hat{\mathbf{i}}_l^{(m)}$. For the sake of brevity and clarity, we demonstrate in this section computing the response arising only from auxiliary sources, as it will be clearly seen that the approach used with auxiliary sources can be utilized with the external sources in an almost identical manner.

Inserting the auxiliary voltage sources at the ports of the circuit on the left side of the divide in Fig. 2, then (1) shifting the time origin to lh via the change of time variable $t \to h\hat{t} + h(l-1)$ and taking the Laplace transform with respect to $\hat{t}$, the circuit on the left side can then be represented in the Laplace-domain using an augmented MNA formulation

$$\left(\hat{s} \overbrace{\begin{bmatrix} \frac{1}{h}\mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}^{\bar{\mathbf{C}}} + \overbrace{\begin{bmatrix} \mathbf{G} & \mathbf{d} \\ \mathbf{d}^\top & \mathbf{0} \end{bmatrix}}^{\bar{\mathbf{G}}} \right) \overbrace{\begin{bmatrix} \mathbf{X}(\hat{s}) \\ \mathbf{I}_{\text{aux}}(\hat{s}) \end{bmatrix}}^{\bar{\mathbf{X}}(\hat{s})} = \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{U}_l(\hat{s}) \end{bmatrix}}_{\bar{\mathbf{B}}(\hat{s})} + \bar{\mathbf{C}}\bar{\mathbf{x}}(0) \quad (18)$$

where $\mathbf{C}$ and $\mathbf{G}$ are the same matrices representing the linear part of the original circuit (before the splitting action), $\mathbf{I}_{\text{aux}}(\hat{s}) \in \mathbb{C}^n$ is the Laplace-domain currents in the auxiliary sources, $\mathbf{I}_{\text{aux}}(\hat{s}) = \mathscr{L}\{\mathbf{i}(\hat{t})\}$, and $\mathbf{U}_l(\hat{s}) = \widehat{\mathscr{L}}\{\mathbf{u}_l(\hat{t})\}$ and $\bar{\mathbf{X}}(\hat{s}) = \widehat{\mathscr{L}}\{\bar{\mathbf{x}}(\hat{t})\}$, $\bar{\mathbf{x}}(t)$ being the variables of the circuits after augmenting them with the currents in the auxiliary sources.

The NILT computational process can be then be invoked to obtain $\hat{\mathbf{i}}_l^{(m)}$. We first consider the case where $m = 0$.

1) Computing $\hat{\mathbf{i}}_l^{(m)}$ for $m = 0$: Note that by definition

$$\hat{\mathbf{i}}_l^{(0)} = \widehat{\mathscr{L}}_{\text{NILT}}^{-1}\{\mathbf{I}_{\text{aux}}(\hat{s})\}(\hat{t})\big|_{\hat{t}=1} = -\sum_{i=1}^{M/2} 2\Re\left\{k_i \hat{\mathbf{I}}_{\text{aux}}(z_i)\right\} \quad (19)$$

where

$$\hat{\mathbf{I}}_{\text{aux}}(z_i) = \underbrace{[\mathbf{0} \quad \mathbf{E}_n]}_{\tilde{\mathbf{d}}} \big(z_i \bar{\mathbf{C}} + \bar{\mathbf{G}}\big)^{-1} \begin{bmatrix} \frac{1}{h}\mathbf{C}\tilde{\mathbf{x}}_{l-1} \\ \mathbf{U}_l(z_i) \end{bmatrix}. \quad (20)$$

2) Computing $\hat{\mathbf{i}}_l^{(m)}$ for $m > 0$: This case is handled through the following identity of the Laplace transform

$$\frac{d^m \mathbf{i}(\hat{t})}{d\hat{t}^m} = \mathscr{L}^{-1}\left\{ \hat{s}^m \mathbf{I}_{\text{aux}}(\hat{s}) - \sum_{j=1}^{m} \left.\frac{d^{j-1}\mathbf{i}(\hat{t})}{d\hat{t}^{j-1}}\right|_{\hat{t}=0} \hat{s}^{m-j} \right\}(\hat{t}). \quad (21)$$

Note that through the change of variables $t = h\hat{t} + (l-1)h$, we have

$$\left.\frac{d^m \hat{\mathbf{i}}(\hat{t})}{d\hat{t}^m}\right|_{\hat{t}=1} = h^m \left.\frac{d^m \mathbf{i}(t)}{dt^m}\right|_{t=lh}. \quad (22)$$

Noting the definition of $\hat{\mathbf{i}}_l^{(m)}$ in (13) and invoking the NILT process to express the right side of (21), then $\hat{\mathbf{i}}_l^{(m)}$ can be expanded as shown in the following steps

$$\begin{aligned} \hat{\mathbf{i}}_l^{(m)} &= \widehat{\mathscr{L}}_{\text{NILT}}^{-1}\left\{ \hat{s}^m \hat{\mathbf{I}}_{\text{aux}}(\hat{s}) - \sum_{j=1}^{m} \left.\frac{d^{j-1}\hat{\mathbf{i}}(\hat{t})}{d\hat{t}^{j-1}}\right|_{\hat{t}=0} \hat{s}^{m-j} \right\}\Bigg|_{\hat{t}=1} \\ &= -\sum_{i=1}^{M/2} 2\Re\left\{k_i z_i^m \hat{\mathbf{I}}_{\text{aux}}(z_i)\right\} \\ &\quad + \sum_{i=1}^{M/2}\sum_{j=1}^{m-1} 2\Re\left\{k_i z_i^{m-j}\, \hat{\mathbf{i}}_{l-1}^{(j-1)}\right\} + \hat{\mathbf{i}}_l^{(m)} \overbrace{\sum_{i=1}^{M/2} \Re(k_i)}^{0} \\ &= -\sum_{i=1}^{M/2} 2\Re\left\{k_i z_i^m \hat{\mathbf{I}}_{\text{aux}}(z_i) - k_i \sum_{j=1}^{m-1} z_i^{m-j}\, \hat{\mathbf{i}}_{l-1}^{(j-1)}\right\} \end{aligned} \quad (23)$$

where the property of NILT that $\sum_{i=1}^{M/2} k_i = 0$ was used in the second step [10, Ch. 10].

It will be more convenient for the sake of conciseness of the following derivations to assemble the relation between the independent variables $\hat{\mathbf{v}}_l^{(j)}$, $j = 0, 1, \ldots, p$ and the $\hat{\mathbf{i}}_l^{(m)}$, $m = 0, 1, \ldots, p-1$ in a single expression. This can be done by first writing the Laplace transform expression of $\mathbf{u}_l(\hat{t})$ with respect to $\hat{t}$, explicitly in terms of $\hat{s}$. To this end, we define $\theta_{l-1,j}(\hat{s})$ and $\theta_{l,j}(\hat{s})$, through

$$\theta_{l-1,j}(\hat{s}) = \frac{1}{j!}\sum_{\upsilon=0}^{p+1}\sum_{k=j}^{q} \frac{(-1)^\upsilon (p+1)!(k+\upsilon)!(p+1)_{[k-j]}}{(p+1-\upsilon)!(k-j)!\upsilon!\hat{s}^{k+\upsilon+1}}$$

$$\theta_{l,j}(\hat{s}) = \frac{1}{j!}\sum_{k=j}^{p}\sum_{\upsilon=0}^{k} \frac{(-1)^{\upsilon+j}(\upsilon+q+1)!k!(q+1)_{[k-j]}}{(k-\upsilon)!(k-j)!\upsilon!\hat{s}^{\upsilon+q+2}}. \quad (24)$$

The above two expressions are obtained from taking the Laplace transformations for the jth term under the two summations in (17). Based on that, $\mathbf{U}_l(\hat{s})$ can be written as

$$\mathbf{U}_l(\hat{s}) = \sum_{j=0}^{q} \theta_{l-1,j}(\hat{s})\hat{\mathbf{v}}_{l-1}^{(j)} + \sum_{j=0}^{p} \theta_{l,j}(\hat{s})\hat{\mathbf{v}}_{l}^{(j)}. \tag{25}$$

Now $\hat{\mathbf{i}}_l^{(m)}$ can be explicitly expressed in terms of $\hat{\mathbf{v}}_l^{(j)}$ (26), as shown at the bottom of the page.

For the illustrative circuit used in Fig. 3, we would have $\mathbf{d} = 1$, $\bar{\mathbf{d}} = [1\ 0]$ $\mathbf{G} = \lambda$, $\mathbf{C} = 1$, $\bar{\mathbf{C}} = \left[\begin{smallmatrix} 1/h & 0 \\ 0 & 0 \end{smallmatrix}\right]$, $\bar{\mathbf{G}} = \left[\begin{smallmatrix} \lambda & 1 \\ 1 & 0 \end{smallmatrix}\right]$ and $\left(z_i\bar{\mathbf{C}} + \bar{\mathbf{G}}\right)^{-1} = \left[\begin{smallmatrix} 0 & 1 \\ 1 & -\frac{z_i}{h} \end{smallmatrix}\right]$.

D. Computing $\hat{\mathbf{j}}_l^{(m)}$

Deriving the computational process used to obtain $\hat{\mathbf{j}}_l^{(m)}$, $m = 0, 1, 2, \ldots, p-1$ from $\hat{\mathbf{v}}_l^{(n)}$, $n = 0, 1, 2, \ldots, p$ is done by first recasting $\mathbf{u}_l(\hat{t}\,)$ of (17) in the form

$$\mathbf{u}_l(\hat{t}\,) = \sum_{r=0}^{p+q+1} \hat{\mathbf{u}}_{l,r}\ \left(\hat{t}-1\right)^r. \tag{27}$$

Notice that, by the construction of $\mathbf{u}_l(\hat{t}\,)$, $\hat{\mathbf{u}}_{l,r} = (1/r!)\hat{\mathbf{v}}_l^{(r)}$, for $r = 0, 1, 2, \ldots, p$. Computation of $\hat{\mathbf{j}}_l^{(m)}$ from $\hat{\mathbf{v}}_l^{(r)}$ is performed using notion of Rooted Trees (RTs), which is detailed in Appendix A using an illustrative example.

E. Constraints for Ensuring the Stability and Order

The previous subsections established $\hat{\mathbf{v}}_l^{(j)} \in \mathbb{R}^n$, $j = 0, 1, 2, \ldots, p$ as the set of independent variables required to enforce the constraints specified by (12) and showed how to use those variables to compute both sides of those constraints. However, it should be noted here that the number of those constraints is np, while the number of independent variables is $n(p+1)$, thereby leaving n degrees of freedom to choose those variables. In order to create a unique set of variables that satisfy those constraints and at the same time guarantee the numerical stability throughout the time marching while maintaining the high-order approximation, which is key to the NILT approach, an additional set of n constraints are appended to (12). Those constraints are given by

$$\sum_{j=0}^{p}(-1)^j\alpha_{p,q,j}\hat{\mathbf{v}}_l^{(j)} - \sum_{j=0}^{q}\alpha_{q,p,j}\hat{\mathbf{v}}_{l-1}^{(j)} = \mathbf{0} \tag{28}$$

where $\alpha_{m,n,j} = (m+n-j)!m!/((m+n)!j!(m-j)!)$.

It can be shown that imposing the set of n constraints (28) is equivalent to using a polynomial of degree $p+q$ to interpolate the device voltage $\mathbf{v}_d(t)$, and its derivatives, at $t = (l-1)h$ and $t = lh$ [4], [32]. More precisely, this polynomial matches $\hat{\mathbf{v}}_{l-1}^{(0)}$ and $\hat{\mathbf{v}}_l^{(0)}$ at $t = (l-1)h$ and $t = lh$, respectively. In addition, the polynomial derivatives at $\hat{\mathbf{v}}_{l-1}^{(i)}$ and $\hat{\mathbf{v}}_l^{(j)}$, $i = 1, \ldots, q$ and $j = 1, \ldots, p$, respectively. As will be shown in Section V satisfying those constraints is the sufficient condition guaranteeing the stability and the order of $p+q$, making the proposed approach inherit the desirable properties of NILT.

F. Computing the Jacobian Matrix

The Jacobian matrix of the set of equations in (12) appended by the set of constraints in (28) will assume the following structure

$$\mathbf{J} = \left[\begin{array}{c} \underbrace{\boxed{\mathbf{J}_L + \mathbf{J}_R}}_{np \times n(p+1)} \\ \hline \left(\alpha_{p,q,0}\mathbf{E}_n\right) \quad \left(-\alpha_{p,q,1}\mathbf{E}_n\right) \quad \cdots \quad \left((-1)^p\alpha_{p,q,p}\mathbf{E}_n\right) \end{array}\right] \tag{29}$$

where $\mathbf{J}_L$ and $\mathbf{J}_R$ denote, respectively, the contributions of the left (linear) and right (nonlinear) partition of the split structure. The matrix arising from each partition is derived in the next sections.

1) The Linear Part of the Jacobian Matrix $\mathbf{J}_L$: The Jacobian matrix $\mathbf{J}_L \in \mathbb{R}^{np\times n(p+1)}$, is a block-structured matrix defined by

$$\mathbf{J}_L = \left[\frac{\partial\hat{\mathbf{i}}_l^{(\rho_1)}}{\partial\hat{\mathbf{v}}_l^{(\rho_2)}}\right]_{\substack{\rho_1=0,1,2,\ldots,p-1 \\ \rho_2=0,1,2,\ldots,p}} \tag{30}$$

where $((\partial\hat{\mathbf{i}}_l^{(\rho_1)})/(\partial\hat{\mathbf{v}}_l^{(\rho_2)}))$ is a block in $\mathbb{R}^{n\times n}$. Since this matrix belongs to the linear part of the circuit, it is independent of $\hat{\mathbf{v}}_l^{(m)}$, and therefore remains constant throughout the simulation. $((\partial\hat{\mathbf{i}}_l^{(\rho_1)})/(\partial\hat{\mathbf{v}}_l^{(\rho_2)}))$ can be obtained from (26) by differentiating the right-side with respect to $\hat{\mathbf{v}}_l^{(m\rho_2)}$, resulting in

$$\frac{\partial\hat{\mathbf{i}}_l^{(\rho_1)}}{\partial\hat{\mathbf{v}}_l^{(\rho_2)}} = -\sum_{i=1}^{M/2} 2\Re\left\{z_i^{\rho_1}k_i\tilde{\mathbf{d}}\left(\frac{z_i}{h}\tilde{\mathbf{C}} + \tilde{\mathbf{G}}\right)^{-1}\left[\begin{array}{c}\mathbf{0}\\ \theta_{l,\rho_2}(z_i)\mathbf{E}_n\end{array}\right]\right\}. \tag{31}$$

2) Nonlinear Part of the Jacobian Matrix $\mathbf{J}_R$: $\mathbf{J}_R \in \mathbb{R}^{np\times n(p+1)}$ is given by

$$\mathbf{J}_R = \left[\frac{\partial\hat{\mathbf{j}}_l^{(\rho_1)}}{\partial\hat{\mathbf{v}}_l^{(\rho_2)}}\right] = \left[\frac{1}{\rho_2!}\cdot\frac{\partial\hat{\mathbf{j}}_l^{(\rho_1)}}{\partial\hat{\mathbf{u}}_{l,\rho_2}}\right]_{\substack{\rho_1=0,1,2,\ldots,p-1 \\ \rho_2=0,1,2,\ldots,p}}. \tag{32}$$

Consider an arbitrary block in $\mathbf{J}_R$, such as $((\partial\hat{\mathbf{j}}_l^{(\rho_1)})/(\partial\hat{\mathbf{u}}_{l,\rho_2}))$, where $\rho_1 > 0$. Starting with the definition of $\hat{\mathbf{j}}_l^{(\rho_1)}$ given in (13), this block can be analyzed through the following steps

$$\begin{aligned}\frac{\partial\hat{\mathbf{j}}_l^{(\rho_1)}}{\partial\hat{\mathbf{u}}_{l,\rho_2}} &= h^{\rho_1}\frac{\partial}{\partial\hat{\mathbf{u}}_{l,\rho_2}}\left.\frac{d^{\rho_1}\mathbf{j}_l(\mathbf{u}_l(t))}{dt^{\rho_1}}\right|_{t=lh} \\ &= h^{\rho_1}\frac{\partial}{\partial\hat{\mathbf{u}}_{l,\rho_2}}\left.\frac{d^{\rho_1-1}}{dt^{\rho_1-1}}\left(\frac{d\mathbf{j}_l(\mathbf{u}_l(t))}{dt}\right)\right|_{t=lh}, \quad \rho_1 > 0.\end{aligned} \tag{33}$$

$$\hat{\mathbf{i}}_l^{(m)} = -\sum_{i=1}^{M/2} 2\Re\left\{k_i z_i^m\bar{\mathbf{d}}\left(z_i\bar{\mathbf{C}} + \bar{\mathbf{G}}\right)^{-1}\left[\begin{array}{c}\frac{1}{h}\mathbf{C}\tilde{\mathbf{x}}_{l-1}\\ \sum_{j=0}^{q}\theta_{l-1,j}(z_i)\hat{\mathbf{v}}_{l-1}^{(j)} + \sum_{j=0}^{p}\theta_{l,j}(z_i)\hat{\mathbf{v}}_l^{(j)}\end{array}\right] - k_i\sum_{j=1}^{m-1}z_i^{m-j}\,\hat{\mathbf{i}}_{l-1}^{(j-1)}\right\} \tag{26}$$

Using the chain rule of differentiation in the matrix between the brackets

$$\frac{d\mathbf{j}_l(\mathbf{u}(t))}{dt} = \frac{\partial \mathbf{j}_l(\mathbf{u}_l(t))}{\partial \mathbf{u}_l}\frac{d\mathbf{u}_l(t)}{dt} \tag{34}$$

where $\partial \mathbf{j}_l(\mathbf{u}_l(t))/\partial \mathbf{u}_l$ is a matrix of partial derivatives

$$\frac{\partial \mathbf{j}_l(\mathbf{u}_l(t))}{\partial \mathbf{u}_l} = \left[\frac{\partial j_{n_r}}{\partial u_{n_c}}\right]_{\substack{n_r=1,2,\ldots,n\\ n_c=1,2,\ldots,n}}. \tag{35}$$

This matrix, considered a function of time t, can be expanded in a Taylor series in $\hat{t}$ using

$$\frac{\partial \mathbf{j}_l(\mathbf{u}(t))}{\partial \mathbf{u}} = \sum_{\mu_2=0}^{\infty}\frac{1}{\mu_2!}\hat{\mathbf{R}}_l^{(\mu_2)}\left(\hat{t}-1\right)^{\mu_2} \tag{36}$$

where $\hat{\mathbf{R}}_l^{(\mu_2)}$ is the (matrix-valued) derivative, w.r.t. $\hat{t}$, of $((\partial \mathbf{j}(\mathbf{u}(t)))/\partial \mathbf{u})$ computed at $\hat{t}=1$ or $t=lh$.

Substituting from (36) and (27) into (34) and proceeding in the domain of $\hat{t}$, we get

$$\frac{d\mathbf{j}_l(\mathbf{u}(t))}{d\hat{t}} = \sum_{\mu_1=1}^{p+q+1}\sum_{\mu_2=0}^{\infty}\frac{\mu_1}{\mu_2!}\hat{\mathbf{R}}_l^{(\mu_2)}\hat{\mathbf{u}}_{l,\mu_1}\left(\hat{t}-1\right)^{\mu_1+\mu_2-1}. \tag{37}$$

Differentiating the above expression ρ_1-1 times with respect to $\hat{t}$ substituting $\hat{t}=1$ (equivalent to $t=lh$), leaves only terms where the power $\mu_1+\mu_2-1=\rho_1-1$ or $\mu_2=\rho_1-\mu_1$, with $1\le\mu_1\le\rho_1$. Thus

$$\frac{d^{\rho_1-1}}{d\hat{t}^{\rho_1-1}}\left(\frac{d\mathbf{j}_l(\mathbf{u}(t))}{d\hat{t}}\right) = \sum_{\mu_1=1}^{\rho_1}\frac{\mu_1(\rho_1-1)!}{(\rho_1-\mu_1)!}\hat{\mathbf{R}}_l^{(\rho_1-\mu_1)}\hat{\mathbf{u}}_{l,\mu_1}. \tag{38}$$

Substituting from (38) into (33) retains the term with $\mu_1=\rho_2$ producing

$$\frac{\partial\hat{\mathbf{j}}_l^{(\rho_1)}}{\partial\hat{\mathbf{u}}_{l,\rho_2}} = \begin{cases}\dfrac{\rho_2\,(\rho_1-1)!}{(\rho_1-\rho_2)!}\hat{\mathbf{R}}_l^{(\rho_1-\rho_2)}, & \rho_1\ge\rho_2\\ \mathbf{0}, & \rho_1<\rho_2.\end{cases} \tag{39}$$

A similar procedure can be repeated for the case $\rho_1=0$

$$\frac{\partial\hat{\mathbf{j}}_l^{(0)}}{\partial\hat{\mathbf{u}}_{l,\rho_2}} = \begin{cases}\hat{\mathbf{R}}_l^{(0)}, & \rho_2=0\\ \mathbf{0}, & \rho_2>0.\end{cases} \tag{40}$$

Using (39) and (40) into (32), the desired Jacobian matrix is formulated in the following block-structured matrix $\mathbf{J}_R$ can be expressed as

$$\mathbf{J}_R = \sum_{\rho=0}^{p}\mathbf{T}^{\rho}\otimes\hat{\mathbf{R}}_l^{(\rho)}. \tag{41}$$

Computation of the blocks $\hat{\mathbf{R}}_l^{(\mu_2)}$ is done similar to computing $\hat{\mathbf{v}}_l^{(m)}$, based on the concept of rooted trees. Here, as in the case of $\hat{\mathbf{v}}_l^{(m)}$, a rooted tree is constructed for each nonlinear expression entry in the matrix of partial derivatives $((\partial \mathbf{j}_l(\mathbf{u}_l(t)))/\partial \mathbf{u}_l)$. The operation of rooted tree enables computing $\hat{\mathbf{R}}_l^{(\mu_2+1)}$ from $\hat{\mathbf{R}}_l^{(m)}$ and $\hat{\mathbf{v}}_l^{(m)}$, $m=0,1,\ldots,\mu_2$. The Appendix-A expands further on the idea of utilizing rooted trees in this task.

Algorithm 1 Outline of the Proposed Algorithm

input : h, p, L, $\mathbf{x}_0^{(m)}, m=0,1,\ldots,q$,
1 $\hat{\mathbf{v}}_0^{(m)} \leftarrow \mathbf{d}^{\top}h^m\mathbf{x}_0^{(m)}$;
2 Split the circuit as indicated in Fig. 2 ;
3 Create RTs for the n nonlinear expression in Φ_R;
4 Create RTs for the $n\times n$ entries in the matrix $\frac{\partial\Phi_R}{\partial\mathbf{u}}$;
5 Set $\epsilon\leftarrow 10^{-14}$, $l\leftarrow 1$;
6 **while** $l\le L$ **do**
7 $\quad\boldsymbol{\xi}_l^{[0]} \leftarrow \left[\hat{\mathbf{v}}_{l-1}^{(0)\top}, \hat{\mathbf{v}}_{l-1}^{(1)\top},\ldots,\hat{\mathbf{v}}_{l-1}^{(q)\top},\mathbf{0},\ldots,\mathbf{0}\right]^{\top}$;
8 $\quad$Set $\phi\leftarrow\epsilon+10$ and $k\leftarrow 1$;
9 $\quad$**while** $\phi>\epsilon$ and $k\le\kappa$ **do**
10 $\quad\quad\left[\hat{\mathbf{v}}_l^{(0)\top}, \hat{\mathbf{v}}_l^{(1)\top},\ldots,\hat{\mathbf{v}}_l^{(p)\top}\right]^{\top}\leftarrow\boldsymbol{\xi}_l^{[k]}$;
11 $\quad\quad$Use (26) to compute $\hat{\mathbf{i}}_l^{(m)}$, $m=0,1,\ldots,p-1$;
12 $\quad\quad$Leaf nodes of RTs of $\boldsymbol{\Phi}_R\leftarrow\hat{\mathbf{v}}_l^{(m)}$;
13 $\quad\quad$Compute $\hat{\mathbf{j}}_l^{(m)}$ for $m=0,1,\ldots,p-1$;
14 $\quad\quad\boldsymbol{\delta}\leftarrow\sum_{j=0}^{p}(-1)^j\alpha_{p,q,j}\hat{\mathbf{v}}_l^{(j)}-\sum_{j=0}^{q}\alpha_{q,p,j}\hat{\mathbf{v}}_{l-1}^{(j)}$;
15 $\quad\quad\boldsymbol{\Psi}\leftarrow[(\hat{\mathbf{i}}_l^{(0)}-\hat{\mathbf{j}}_l^{(0)})^{\top},\ldots,\left(\hat{\mathbf{i}}_l^{(p-1)}-\hat{\mathbf{j}}_l^{(p-1)}\right)^{\top}\boldsymbol{\delta}^{\top}]^{\top}$
$\quad\quad\phi\leftarrow||\boldsymbol{\Psi}||$;
16 $\quad\quad$Use (30) and (31) to compute $\mathbf{J}_L$;
17 $\quad\quad$Using RTs of $\frac{\partial\Phi_R}{\partial\mathbf{u}}$ compute $\hat{\mathbf{R}}_l^{(m)}$, then substitute in (41) to compute $\mathbf{J}_R$;
18 $\quad\quad$Use (29) to compute $\mathbf{J}$;
19 $\quad\quad\boldsymbol{\xi}_l^{[k+1]}\leftarrow\boldsymbol{\xi}_l^{[k]}-\mathbf{J}^{-1}\boldsymbol{\Psi}$;
20 $\quad\quad k\leftarrow k+1$;
21 $\quad l\leftarrow l+1$;

G. Summary and Pseudo-Code Representation

The procedure outlined in Algorithm 6 depicts the main computational building blocks of the proposed approach in the form of a pseudo-code. A point worthy of note on this procedure is the input parameters to the algorithm which include $\mathbf{x}_0^{(m)}$ or the mth-order time-domain derivative (up to order $m=q$) of the circuit variables $\mathbf{x}(t)$ at the initial point $t=0$. Computing those derivatives is carried out through the concept of rooted trees given a set of initial condition $\mathbf{x}_0^{(0)}$, which can be obtained using a dc operating point analysis. High-order derivatives, $m>0$, are computed recursively from the lower order ones. This process is described in more detail in [16]. Another point worthy of note is the assignment on line 7, which shows the initialization of the trial vector from the first q derivatives found from the previous $(l-1)$ time step.

It should also be added that the proposed approach can be used to recover the value of any variable within the linear subnetwork. Indeed, upon the convergence of the NR method, then using a suitable selector matrix, other than $\bar{\mathbf{d}}$ in (26) for $m=0$, will recover the time domain of the desired variable at the required point in time.

V. Order and Stability Characterization

The previous developments indicate that although the proposed method is used in general circuit simulation in the

time domain, it can also be considered as a new and general method to solve a system of differential equations numerically. Therefore, to complete its characterization, the classical theory of the numerical methods in differential equations must be used to shed light on its main performance characteristics. Two main characteristics that are key in circuit simulation are considered in this section, namely, the numerical stability and the order of approximating the exact solution of the differential equations. The developments in this section proves that the proposed method carries all the desirable features of NILT to the domain of nonlinear circuits.

The starting point in the characterization of the order and stability is to state an important property of NILT using the following lemma.

Lemma 1: Let NILT be used with parameters integers N and M such that $N + M + 1 \geq m$ for some integer m. Then

$$\mathscr{L}^{-1}\left\{\frac{1}{s^m}\right\}(t)\bigg|_{t=h} = \widehat{\mathscr{L}}^{-1}_{\text{NILT}}\left\{\frac{1}{s^m}\right\}(\hat{t}\,)\bigg|_{\hat{t}=1}. \tag{42}$$

Proof of this lemma can be found in [10, Ch 10].

Studying the order and stability criteria of the proposed method is done using the Dahlquist approach [8] which characterizes a given method for solving differential equations through studying its behavior in approximating the solution to the scalar test problem

$$\frac{dx(t)}{dt} = \lambda x(t) \tag{43}$$

with $x(0) = x_0$, which is equivalent to the circuit on the left of the arrow in Fig. 3 but without the diode or current source. A method is said to be A-stable if the successive approximations to the scalar test problem satisfy $|x_i| < |x_j|$, $(j > i)$ for all values of $\lambda \in \mathbb{C}^-$, L-stable if $|x_i| < |x_j|$, $(j > i)$, for all $\lambda \in \mathbb{C}^- \cup \{\infty\}$, and $A(\alpha)$-stable if $|x_i| < |x_j|$, $(j > i)$ for all $\lambda \in \{\lambda; |\arg(-\lambda)| < \alpha, \lambda \neq 0\}$. Note here that L-stability implies necessarily A-stability. It is crucial that a method used in general circuit simulation to be A-stable, although, it is more preferable to be of the L-stable kind to simulate stiff circuits [33].

The order of a given numerical method is typically defined as the number of the first Taylor series coefficients in the exact solution of (43), $x(t)$, at $t = h$ $(x(h) = \mathrm{e}^{\lambda h}x_0)$ that are matched by the approximation generated from the method, x_1 when expanded as a Taylor series in h.

The proposed method can be applied to the scalar test problem (43) through considering a circuit of a resistor $(-1/\lambda)$ Ω and a capacitor of 1F connected in parallel. In this case $x(t)$ is the voltage across the parallel structure, $v(t)$, and $v(0) = x_0$. The idea of splitting the circuit can be carried out by taking the capacitor as the circuit on the left side, connecting an auxiliary source with one terminal, i.e., $n = 1$, while leaving the resistor to the circuit on the right side of the split with the same auxiliary source. The initial conditions in starting the proposed method, or $\hat{\mathbf{v}}_0^{(m)}$, are scalars and given by $\lambda^m x_0$, where $m = 0, 1, 2, \ldots, q$. Furthermore, the scalar test problem settings entails that the terms used in the proposed approach be given by $\bar{\mathbf{d}}(z_i\bar{\mathbf{C}} + \bar{\mathbf{G}})^{-1} = [1 \ -(z_i/h)]$, $(1/h)\mathbf{C}\tilde{\mathbf{x}}_0 = (x_0/h)$, with scalar auxiliary source $u(t)$ (Laplace-domain $U(s)$) and scalar currents $i(t)$ and $j(t)$, noting that $j(t) = -\lambda u(t)$.

The following theorem states the sufficient condition that makes the proposed method of the A-stable or L-stable type and provides its order.

Theorem 1: Let the proposed splitting method be implemented with parameters q and p, and assume that the NILT technique is used on the linear part with parameters N and M. Then the method is A-stable if $N + M \geq p + q$ and $p - 2 \leq q \leq p$ and L-stable if $N + M \geq p + q$ and $p - 2 \leq q < p$. Furthermore, the order of the method is given by $p + q$.

Proof of the above theorem is given in Appendix B. The next corollary follows from Theorem 1.

Corollary 1: The proposed approach is equivalent to the NILT approach if $q = N$ and $p = M$.

The above corollary shows that if $p = 1$, i.e., enforcing simple equality of currents at the ports, then the proposed approach will have maximum order of 2 regardless of the value used for M.

VI. Handling the General Circuits

Generalization of the proposed approach to handle general nonlinear circuits requires expanding the notations used with the special case of a single nonlinear device. For this purpose, we assume that the circuit has P nonlinear devices, where the ith device has n_i controlling terminal voltages. The set of auxiliary sources placed at the terminals of the devices are denoted by $\mathbf{u}_i(t) \in \mathbb{R}^{n_i}$. The approximations to the mth derivative at the lth time point, which are needed to construct $\mathbf{u}_i(t)$, are denoted by $\hat{\mathbf{v}}_{l,i}^{(m)}$, $m = 0, \ldots, p$ and $l = 1, 2, 3, \ldots$

Meanwhile, $\mathbf{u}(t) \in \mathbb{R}^{\mathfrak{N}}$ is now used to represent the stacking, column-wise, of all vectors of $\mathbf{u}_i(t)$, where $\mathfrak{N}$ is the sum of all number of controlling terminal voltages $\mathfrak{N} = \sum_1^P n_i$. In a similar manner, $\mathbf{i}(t)$ and $\mathbf{j}(t)$ are redefined as vectors in $\mathbb{R}^{\mathfrak{N}}$ that contain all the currents in all ports of the all nonlinear devices from the linear and nonlinear partitions that are supplied/drawn by the auxiliary sources,

To solve system of equations in (12) using Newton-Raphson method, a set of $\mathfrak{N}(p+1)$ independent variables $\boldsymbol{\xi}_l^{[i]}$ are defined that must be updated at each iteration. Finally, the Laplace transform of $\mathbf{u}(t)$ is achieved by using (25) with the difference that here $\mathbf{U}(s)$ groups rth order polynomial coefficients, for the polynomials of all the $\mathfrak{N}$ ports at the lth time step.

The Jacobian matrix will have a similar structure to the structure shown by (29), with the exception that the size of matrices $\mathbf{J}_L$ and $\mathbf{J}_R$ will be $\mathfrak{N} \times \mathfrak{N}(p + 1)$, and $\mathbf{E}_n$ will be replaced by $\mathbf{E}_{\mathfrak{N}}$.

Finally, it is worthy to stress that the splitting framework can be used for general devices, such as current-controlled devices. In that situation the auxiliary sources will be current sources and the independent variables will be the currents in those sources.

VII. Numerical Examples

In this section, three numerical examples are presented to show the accuracy and efficiency of the proposed method. The proposed approach was implemented by parsing and flattening the circuit netlist into a global list of primitive circuit elements.

The topological information of each element (e.g., connection nodes) are embedded in the list elements, which are sequentially polled to create a "structural map." This map identifies the nonlinear elements, their connecting nodes, and their controlling voltages, and is used to form the partitions. The implementation was mostly done in MATLAB (R2021), with the exception of the RTs which was implemented in C++.

A. Example 1: Verification of Theoretical Results

This example demonstrates experimentally that the theoretical results obtained in Section V are valid beyond the scalar test problem used in the proof. The circuit in this example is the circuit shown in Fig. 3, and is executed by exciting it with a sinusoidal current source steady-state, and computing its steady-state response, denoted here $\mathbf{y}_{ss}(t)$. In order to avoid contaminating $\mathbf{y}_{ss}(t)$ with the error that inevitably arises from any numerical time stepping technique, the harmonic balance (HB) technique [34] has been used. HB first expresses $\mathbf{y}_{ss}(t)$ as a Fourier series expansion with unknown coefficients HB then computes these coefficients through solving an algebraic problem, which is free from the approximation errors that erupt in the time stepping.[3] Thus, $\mathbf{y}_{ss}(t)$ computed from HB can be used as a reference in this special circuit against which the results of the proposed approach can be compared. It is important to stress here that HB is still considered a numerical solution, and is only selected for comparison here due to it being free from the truncation error that comes with regular methods used in solving differential equations.

To demonstrate numerically the validity of Theorem 1 [given in (51)], we start the proposed approach from any arbitrary point on $\mathbf{y}_{ss}(t)$, say at $\mathbf{y}_{ss}(0)$, by setting $\mathbf{x}_0^{(0)} = \mathbf{y}_{ss}(0)$ and computing the higher derivatives $\mathbf{x}_0^{(m)}, m = 1, \ldots, q$ at $t = 0$ as explained. The proposed approach, using the splitting indicated on the figure, is then used to advance to time $t = h$, computing $\mathbf{x}_1^{(m)}$. The value of $|\mathbf{x}_1^{(0)} - \mathbf{y}_{ss}(h)|$ may therefore be used to examine the approximation error resulting from single step h and compare its behavior, with regards to h, against the behavior predicted by the above theorem.

Fig. 4 plots, using a log graph, $|\mathbf{x}_1^{(0)} - \mathbf{y}_{ss}(h)|$ versus the length of the time step h, for the two cases $p = 2$ and $q = 0$ of $p = 3, q = 1$. Here, the NILT part of the proposed approach was implemented with $M = 6$ and $N = 4$. According to the result of the theorem in (51), this value (the approximation error $|\mathbf{x}_1^{(0)} - \mathbf{y}_{ss}(h)|$) should asymptotically be proportional to h^{p+q+1} in the limit $h \to 0$. Indeed, as shown in Fig. 4, the error displays a linear slope (due to the log graph) that is in agreement with the predicted error behavior, i.e., slopes of 3 and 5, respectively.

B. Example 2: A BJT Circuit With a Transmission Lines

Fig. 5 shows the schematic of the circuit used in this example. This circuit contains three TL segments consisting of four coupled conductors each, whose physical structure and

[3]Needless to say that this technique is not applicable to the circuits targeted in this article, since the stimulus in those circuits is typically non-sinusoidal and the circuits are typically large to make HB practical.

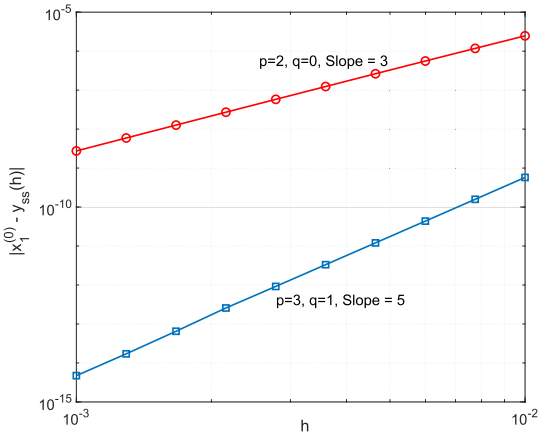

Fig. 4. Asymptotic error behavior.

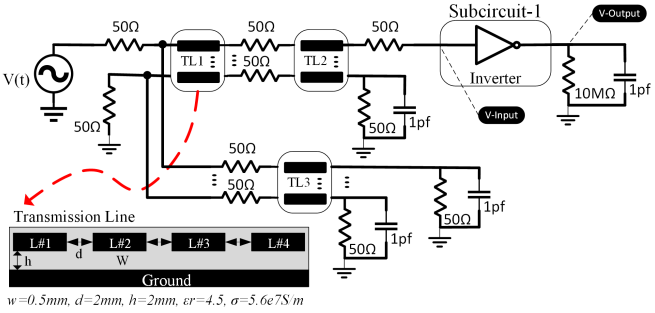

Fig. 5. Three TL circuit diagram.

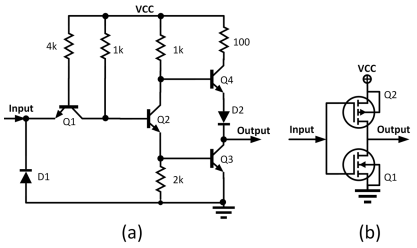

Fig. 6. Inverters used in examples 2, 3 and 4. (a) BJT-based model. (b) MOSFET-based model.

parameters are given on the same figure. The TL model used in this experiment is based on segmentation using 100 sections of lumped RLKC elements [35]. The circuit is excited by a trapezoidal voltage source that is attached to the near end of the first line of the first TL segment.

In this example, the terminating inverter was implemented using BJT inverter shown in Fig. 6(a). the BJT model is the Ebers-Moll Model including all junction capacitance [10]. The size of the MNA formulation is 3652. The circuit is excited using a 1.8 V trapezoidal pulse with 0.1 ns rise/fall times and pulsewidth of 4 ns. The time-domain response at both outputs of TL and inverter are obtained by the proposed method using $M = 4$, $N = 2$, $p = 2$ and $q = 2$ and is displayed in Fig. 7. Fig. 7 compares the waveforms obtained from the proposed approach (circles) against a SPICE-like solver implemented using the TR method (dots).

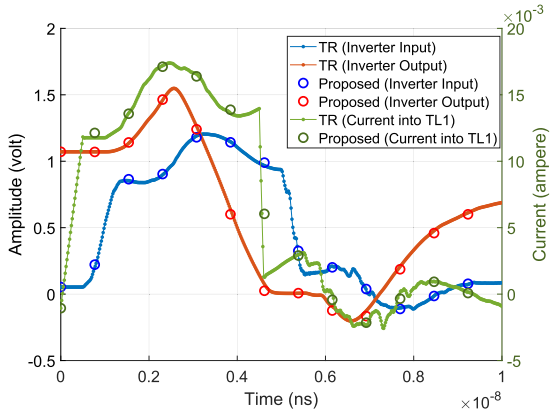

Fig. 7. Waveforms obtained in the BJT inverter circuit.

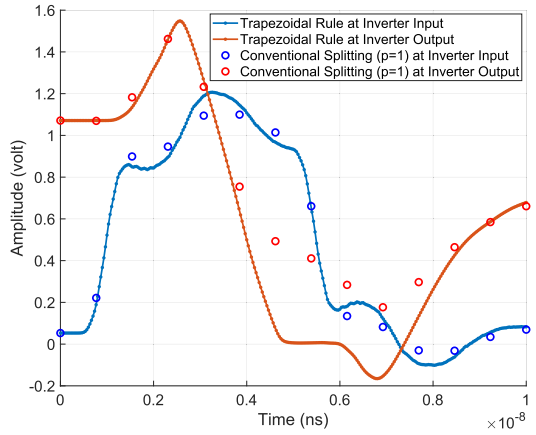

Fig. 8. Waveforms of example 2 using $p = 1$.

Comparison With Existing Splitting Methods: This experiment is used to highlight the fundamental difference between the splitting method proposed in the new approach and known circuit partitioning techniques in existence. For this purpose, the splitting method based on the node tearing method of [36] has been selected for comparison. This method can be shown to be equivalent to setting $p = 1$ in the proposed approach, thereby forgoing the need to match the high-order derivatives of the currents at the ports of the nonlinear devices. Fig. 8 shows the results obtained from that setting for the same step size used in Fig. 7, again comparing it against the TR used with a small time step. Fig. 8 clearly demonstrates the loss of accuracy arising from the conventional splitting based on node tearing method when executed through the same length of time step used in the proposed approach.

C. Example 3: A CMOS Inverter With Transmission Line

This example uses the circuit of Fig. 5, while implementing the inverter as the CMOS of Fig. 6(b). The MOSFET models are taken from [37]. The size of the MNA formulation in this example is 3623. The circuit is excited with a 1.8 V trapezoidal pulse with 0.1 ns rise/fall times and pulsewidth of 5 ns. The time-domain response at both outputs of TL and CMOS inverter are obtained by the proposed approach.

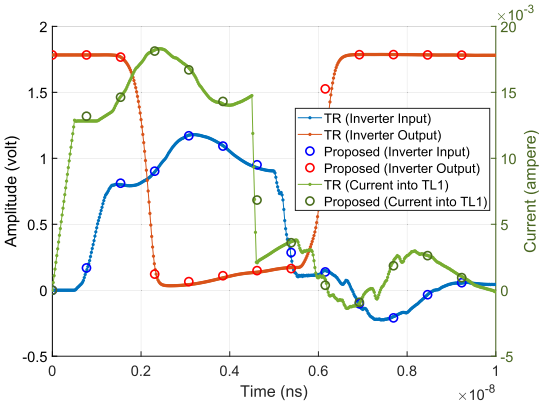

Fig. 9. Waveform obtained in the CMOS inverter circuit.

TABLE I
PERFORMANCE COMPARISON

Ex.	MNA Size	TR		Proposed Approach		Speedup
		#Points	CPU (sec)	#Points	CPU (sec)	
Ex.3	3623	517	6.67	14	0.2	33x
Ex.4	12171	630	93.9	20	0.84	111x

Fig. 9 compares the waveforms obtained from the proposed approach (circles) against a SPICE-like solver implemented using the TR method (dots).

In both of the previous experiments, proposed approach used $M = 4$, $N = 2$, $p = 2$ and $q = 2$. The first row in Table I compares the key performance metrics in the SPICE solver and proposed approach, both based on MATLAB implementation. The table displays that while SPICE required 517 points, the proposed approach needed only 14 points. The reduction in the number of time points is attributable to the larger step size h enabled by the high-order approximation of the proposed approach. Table I also compares the CPU MATLAB computational time between TR and proposed approach for this example indicating a speedup factor of 33 times.

The circuit in Fig. 5 was simulated in the commercial circuit simulator of HSPICE. The runtime statistics reported by the simulator show a total number of NR iterations of 1810 and simulation time of 2.1 s.

D. Example 4: Network of 10 TLs

This experiment used the circuit shown in Fig. 10, which is constructed from ten networks of TLs with a physical structure similar to the one used in the previous example. The inverters in this example were realized using the CMOS family shown in Fig. 6, and the input excitation used was similar to the one utilized for the previous example. The second row in Table I shows the performance comparison between proposed approach ($M = 4, N = 2, p = 3, q = 2$) and the TR method, signifying a speedup of 111 times.

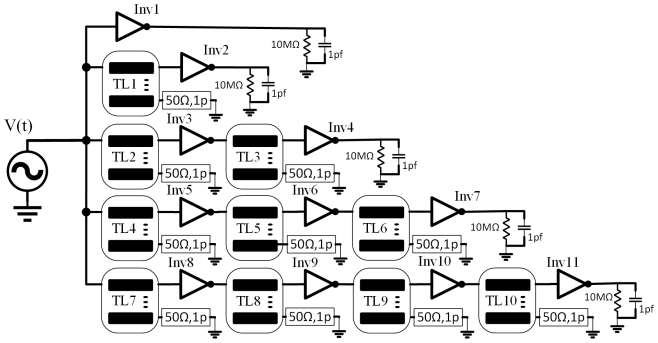

Fig. 10. Circuit diagram for Example 4.

VIII. Conclusion

This article presented a new approach for simulating nonlinear circuits. The proposed approach is developed with the objective of generalizing the notion of the numerical inversion of the Laplace transform (NILT) to the domain nonlinear circuits. The new method enables extending the numerical advantages of the NILT framework which has been limited to linear circuits to the simulation of general nonlinear circuits. Theoretical characterization and numerical validation have been presented showing that the proposed approach does indeed match the performance characteristics of the NILT methodology.

Appendix A
Operation of the Rooted Trees

The purpose of employing the concept of RT in computing the high-order derivatives of the nonlinear part is to make the proposed approach applicable with arbitrarily complex, used-defined form of non-linearity. This appendix illustrates this process in computing $\hat{\mathbf{j}}_l^{(m)}$ though the example shown in Fig. 3 where the current in the diode is assumed to be given by $I_0(\mathrm{e}^{v_1(t)/V_T}-1)$. This process takes the $\hat{\mathbf{v}}_l^{(m)}$ as the input and produces $\hat{\mathbf{j}}_l^{(m)}$ as the output. As mentioned above $\hat{\mathbf{v}}_l^{(m)}$ are provided as part of the κ^{th} trial vector in the course of the NR iterations. For brevity, we shall use "derivatives" to mean derivatives w.r.t t at $t = lh$ and scaled by power of h.

The splitting approach, illustrated on Fig. 3, would only one auxiliary voltage source ($n = 1$), whose current is $j_1(t) = \phi_1(v(t)) \equiv I_0(\mathrm{e}^{v(t)/V_T}-1)$. Therefore, the objective of computing $\hat{\mathbf{j}}_l^{(m)}$, for this example, becomes the objective of computing $h^m((d^m j_1(t))/dt^m)$ or, alternatively, $h^m((d^m\phi_1(v(t)))/dt^m)$. Also $h^m((d^m v_1(t))/dt^m)$ at $t = lh$ play the role of $\hat{\mathbf{v}}_l^{(m)}$. Computing the derivatives of ϕ_1, is done by first representing it in the form of a RT. The RT representing the nonlinear expression of $\phi_1(v(t))$ is shown in Fig. 11.

The nodes in a general RT fall into one of three categories: 1) *Root Node*: This is a unique node that sits at the bottom of the tree (parentless). The tree contains only one such node representing the nonlinear expression, or ϕ_1 in the example taken; 2) Leaf nodes existing at the highest level of the tree (henceforth termed *childless* nodes). Leaf nodes are depicted on the tree using triangles, and represent two types of terms: constant terms in the nonlinearity,

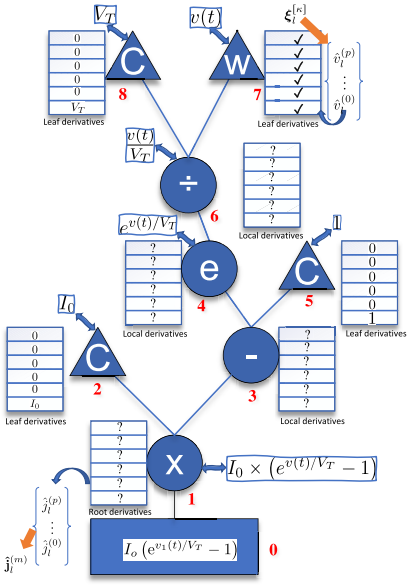

Fig. 11. Representing $I_0(\mathrm{e}^{v(t)/V_T}-1)$ as a RT.

e.g., I_0 (node 2), which are independent of time t, and the controlling voltages, e.g., node 7 for $v(t)$ in the example used. For clarity, the former is assigned letter C and the latter is assigned the letter W on Fig. 11; and 3) *Functional Nodes* Those nodes (designated using circles on the tree) represent the different operations involved in the nonlinear expression. Functional nodes have only one parent node but could have more than one child node and represent either atomic functions, such the exponential or the logarithm function, see node 4, or arithmetic operators such as multiplication and division, such as nodes 1, 3, and 6, in Fig. 11.

Each node in the tree maintains a local storage that will receive p values of the derivatives, at that node level. For example, consider node 4 in Fig. 11 which represents the exponent $\mathrm{e}^{v(t)/V_T}$: its local storage (depicted as an array on the figure) will hold the derivatives of that term $\mathrm{e}^{v(t)/V_T}$. At the start of the process, leaf nodes will have those derivatives known *a priori*. Take for example the nodes representing the constant terms in the tree. The derivatives w.r.t. t of those terms are trivial, by virtue of being constants. For example node 2, representing the constant I_0, have all zero high-order derivatives. This fact is reflected on array attached to node 2 in Fig. 11. Other leaf nodes will have their derivatives, provided in the current trial vector of the NR iteration, that is $\boldsymbol{\xi}_l^{[\kappa]}$. This is the case with node 7 which will have its local storage filled with the derivatives of the voltage $v_1(t)$ that are accessible from the κth trial vector $\boldsymbol{\xi}_l^{[\kappa]}$. More generally, all leaf nodes will have the derivatives (of the terms they represent) readily stored in their local storage at beginning of the computation in a given NR iteration. At this stage, however, derivatives of the terms represented by the other non-leaf nodes are still unknowns (marked by '?') that need to be computed. In particular, the derivatives of the root node, which is the end goal of the entire process, still need to be computed.

The fundamental operating principle in the RT is that derivatives at any non-leaf node can be computed from the

derivatives of the children (higher level) nodes. The formulae that are used for this purpose vary according to the function or the arithmetic operator that the node represents in the RT. Those formulae have been developed for primitive functions, see [38], and preprogrammed in each node.

Thus, the derivatives on the root node 0 are obtained from the derivative of its children, or node 1, which represents the multiplication operator of the constant I_0 and the term $e^{v(t)/V_T} - 1$. The first phase of computation, therefore, starts at the root node, by invoking a command, or a function, that queries the node about its zero-order derivative, or $\hat{j}_l^{(0)}$. The root node, not having computed this derivative yet, passes this command onto the next level. The command thus propagates upward until it reaches the uppermost level of the RT where the leaf nodes exist. The leaf nodes, having their derivatives preassigned, respond by passing the required derivative to the immediate lower level (its parent node) in the tree. Upon receiving the derivatives from all the children nodes, the parent nodes use those derivatives in the preprogrammed formula to compute its own derivative, i.e., the derivative demanded by their lower level ancestral node. The computed derivative is stored in the local storage next to the node for later use, and passed down one level in the tree. This is the second phase of computation, which propagates downward until it reaches the root node providing the required derivative needed by the algorithm, or $\hat{j}_l^{(m)}$. This two-phase process is launched for every derivative needed, $\hat{j}_l^{(m)}$, $m = 0, 1, \ldots, p$.

Construction of the RTs has been fully automated in the implementation of the proposed approach, requiring only the description of the nonlinear devices in the circuit netlist. This description is provided as MATLAB-like script that specifies the nonlinear behavior through a sequence of assignment and conditional statements, thereby allowing representation of fairly complex device behavior. A compiler, designed using ANTLR tool [39], compiles the MATLAB script, converting it into a rooted tree structure. The RT is built in an object-oriented programming (OOP) using the C++ language, which represent nodes as objects of classes with specified inheritance architecture. The OOP allows encapsulating in each node the required preprogrammed formula, as a virtual function resolved at run-time according to the type of expression it represents in the tree, along with the data that are used to in storing the derivatives. The above description thus sums up the process of computing $\hat{\mathbf{j}}_l^{(m)}$ given $\hat{\mathbf{v}}_l^{(m)}$.

APPENDIX B
PROOF OF THEOREM 1

Proof of Theorem 1: From the scalar test problem settings, applying (25) and (20) in (19) while setting $l = 1$, and noting that $\sum_{i=1}^{M/2} k_i = 0$, yields

$$i_1^{(0)} = \frac{1}{h} \sum_{i=2}^{M/2} z_i k_i U_1(z_i). \tag{44}$$

Seeing that the right-side of (44) is equal to $-\widehat{\mathscr{L}}_{\text{NILT}}^{-1}\{\hat{s}U_l(\hat{s})\}(\hat{t})$ at $\hat{t} = 1$ ($t = lh, l = 1$) and noting that $\hat{s}U_1(\hat{s})$ is a summation of terms proportional to $(1/\hat{s}^m)$ with $m \le p + q + 1$, then it follows from the assumption of the theorem ($N + M + 1 \ge p + q + 1$) and the result of Lemma 1 that $-\widehat{\mathscr{L}}_{\text{NILT}}^{-1}\{\hat{s}U_1(\hat{s})\}(1)$ must be equal to $-\mathscr{L}^{-1}\{sU_1(s)\}(t)$ at $t = h$. Since, by the differentiation property of the Laplace transformation, we have $\mathscr{L}^{-1}\{sU_1(s)\}(t) = ((du_1(t))/dt)$ for $t > 0$, then

$$i_1^{(0)} = \frac{-1}{h} \left.\frac{du_1(t)}{dt}\right|_{t=lh}. \tag{45}$$

Since $u_1(t)$ is created such that it has the same (but hitherto unknown) first derivative as the auxiliary source at $t = h$, or $v_1^{(0)}$, then

$$i_1^{(0)} = -\frac{v_1^{(1)}}{h}. \tag{46}$$

In the same vein, the above manipulation can be repeated by substituting into (26) for $m > 0$, leading to

$$i_1^{(m)} = -\frac{v_1^{(m+1)}}{h}, \quad m = 1, 2, \ldots, p-1. \tag{47}$$

The fact that $j(t) = -\lambda u(t)$ necessitates $j_1^{(m)} = -\lambda((d^m u(t))/dt^m)|_{t=lh}$, $m = 0, 1, \ldots, p$. By the properties built in $u(t)$, namely that its derivatives interpolates $v_1^{(m)}$, we have $((d^m u(t))/dt^m)|_{t=lh} = v_1^{(m)}$, and hence

$$j_1^{(m)} = \lambda v_1^{(m)}, \quad m = 0, 1, 2, \ldots, p-1. \tag{48}$$

Substituting from (46)–(48) into (12) and then appending the constraints of (28) ($n = 1, l = 1$) leads to the matrix form $\mathbf{B}\bar{\hat{\mathbf{v}}}_{1,1} = \mathbf{e}_{p+1}\Psi_0^\top \bar{\hat{\mathbf{v}}}_{1,0}$ where $\mathbf{e}_i$ denotes i^{th} column in an identity matrix of size $p + 1$, and

$$\bar{\hat{\mathbf{v}}}_{1,0} = \begin{bmatrix} x_0 & \lambda x_0 & \cdots & \lambda^q x_0 \end{bmatrix}^\top$$

$$\bar{\hat{\mathbf{v}}}_{1,1} = \begin{bmatrix} v_1^{(0)} & v_1^{(1)} & \cdots & v_1^{(p)} \end{bmatrix}^\top$$

$$\Psi_0 = \begin{bmatrix} \alpha_{q,p,0} & \alpha_{q,p,1} & \cdots & \alpha_{q,p,q} \end{bmatrix}^\top$$

$$\mathbf{B} = \begin{bmatrix} -\lambda & 1/h & 0 & \cdots & 0 & \\ 0 & -\lambda & 1/h & 0 & \cdots & 0 \\ & & \ddots & \ddots & & \\ \alpha_{p,q,0} & -\alpha_{p,q,1} & \cdots & & & (-1)^p \alpha_{p,q,p} \end{bmatrix}.$$

For the matrix $\mathbf{B}$, its determinant $\det(\mathbf{B})$ is

$$\det(\mathbf{B}) = (-1)^p \sum_{i=0}^{p} \alpha_{p,q,i} (-\lambda)^i h^{(i-p)} \tag{49}$$

whereas the last column of its adjugate is given by

$$(-1)^p \begin{bmatrix} h^{-p} & \lambda h^{-p+1} & \cdots & \lambda^{p-1} h^{-1} & \lambda^p \end{bmatrix}^\top.$$

This makes $v_1^{(0)}$ computed by the proposed method equal to

$$v_1^{(0)} = \mathbf{e}_1^\top \mathbf{B}^{-1} \mathbf{e}_{p+1} \Psi_0^\top \bar{\hat{\mathbf{v}}}_0 = \underbrace{\left(\frac{\sum_{i=0}^{q} \alpha_{q,p,i} (\lambda h)^i}{\sum_{i=0}^{p} (-1)^i \alpha_{p,q,i} (\lambda h)^i} \right)}_{R_{q,p}(\lambda h)} x_0.$$

The above fraction between the brackets can be recognized as the (q, p) rational Padé approximant to the exponential function [17], [18], $R_{q,p}(\lambda h)$. Therefore, the approximations obtained for the voltage using the proposed method on the scalar test problem after marching in time n steps will be

$$v_n^{(0)} = \left(R_{q,p}(\lambda h)\right)^n x_0. \tag{50}$$

An important property of the Padé approximants is that, for all $z \in \mathbb{C}^-$, $|R_{q,p}(z)| < 1$ if, and only if, $p-2 \leq q \leq p$, and also that, for all $z \in \mathbb{C}^- \cup \{\infty\}$, $|R_{q,p}(z)| < 1$ if, and only if, $p-2 \leq q < p$, [40], [41]. This property thus ensures that $v_i^{(0)} < v_j^{(0)}$, $(i > j)$ for all $z \in \mathbb{C}^-$ and hence the A-stability under the conditions that $p-2 \leq q \leq p$ is proven. The L-stability is guaranteed for $p-2 \leq q < p$, proving that the proposed method is both L-stable and A-stable.

The order of the method can be deduced from the following identity of $R_{q,p}(z)$ [41], namely,

$$\mathrm{e}^z - R_{q,p}(z) = \frac{(-1)^q q!\, p!\, z^{q+p+1}}{(q+p)!\,(q+p+1)!} + \mathcal{O}\left(z^{q+p+2}\right) \qquad (51)$$

which shows clearly that the approximation obtained by the proposed method after taking time step h (i.e., $v_1^{(0)}$) has a Taylor series expansion that agrees with the Taylor series of the exact waveform $\mathrm{e}^{\lambda h} x_0$ in the first $q+p$ terms. This fact establishes that the proposed method is indeed of order $q+p$, and therefore completes the proof of the theorem. □

REFERENCES

[1] A. E. Ruehli, "Equivalent circuit models for three-dimensional multiconductor systems," *IEEE Trans. Microw. Theory Techn.*, vol. MTT-22, no. 3, pp. 216–221, Mar. 1974.

[2] A. E. Ruehli, G. Antonini, and L. Jiang, *Circuit Oriented Electromagnetic Modeling Using the PEEC Techniques*. Hoboken, NJ, USA: Wiley, 2017.

[3] E. Hairer, S. P. Nøsett, and G. Wanner, *Solving Ordinary Differential Equations I Nonstiff Problems*, 3rd ed. Berlin, Germany: Springer, 2008.

[4] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. Hoboken, NJ, USA: Wiley, 2003.

[5] F. N. Najm, *Circuit Simulation*. Hoboken, NJ, USA: Wiley, 2010.

[6] C. Gear, "Simultaneous numerical solution of differential-algebraic equations," *IEEE Trans. Circuit Theory*, vol. CT-18, no. 1, pp. 89–95, Jan. 1971.

[7] R. K. Brayton, F. G. Gustavson, and G. D. Hachtel, "A new efficient algorithm for solving differential-algebraic systems using implicit backward differentiation formulas," *Proc. IEEE*, vol. 60, no. 1, pp. 98–108, Jan. 1972.

[8] G. G. Dahlquist, "A special stability problem for linear multistep methods," *BIT Numer. Math.*, vol. 3, no. 1, pp. 27–43, Mar. 1963.

[9] K. Kundert, *The Designer's Guide to SPICE and Spectre*. Norwell, MA, USA: Kluwer, 1995.

[10] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York, NY, USA: Van Nostrand Reinhold, 1983.

[11] Y. Tao, B. Nouri, M. S. Nakhla, M. A. Farhan, and R. Achar, "Variability analysis via parameterized model order reduction and numerical inversion of Laplace transform," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 7, no. 5, pp. 678–686, May 2017.

[12] L. Lombardi, Y. Tao, B. Nouri, F. Ferranti, G. Antonini, and M. S. Nakhla, "Parameterized model order reduction of delayed PEEC circuits," *IEEE Trans. Electromagn. Compat.*, vol. 62, no. 3, pp. 859–869, Jun. 2020.

[13] H. Heeb and A. E. Ruehli, "Three-dimensional interconnect analysis using partial element equivalent circuits," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 39, no. 11, pp. 974–982, Nov. 1992.

[14] A. E. Ruehli, L. Lombardi, G. Antonini, Y. Tao, M. S. Nakhla, and F. Ferranti, "Impulse response for full wave PEEC models avoiding late time instability," in *Proc. IEEE 28th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2019, pp. 1–3.

[15] B. Bandali, E. Gad, and M. Nakhla, "Fast and stable transient simulation of nonlinear circuits using the numerical inversion of the Laplace transform," in *Proc. IEEE 25th Workshop Signal Power Integrity (SPI)*, May 2021, pp. 1–4.

[16] E. Gad, M. Nakhla, R. Achar, and Y. Zhou, "A-stable and L-stable high-order integration methods for solving stiff differential equations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 9, pp. 1359–1372, Sep. 2009.

[17] G. A. Baker and P. Graves-Morris, *Padé Approximants* (Encyclopedia of Mathematics), G. C. Rota, Ed. New York, NY, USA: Cambridge Univ. Press, 1995.

[18] G. A. Baker, *Essentials of Padé Approximants*. New York, NY, USA: Academic, 1975.

[19] Y. Tao, E. Gad, and M. Nakhla, "Fast and stable time-domain simulation based on modified numerical inversion of the Laplace transform," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 11, no. 5, pp. 848–858, May 2021.

[20] A. Sangiovanni-Vincentelli, L.-K. Chen, and L. Chua, "An efficient heuristic cluster algorithm for tearing large-scale networks," *IEEE Trans. Circuits Syst.*, vol. CT-24, no. 12, pp. 709–717, Dec. 1977.

[21] R. A. Rohrer, "Circuit partitioning simplified," *IEEE Trans. Circuits Syst.*, vol. 35, no. 1, pp. 2–5, Jan. 1988.

[22] G. Kron, *Tensor Analysis of Networks*, (General Electric Series). Hoboken, NJ, USA: Wiley, 1939.

[23] T.-W. Huang, B. Houshmand, and T. Itoh, "The implementation of time-domain diakoptics in the FDTD method," *IEEE Trans. Microw. Theory Techn.*, vol. 42, no. 11, pp. 2149–2155, 1994.

[24] A. R. Newton and A. L. Sangiovanni-Vincentelli, "Relaxation-based electrical simulation," *IEEE Trans. Electron Devices*, vol. ED-30, no. 9, pp. 1184–1207, Sep. 1983.

[25] E. Lelarasmee, A. E. Ruehli, and A. L. Sangiovanni-Vincentelli, "The waveform relaxation method for time-domain analysis of large scale integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. CAD-1, no. 3, pp. 131–145, Jul. 1982.

[26] N. Fröhlich, B. M. Riess, U. A. Wever, and Q. Zheng, "A new approach for parallel simulation of VLSI circuits on a transistor level," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 45, no. 6, pp. 601–613, Jun. 1998.

[27] P. F. Cox, R. G. Burch, D. E. Hocevar, P. Yang, and B. D. Epler, "Direct circuit simulation algorithms for parallel processing (VLSI)," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 10, no. 6, pp. 714–725, Jun. 1991.

[28] D. Paul, R. Achar, M. S. Nakhla, and N. M. Nakhla, "Addressing partitioning issues in parallel circuit simulation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 12, pp. 2713–2723, Dec. 2014.

[29] I. Hajj, "Sparsity considerations in network solution by tearing," *IEEE Trans. Circuits Syst.*, vol. CS-27, no. 5, pp. 357–366, May 1980.

[30] R. A. Saleh, K. A. Gallivan, M. Chang, I. N. Hajj, D. Smart, and T. N. Trick, "Parallel circuit simulation on supercomputers," *Proc. IEEE*, vol. 77, no. 12, pp. 1915–1931, Dec. 1989.

[31] F. J. Hickernell and S. Yang, "Explicit Hermite interpolation polynomials via the cycle index with applications," *Int. J. Numer. Anal. Model.*, vol. 5, no. 3, pp. 457–465, 2008.

[32] B. L. Ehle, "High order a-stable methods for the numerical solution of systems of D.E.'s," *BIT*, vol. 8, no. 4, pp. 276–278, Dec. 1968.

[33] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*. Upper Saddle River, NJ, USA: Prentice-Hall, 1971.

[34] K. S. Kundert, J. K. White, and A. Sangiovanni-Vincentelli, *Steady-State Methods for Simulating Analog and Microwave Circuits*. Boston, MA, USA: Kluwer, 1990.

[35] C. Paul, *Analysis of Multiconductor Transmission Lines*. New York, NY, USA: Wiley, 1994.

[36] R. Rohrer, "Circuit partitioning simplified," *IEEE Trans. Circuits Syst.*, vol. CS-35, no. 1, pp. 2–5, Jan. 1988.

[37] C. C. McAndrew, B. K. Bhattacharyya, and O. Wing, "A single-piece C^∞-continuous MOSFET model including subthreshold conduction," *IEEE Electron Device Lett.*, vol. 12, no. 10, pp. 565–567, Dec. 1991.

[38] E. Gad, R. Khazaka, R. S. Nakhla, and R. Griffith, "A circuit reduction technique for finding the steady-state solution of nonlinear circuits," *IEEE Trans. Microw. Theory Techn.*, vol. 48, no. 12, pp. 2389–2396, Dec. 2000.

[39] T. Parr, *The Definitive ANTLR 4 Reference*, 2nd ed. Raleigh, NC, USA: Pragmatic Bookshelf, 2013.

[40] A. Iserles and S. P. Nørsett, *Order Stars*. Cambridge, U.K.: Univ. of Cambridge, 1991.

[41] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, vol. 2. New York, NY, USA: Springer-Verlag, 1996.

Bardia Bandali (Graduate Student Member, IEEE) received the B.Sc. degree in electronic engineering from the K. N. Toosi University of Technology, Tehran, Iran, in 1991, and the M.Sc. degree from the University of Ottawa, Ottawa, ON, Canada, in 2013, where he is currently pursuing the Ph.D. degree with the School of Electrical Engineering and Computer Science.

He has been involved with several projects that include algorithm design, implementation, simulation, and optimization using MATLAB, C, C++, and OpenCL. His current research interests include nonlinear circuit simulation, development of multiprocessor systems, with activities including project planning, hardware design, programming, and new product development.

Emad Gad (Senior Member, IEEE) received the Ph.D. degree from Carleton University, Ottawa, ON, Canada, in 2003.

He is currently an Associate Professor with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON. His current research interests include numerical simulation and modeling approaches of high speed and RF circuits.

Dr. Gad was the co-recipient of the 2002 IEEE Microwave Prize for a significant contribution in a field of endeavor of the IEEE Microwave and Theory Techniques Society and also the recipient of several honorary awards, including the Governor General Gold Medal, the Carleton University Medal for the Outstanding Academic Performance at the graduate level, and the Ottawa Center for Research and Innovation 2003 as a Student Researcher of the year. He is a Practicing Professional Engineer in the province of Ontario, Canada.

Michel Nakhla (Life Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1975.

From 1976 to 1988, he was with Bell-Northern Research, Ottawa, ON, as the Senior Manager of the Computer-Aided Engineering Group. In 1988, he joined Carleton University, Ottawa, as a Professor and holder of the Senior Computer-Aided Engineering Industrial Chair established by Bell-Northern Research and the Natural Sciences and Engineering Research Council of Canada. He is the Founder of the High-Speed CAD Research Group with Carleton University, Ottawa, where he is currently a Chancellor's Professor of electrical engineering. He has authored more than 370 peer-reviewed research articles. His current research interests include modeling and simulation of high-speed circuits and interconnects, uncertainty quantification, model-order reduction, parallel processing, nonlinear and microwave circuits.

Dr. Nakhla is fellow of the Canadian Academy of Engineering. He is currently an Associate Editor of the IEEE TRANSACTIONS ON COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY and served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. He is on various international committees, including the Standing Committee of the IEEE Signal and Power Workshop. He also served on several Canadian and international government-sponsored research grants selection panels.