Michael Striebel and Joost Rommes[†]

[†]NXP Semiconductors, Eindhoven, the Netherlands

Model order reduction of parameterized nonlinear systems by interpolating input-output behavior

Model order reduction of parameterized nonlinear systems by interpolating input-output behavior

Michael Striebel and Joost Rommes

Abstract—With the increasing amount of layout parasitics and devices in VLSI circuits, model order reduction methods are needed to speed up or enable analog simulation. While reduction methods for linear systems are finding their way to analog simulators, methods for large-scale nonlinear systems are still under development. In this paper we propose a new approach for model order reduction which is applicable for multi-terminal and parameterized nonlinear systems. Instead of projecting onto the dominant state space, an analog macromodel is constructed for the dominant input-output behavior. This macromodel is suitable for (re)use in analog circuit simulators. The performance of the approach is illustrated for benchmark and industrial nonlinear systems.

Index Terms—model order reduction, large-scale systems, nonlinear dynamical systems, table modeling, interpolation, parameterized system, analog macromodeling

I. INTRODUCTION

SIMULATION of VLSI chips is becoming more and more CPU and memory intensive due to the increasing amount of layout parasitics and devices in analog designs. Even if a single simulation is possible, typical design tasks such as circuit optimization can become very expensive or impossible due to the high simulation times and memory requirements. In some cases simulation is not possible at all. Hence, there is need for methods that speed up and/or enable analog simulation of such large designs. Depending on the goal of the simulation, the method must preserve a certain level of accuracy and electrical properties such as stability and passivity.

A popular method for speeding up and/or enabling simulation of large-scale dynamical systems is model order reduction, see, e.g., [1], [2], [3]. For linear systems, typically large RCLk networks arising from parasitic extraction, several methods [4], [5], [6], [7] have been developed in the past decades that are now used in industrial circuit simulators. Although reduction and synthesis of linear systems with many ports is still a challenge [7], the fact that these methods are used in practice indicates that they have reached an acceptable level of robustness.

For nonlinear systems, however, the situation is different. Although there are several methods for reduction of nonlinear dynamical systems (see [3], [8] for an overview), there are two types of methods for reduction of nonlinear dynamical systems

that are being applied to systems arising in circuit simulation: Proper Orthogonal Decomposition (POD) based methods [9], [10], [11] and piecewise-linearization (PWL) methods [12], [13], [14]. Both approaches are characterized by the fact that they try to obtain reduction by projection on the dominant dynamics. They differ in the way the dominant dynamics are obtained and in how the projection is done: the POD based approaches determine the dominant dynamics via a singular value decomposition [15] of selected state space snapshots and project the nonlinear system directly onto a basis for the dominant dynamics. The trajectory piecewise-linear (TPWL) approaches, on the other hand, first approximate the nonlinear system by linearizing around selected linearization points, and reduce the linearized systems by usual model order reduction methods for linear systems. Both approaches have been applied successfully to benchmark and industrial circuits, see for instance [9], [10], [11], [12], [13], [14].

However, as has been studied in [8], both approaches may suffer from difficulties that may limit their practical use. The POD based approaches have as drawback that the reduced systems may be even more expensive to simulate than the original systems. Although there are solutions for this [10], [11], it is not clear how this can be implemented in a robust way. The PWL based approaches, on the other hand, heavily depend on the selection of linearization points, and, during resimulation, on the proper selection of weights for the different linearized models. Some improvements for choosing the linearization points and weights are described in [8], but it is still a challenge to make this robust.

In this paper we present a new method for the reduction of large nonlinear systems. The most significant difference with respect to existing methods is that instead of focusing on the dominant state dynamics, the proposed method tries to capture the dominant input-output behavior. This is motivated as follows: when (re)using a reduced order model as part of a larger system, one is primarily interested in that the response of the reduced model to certain inputs approximates the response of the original system. Hence, similar to reduction methods for linear systems, the proposed method especially takes into account the input-to-state and state-to-output mappings. Another novelty is that the resulting analog macromodel [16], [17] behaves like a circuit element in the sense that given input voltages and/or currents at its terminals, it returns the current, voltage, charge and/or flux contributions and corresponding derivatives. As a result, it can easily be reused in existing circuit simulators. Furthermore, the proposed method is less sensitive on the selection of training inputs and can be extended to multiterminal and parameterized systems, as will be described.

Our approach is conceptually different from table modeling approaches for device modeling [18] since we table model entire subcircuits consisting of many devices.

This paper is organized as follows. In Section II we describe the basics of circuit modeling and time integration. The proposed input-output behavioral modeling approach is described in Section III. In Section IV, we show numerical results for developed algorithms. Section V concludes.

II. CIRCUIT MODELING AND SIMULATION

In general, nonlinear circuits are modeled using modified nodal analysis (MNA), yielding network equations

$$0 = \mathbf{A}_C \frac{d}{dt} \mathbf{q}_C(\mathbf{A}_C^T \mathbf{e}) + \mathbf{A}_R \mathbf{r}(\mathbf{A}_R^T \mathbf{e}) +$$
$$+ \mathbf{A}_L \jmath_L + \mathbf{A}_V \jmath_V + \mathbf{A}_I \imath(t) \quad (1a)$$

$$0 = \frac{d}{dt} \mathbf{\Phi}_L(\jmath_L) - \mathbf{A}_L^T \mathbf{e} \quad (1b)$$

$$0 = \mathbf{v}(t) - \mathbf{A}_V^T \mathbf{e}, \quad (1c)$$

where $\mathbf{e} \in \mathbb{R}^{n_e}, \jmath_L \in \mathbb{R}^{n_L}, \jmath_V \in \mathbb{R}^{n_V}$ denote the unknown node voltages and inductor and voltage source branch currents, respectively. The currents through the n_R resistors are described by a function $\mathbf{r} : \mathbb{R}^{n_R} \rightarrow \mathbb{R}^{n_R}$, taking the n_R resistor-branch voltages $\mathbf{u}_R = \mathbf{A}_R^T \mathbf{e}$ as argument. Accordingly, $\mathbf{q}_C : \mathbb{R}^{n_C} \rightarrow \mathbb{R}^{n_C}$ and $\mathbf{\Phi}_L : \mathbb{R}^{n_L} \rightarrow \mathbb{R}^{n_L}$ derive the charges and fluxes of capacitors and inductors from the capacitor- and inductor-branch voltages $\mathbf{u}_C = \mathbf{A}_C^T \mathbf{e}$ and branch currents $\jmath_L$, respectively. The element functions $\mathbf{r}, \mathbf{q}_C, \mathbf{\Phi}_L$ are nonlinear, in general, in the presence of, e.g., transistors. The incidence matrices $\mathbf{A}_C, \mathbf{A}_R, \mathbf{A}_L, \mathbf{A}_I, \mathbf{A}_V$ with $\mathbf{A}_\Omega \in \{0, -1, +1\}^{n_e \times n_\Omega}$ for $\Omega = C, R, L, I, V$, respectively, describe the topology of the circuit. Voltage and current sources $\mathbf{v}$ and $\imath$ might be controlled by branch currents and branch voltages. In charge oriented MNA, additional unknowns for the charges $\mathbf{q} = \mathbf{q}_C(\cdot)$ and fluxes $\varphi = \mathbf{\Phi}_L(\cdot)$ are introduced. For a more detailed discussion on MNA and the properties of the differential algebraic system (1) we refer to [19].

In time-domain simulation, i.e., transient analysis, the dynamical system (1) is solved for $\mathbf{e}(t), \jmath_L(t), \jmath_V(t)$, when the system is driven by voltage and/or current sources $\mathbf{v}(t)$ and $\imath(t)$, respectively. As the system (1) usually can not be solved analytically, numerical integration is carried out. Both onestep and multistep methods discretize problem (1), leading at some point to linear systems that have to be solved. In multistep methods and implicit Runge-Kutta methods [20], the linear systems arise when a Newton-Raphson technique is applied for solving the occurring nonlinear equations. In linear implicit methods, like Rosenbrock-Wanner schemes [20], the linear systems arise directly in the method's definition.

Whatever scheme one may apply, setting up the corresponding linear equations involves evaluating the element functions $\mathbf{r}, \mathbf{q}_C, \mathbf{\Phi}_L$ and the Jacobians $\mathbf{G}(\mathbf{A}_R^T \mathbf{e}) = \frac{d\mathbf{r}}{d\mathbf{u}_R}|_{\mathbf{A}_R^T \mathbf{e}}$, $\mathbf{C}(\mathbf{A}_C^T \mathbf{e}) = \frac{d\mathbf{q}_C}{d\mathbf{u}_C}|_{\mathbf{A}_C^T \mathbf{e}}$ and $\mathbf{L}(\jmath_L) = \frac{d\mathbf{\Phi}_L}{d\jmath_L}|_{\jmath_L}$. Applying, for instance, the Euler backward method [20] to (1) leads to a system of nonlinear equations that define the approximation $\mathbf{x}_n = (\mathbf{e}_n^T, \jmath_{L,n}^T, \jmath_{V,n}^T)^T$ to the unknowns at a time point $t_n = t_{n-1} + h$. Applying a Newton-Raphson technique, from

some initial guess $\mathbf{x}^{(0)}$, updates $\Delta \mathbf{x}^{(i)}$ are computed until the series $(\mathbf{x}_n^{(0)}, \mathbf{x}_n^{(1)}, \dots)$ with $\mathbf{x}_n^{(i+1)} = \mathbf{x}_n^{(i)} - \Delta \mathbf{x}^{(i)}$ converges. These updates arise from a linear system of the form

$$\mathbf{J} \Delta \mathbf{x}^{(i)} = \mathbf{b} \quad (2)$$

with a system matrix

$$\mathbf{J} = \begin{pmatrix} \frac{1}{h}[\mathbf{A}_C \mathbf{C}(\mathbf{A}_C^T \mathbf{e}_n^{(i)}) \mathbf{A}_C^T] + \mathbf{A}_R \mathbf{G}(\mathbf{A}_R^T \mathbf{e}_n^{(i)}) \mathbf{A}_R^T & \mathbf{A}_L & \mathbf{A}_V \\ -\mathbf{A}_L^T & \frac{1}{h}\mathbf{L}(\jmath_{L,n}^{(i)}) & 0 \\ -\mathbf{A}_V^T & 0 & 0 \end{pmatrix}$$

and right hand side

$$\mathbf{b} = \frac{1}{h} \begin{pmatrix} \mathbf{A}_C \mathbf{q}(\mathbf{A}_C^T \mathbf{e}_n^{(i)}) - \mathbf{A}_C \mathbf{q}(\mathbf{A}_C^T \mathbf{e}_{n-1}) \\ \mathbf{\Phi}_L(\jmath_{L,n}^{(i)}) - \mathbf{\Phi}_L(\jmath_{L,n-1}) \\ 0 \end{pmatrix} +$$
$$+ \begin{pmatrix} \mathbf{A}_R \mathbf{r}(\mathbf{A}_r^T \mathbf{e}_n^{(i)}) + \mathbf{A}_L \jmath_{L,n}^{(i)} + \mathbf{A}_V \jmath_{V,n}^{(i)} + \mathbf{A}_I \imath(t_n) \\ -\mathbf{A}_L^T \mathbf{e}_n^{(i)} \\ -\mathbf{A}_V^T \mathbf{e}_n^{(i)} + \mathbf{v}(t_n) \end{pmatrix}.$$

In summary, solving (1) requires a numerical integration scheme, a nonlinear system solver, and a linear system solver, that all can deal with large-scale systems, as is also clearly described in [21].

For linear systems the Jacobians stay constant and have to be evaluated only once. The function evaluation then boils down to sparse matrix-vector multiplication. Dealing with nonlinear systems, however, the nonlinear functions and the Jacobians have to be evaluated repeatedly, which may be very costly for large systems and/or complex devices [8], [9]. This consideration marks one of the fundamental differences between model order reduction (MOR) methods for linear and nonlinear problems. Linear MOR typically aims at reducing the dimension of the linear equations that have to be solved, trying to keep the final reduced system sparse such that highly elaborated sparse solvers can be applied [5], [7]. MOR methods for nonlinear systems, on the other hand, (have to) focus on reducing the costs for evaluating and solving the system (2). As will be described next, the proposed input-output behavioral modeling method reduces both the costs for evaluation and solution of (2): evaluation is replaced by table-look-up and interpolation, and the (non)linear system sizes are reduced.

III. INPUT-OUTPUT BEHAVIOR MACROMODELING

In MOR for linear problems one considers systems that map input to output usually via some internal states, whose evolution is described by a dynamical system of high dimension. The idea is to replace the internal system by one of lower dimension, retaining the input-output mapping. Reduced macromodels can then be used in system level simulation [5], [7], where subsystems are put together to form a global system. In other words, the focus is on preserving the input-output behavior and the interaction with other parts of the global system, and less on the internal states of the system that is being reduced.

We propose to carry forward this viewpoint, i.e., instantiating subsystems on higher level circuits, to nonlinear systems in a direct way. More precisely, in order to reduce the costs of evaluating the large subsystems, we aim at modeling the

contribution of those subsystems to the system matrix and the right hand sides of the linear equations (2). The main reasoning behind this idea is that in order to (re)use a macromodel in a circuit simulator, in fact all we need are the contributions to (2) at the terminals. So unlike POD and TPWL based methods that tend to focus more on state approximation, the proposed method models input-output behavior. The method differs from other macromodeling methods [16] in the way the behavior is modeled.

We will start with explaining the method for networks with nonlinear resistive elements only. Such networks arise in several industrial applications, for instance during the analysis of electrostatic discharge (ESD) [22], [7], but also in the modeling of transistors. In practice, ESD protection networks are usually modeled using (linear and nonlinear) resistors and diodes.

In the proposed method, subnetworks connected to the next level in the system's hierarchy by a small number of terminals, are replaced by macromodel elements (reduced order models) with similar input-output behavior. This is realized by interpolating precomputed input-output behavior, as will be described in the following.

In a next step, we introduce parameterization, and carry the idea forward to structures made up from capacitive or inductive elements only. Finally, we give an outlook.

A. Resistive structures

In MNA, resistors are modeled as elements that map branch voltages, given by $\mathbf{A}_R^T\mathbf{e}$, to branch currents $\mathbf{r}(\mathbf{A}_R^T\mathbf{e})$. In the process of solving the network equations (1), the element function $\mathbf{r}$ as well as its derivative $\mathbf{G}$ have to be evaluated for different branch voltages $\mathbf{u}_R = \mathbf{A}_R^T\mathbf{e}$ as seen exemplary in (2) for the case of the Euler backward method.

Figure 1 depicts what has to be achieved in nonlinear MOR. A collection of elements, here a diode (modeled as nonlinear resistor) between nodes 2 and 3 and a linear resistor connected to node 3 and ground are to be replaced by one element that contributes to the system in a similar way, i.e., returning the same current at the terminal node 2, when given the same terminal voltage. This can be achieved by a lookup table approach. For purely resistive structures a corresponding table is indexed by terminal voltages and contains terminal currents as well as their derivatives. Consequently, a table is created by computing the input-output relation on a grid of input voltages. When instantiated, the contributions of the macromodel to (2) can be obtained by interpolating the tabulated data.
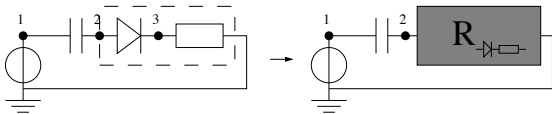

Fig. 1. The diode and resistor in the dashed box on the left are replaced by a macromodel that models the contributions at node 2 only.

1) Extracting a table model: Using the MNA formulation, the problem of computing the current-reply of a purely resistive network, whose n_{pin} terminals are addressed by an

incidence matrix $\mathbf{A}_{\text{pin}} \in \{\pm 1, 0\}^{n_e \times n_{\text{pin}}}$, can be formulated in the following way. Given voltages $\mathbf{v}_{\text{pin}} \in \mathbb{R}^{n_{\text{pin}}}$ at the terminals, solve the system

$$0 = \mathbf{A}_R\mathbf{r}(\mathbf{A}_R^T\mathbf{e}) + \mathbf{A}_{\text{pin}}\boldsymbol{\jmath}_{\text{pin}} \tag{3a}$$

$$0 = \mathbf{v}_{\text{pin}} - \mathbf{A}_{\text{pin}}^T\mathbf{e} \tag{3b}$$

for the terminal currents $\boldsymbol{\jmath}_{\text{pin}}$ and the node voltages $\mathbf{e}$. Clearly, system (3) defines the unknowns $\boldsymbol{\jmath}_{\text{pin}}$ and $\mathbf{e}$ as functions of the terminal voltage $\mathbf{v}_{\text{pin}}$. A differentiation of the system (3) with respect to $\mathbf{v}_{\text{pin}}$ and evaluation at $\tilde{\mathbf{v}}_{\text{pin}}$ yields:

$$0 = \mathbf{A}_R\mathbf{G}(\mathbf{A}_R^T\mathbf{e})\mathbf{A}_R\frac{d\mathbf{e}}{d\mathbf{v}_{\text{pin}}} + \mathbf{A}_{\text{pin}}\frac{d\boldsymbol{\jmath}_{\text{pin}}}{d\mathbf{v}_{\text{pin}}} \tag{4a}$$

$$0 = \mathbf{I}_{n_{\text{pin}}} - \mathbf{A}_{\text{pin}}^T\frac{d\mathbf{e}}{d\mathbf{v}_{\text{pin}}} \tag{4b}$$

where $\mathbf{I}_{n_{\text{pin}}}$ is the $n_{\text{pin}} \times n_{\text{pin}}$ identity matrix and $\mathbf{e} = \mathbf{e}(\tilde{\mathbf{v}}_{\text{pin}})$ and $\boldsymbol{\jmath}_{\text{pin}} = \boldsymbol{\jmath}_{\text{pin}}(\tilde{\mathbf{v}}_{\text{pin}})$ solve (3) for $\mathbf{v}_{\text{pin}} = \tilde{\mathbf{v}}_{\text{pin}}$.

As the contribution of the group of resistive elements to the next higher level in a system's hierarchy is described by the current reply $\boldsymbol{\jmath}_{\text{pin}}(\mathbf{v}_{\text{pin}})$ at the terminals and its derivative $\frac{d}{d\mathbf{v}_{\text{pin}}}\boldsymbol{\jmath}_{\text{pin}}(\mathbf{v}_{\text{pin}})$, a table is constructed in the following way:

1) Choose a discrete set of $k \in \mathbb{N}$ terminal voltages $\mathbf{v}_{\text{pin},1}, \dots, \mathbf{v}_{\text{pin},k}$ with $\mathbf{v}_{\text{pin},i} \in \mathbb{R}^{n_{\text{pin}}}$
2) For each $i \in \{1, \dots, k\}$ compute $\boldsymbol{\jmath}_{\text{pin},i}$ and $\mathbf{e}_i$ by solving the nonlinear system (3) for $\mathbf{v}_{\text{pin}} = \mathbf{v}_{\text{pin},i}$
3) For each $i \in \{1, \dots, k\}$ solve the linear equation (4) for $\mathbf{J}_{\text{pin},i} = \frac{d\boldsymbol{\jmath}_{\text{pin}}}{d\mathbf{v}_{\text{pin}}}$. This amounts to computing

$$\mathbf{J}_{\text{pin},i} = \frac{d\boldsymbol{\jmath}_{\text{pin}}}{d\mathbf{v}_{\text{pin}}} = -\left(\mathbf{A}_{\text{pin}}^T\left(\mathbf{A}_R\mathbf{G}(\mathbf{A}_R^T\mathbf{e}_i)\mathbf{A}_R^T\right)^{-1}\mathbf{A}_{\text{pin}}\right)^{-1}. \tag{5}$$

4) Store the triples

$$\{\mathbf{v}_{\text{pin},i}, \boldsymbol{\jmath}_{\text{pin},i}, \mathbf{J}_{\text{pin},i}\} \quad \text{for } i = 1, \dots, k \tag{6}$$

where $\mathbf{v}_{\text{pin},i} \in \mathbb{R}^{n_{\text{pin}}}$, $\boldsymbol{\jmath}_{\text{pin},i} \in \mathbb{R}^{n_{\text{pin}}}$, $\mathbf{J}_{\text{pin},i} \in \mathbb{R}^{n_{\text{pin}} \times n_{\text{pin}}}$
Note that in step 3) of the algorithm, the matrix $\mathbf{G}(\mathbf{e}_i)^{n_e \times n_e}$ is available from step 2) as for the solution of the nonlinear equation (4), the Jacobian of $\mathbf{r}(\cdot)$ during the Newton-Raphson iterations. Hence, step 4) amounts to compute Schur complements.

Remark: An alternative way of computing $\frac{d\boldsymbol{\jmath}_{\text{pin}}}{dv_{\text{pin}}}$ arises from labeling (maybe after reordering) the node voltages $\mathbf{e} = (\mathbf{e}_p^T, \mathbf{e}_i^T)^T$ as external and internal node voltages $\mathbf{e}_p \in \mathbb{R}^{n_{\text{pin}}}$ and $\mathbf{e}_i \in \mathbb{R}^{n_I}$, respectively, with $n_I = n_e - n_{\text{pin}}$. Accordingly the incidence matrix exhibits the structure $\mathbf{A}_R = (\mathbf{A}_{R_p}^T, \mathbf{A}_{R_i}^T)^T$ with $\mathbf{A}_{R_p} \in \mathbb{R}^{n_{\text{pin}} \times n_R}$ and $\mathbf{A}_{R_i} \in \mathbb{R}^{n_i \times n_R}$. As with this ordering $\mathbf{A}_{\text{pin}}$ results to $(\mathbf{I}_{n_{\text{pin}}}, \mathbf{0})^T$, the system (3) can be reformulated as

$$0 = \begin{pmatrix} \mathbf{A}_{R_p} \\ \mathbf{A}_{R_i} \end{pmatrix}\mathbf{r}(\mathbf{A}_{R_p}^T\mathbf{e}_p + \mathbf{A}_{R_i}^T\mathbf{e}_i) - \begin{pmatrix} \boldsymbol{\jmath}_{\text{pin}} \\ 0 \end{pmatrix} \tag{7a}$$

$$0 = \mathbf{e}_p - \mathbf{v}_p \tag{7b}$$

Differentiating (7) with respect to $\mathbf{v}_{\text{pin}}$ as done above we get:

$$0 = \begin{pmatrix} \mathbf{A}_{R_p} \\ \mathbf{A}_{R_i} \end{pmatrix}\tilde{\mathbf{G}}\left(\mathbf{A}_{R_p}^T \ \mathbf{A}_{R_i}^T\right)\frac{d}{d\mathbf{v}_{\text{pin}}}\begin{pmatrix} \mathbf{e}_p \\ \mathbf{e}_i \end{pmatrix} - \frac{d}{d\mathbf{v}_{\text{pin}}}\begin{pmatrix} \boldsymbol{\jmath}_{\text{pin}} \\ \mathbf{0} \end{pmatrix} \tag{8a}$$

$$0 = \frac{d\mathbf{e}_p}{d\mathbf{v}_{\text{pin}}} - \mathbf{I}_{n_{\text{pin}}} \tag{8b}$$

with $\tilde{\mathbf{G}} = \tilde{\mathbf{G}}(\mathbf{e}_p, \mathbf{e}_i) = \frac{d}{d\mathbf{x}}\mathbf{r}(\mathbf{x})|_{(\mathbf{A}_{R_p}^T \mathbf{e}_p + \mathbf{A}_{R_i}^T \mathbf{e}_i)}$. Introducing the abbreviation

$$\begin{pmatrix} \mathbf{G}_{pp} & \mathbf{G}_{pi} \\ \mathbf{G}_{ip} & \mathbf{G}_{ii} \end{pmatrix} := \begin{pmatrix} \mathbf{A}_{R_p}\tilde{\mathbf{G}}\mathbf{A}_{R_p}^T & \mathbf{A}_{R_p}\tilde{\mathbf{G}}\mathbf{A}_{R_i}^T \\ \mathbf{A}_{R_i}\tilde{\mathbf{G}}\mathbf{A}_{R_p}^T & \mathbf{A}_{R_i}\tilde{\mathbf{G}}\mathbf{A}_{R_i}^T \end{pmatrix},$$

we can also write

$$0 = \begin{pmatrix} \mathbf{G}_{pp} & \mathbf{G}_{pi} \\ \mathbf{G}_{ip} & \mathbf{G}_{ii} \end{pmatrix} \begin{pmatrix} \frac{d\mathbf{e}_p}{d\mathbf{v}_{\text{pin}}} \\ \frac{d\mathbf{e}_i}{d\mathbf{v}_{\text{pin}}} \end{pmatrix} - \begin{pmatrix} \frac{d\jmath_{\text{pin}}}{d\mathbf{v}_{\text{pin}}} \\ 0 \end{pmatrix} \quad (9a)$$

$$0 = \frac{d\mathbf{e}_p}{d\mathbf{v}_{\text{pin}}} - I_{n_{\text{pin}}}. \quad (9b)$$

Hence, the computation of $\mathbf{J}_{\text{pin},i}$ in (5) can be replaced by

$$\mathbf{J}_{\text{pin},i} = \frac{d\jmath_{\text{pin}}}{d\mathbf{v}_{\text{pin}}} = \mathbf{G}_{pp} - \mathbf{G}_{pi}\mathbf{G}_{ii}^{-1}\mathbf{G}_{ip}, \quad (10)$$

which arises from solving the linear equation (9). Arranging nodes into nodes-to-keep and nodes-to-eliminate, resulting in a reduction step similar to (10) is exactly the idea recently used in the reduction of large linear resistive network [7].

Comparing (5) and (10) we note that in the latter a smaller system has to be solved. Furthermore (10) arises from eliminating the voltages $\mathbf{e}_i$ of the inner node by computing the Schur complement [15].

2) Instantiating a table model: Once a table model substituting a resistive structure with $(n_{\text{pin}} + 1)$ interface nodes has been constructed, its instantiation in a network with n_e nodes can be described using an incidence matrix $\mathbf{A}_{TR} \in \{0, \pm 1\}^{n_e \times n_{\text{pin}}}$ together with an element function $\boldsymbol{\tau}_R : \mathbb{R}^{n_{\text{pin}}} \to \mathbb{R}^{n_{\text{pin}}}$. In this way, the current balance part (1a) of the MNA network description changes to

$$0 = \mathbf{A}_C \frac{d}{dt}\mathbf{q}_C(\mathbf{A}_C^T\mathbf{e}) + \mathbf{A}_R\mathbf{r}(\mathbf{A}_R^T\mathbf{e}) +$$
$$+ \mathbf{A}_L\jmath_L + \mathbf{A}_V\jmath_V + \mathbf{A}_I\imath(t) + \boxed{\mathbf{A}_{TR}\boldsymbol{\tau}_R(\mathbf{A}_{TR}^T\mathbf{e})} \quad (11)$$

Consequently, by the appearance of a model of a current defining, voltage driven element, the linear equation (2) that marks the core of each integration scheme changes to

$$\mathbf{J}_{TR}\Delta\mathbf{x}^{(i)} = \mathbf{b}_{TR} \quad (12a)$$

with

$$\mathbf{J}_{TR} = \mathbf{J} + \begin{pmatrix} \mathbf{A}_{TR}\mathbf{T}_R(\mathbf{A}_{TR}^T\mathbf{e}_n^{(i)})\mathbf{A}_{TR}^T & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad (12b)$$

$$\mathbf{b}_{TR} = \mathbf{b} + \begin{pmatrix} \mathbf{A}_{TR}\boldsymbol{\tau}_R(\mathbf{A}_{TR}^T\mathbf{e}_n^{(i)}) \\ 0 \\ 0 \end{pmatrix} \quad (12c)$$

with the Jacobian $\mathbf{T}_R(\mathbf{A}_{TR}^T\mathbf{e}) = \frac{\partial\boldsymbol{\tau}_R}{\partial x}|_{\mathbf{A}_{TR}^T\mathbf{e}}$.

TABLE I
MACROMODEL FOR RESISTIVE STRUCTURE USING TABULATED DATA:
MAPPING INPUT $\mathbf{u}$ TO CURRENT RESPONSE $\boldsymbol{\tau}_R$ AND DERIVATIVE $\mathbf{T}_R$.

$\mathbf{u}$	$\mathbf{v}_{\text{pin},1}$	$\cdots$	$\mathbf{v}_{\text{pin},k}$
$\boldsymbol{\tau}_R$	$\jmath_{\text{pin},1}$	$\cdots$	$\jmath_{\text{pin},1}$
$\mathbf{T}_R$	$\mathbf{J}_{\text{pin},1}$	$\cdots$	$\mathbf{J}_{\text{pin},k}$

The model is realized by a lookup table (Tab. I) and the evaluation of $\boldsymbol{\tau}_R$ and $\mathbf{T}_R$ is done by interpolating from the tabulated data $\jmath_{P,i}$ and $\mathbf{J}_{\text{pin},i}$, respectively, in (6). We emphasize that the Jacobian $\mathbf{T}_R(\cdot)$ is not calculated via numerical differentiation based on the data $\boldsymbol{\tau}_R(\cdot)$ but directly from tabulated values. In this way the macromodel also mimics the differential behavior the detailed, full model would show.

B. Parameterized structures

In a straightforward way also parameterization of element functions can be incorporated in the tabeling approach.

A resistive network, with elements depending on m_ρ parameters $\rho_1, \ldots, \rho_{m_\rho}$ with $\rho_j \in \mathbb{R}$ for $j = 1, \ldots, m_\rho$ is described by

$$0 = \mathbf{A}_R\mathbf{r}(\mathbf{A}_R^T\mathbf{e}; \boldsymbol{\rho}) + \mathbf{A}_{\text{pin}}\jmath_{\text{pin}} \quad (13a)$$
$$0 = \mathbf{v}_{\text{pin}} - \mathbf{A}_{\text{pin}}^T\mathbf{e} \quad (13b)$$

with $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_{m_\rho})^T \in \mathbb{R}^{m_\rho}$.

To create the table, besides sweeping over a range of pin voltages $\mathbf{v}_{\text{pin}} \in \{\mathbf{v}_{\text{pin},1}, \ldots, \mathbf{v}_{\text{pin},k}\}$ we also scan the input-output behavior for different parameters $\boldsymbol{\rho} \in \{\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_l\}$. This leads to an extended macromodel

$$\boldsymbol{\tau} : \mathbb{R}^{n_P} \times \mathbb{R}^{n_{\text{par}}} \to \mathbb{R}^{n_P}, \quad (\mathbf{v}_{\text{pin}}, \boldsymbol{\rho}) \mapsto \mathbf{y} = \boldsymbol{\tau}(\mathbf{v}_{\text{pin}}, \boldsymbol{\rho}), \quad (14)$$

realized by a table with datapoints $[(\mathbf{v}_{\text{pin},\mu}, \boldsymbol{\rho}_\nu), (\boldsymbol{\tau}^{(\mu,\nu)}, \mathbf{T}^{(\mu,\nu)})]$ for $\mu = 1, \ldots, l$ and $\nu = 1, \ldots, k$.

C. Capacitive structures

The element functions $\mathbf{q}_C$ for capacitors are mappings from branch voltages to charges. The corresponding branch currents, needed to set up the current balance equation (1a), arise from the capacitors' charges by differentiation with respect to time.

In applying any numerical integration to the system (1) of differential equations, the value of the function $\mathbf{q}$ and its Jacobian $\mathbf{C}$ at various branch voltages are necessary. This gives raise to the idea of combining structures of connected capacitive elements, linear and nonlinear in nature, to one circuit element, that, given voltages at its terminals returns charges similar to the structure it arises from.

Extracting a table model: Given voltages $\mathbf{v}_{\text{pin}} \in \mathbb{R}^{n_{\text{pin}}}$ at the pins, the distribution of charges and voltages in a network of capacitors only, is described by

$$0 = \mathbf{A}_C\mathbf{q}(\mathbf{A}_C^T\mathbf{e}) - \mathbf{A}_{\text{pin}}\mathbf{q}_{\text{pin}},$$
$$\mathbf{0} = \mathbf{v}_{\text{pin}} - \mathbf{A}_{\text{pin}}^T\mathbf{e}, \quad (15)$$

where $\mathbf{q}_{\text{pin}}$ are point charges at the structure's pins.

Analog to the procedure for resistive structures (Sect. III-A) we tabulate data for purely capacitive structures:

$$\{\mathbf{v}_{\text{pin},i}, \mathbf{q}_{\text{pin},i}, \mathbf{Q}_{\text{pin},i}\} \quad \text{for } i = 1, \ldots, l, \quad (16)$$

where $\mathbf{v}_{\text{pin},i} \in \mathbb{R}^{n_{\text{pin}}}$, $\mathbf{q}_{\text{pin},i} \in \mathbb{R}^{n_{\text{pin}}}$, $\mathbf{Q}_{\text{pin},i} \in \mathbb{R}^{n_{\text{pin}} \times n_{\text{pin}}}$. The $l \in \mathbb{N}$ pin voltages $\mathbf{v}_{\text{pin},.}$ need to be specified. The charges $\mathbf{q}_{\text{pin},1}, \ldots, \mathbf{q}_{\text{pin},l}$ arise from solving the nonlinear equation (15) with the corresponding input. Finally, the Jacobians $\mathbf{Q}_{\text{pin},i}$ for $i = 1, \ldots, l$ are computed as

$$\mathbf{Q}_{\text{pin},i} = \frac{d\mathbf{q}_{\text{pin}}}{d\mathbf{v}_{\text{pin}}}\Big|_{\mathbf{v}_{\text{pin},i}} = -\left(\mathbf{A}_{\text{pin}}^T\left(\mathbf{A}_C\mathbf{C}(\mathbf{A}_C^T\mathbf{e}_i)\mathbf{A}_C^T\right)^{-1}\mathbf{A}_{\text{pin}}\right)^{-1}.$$

(17)

i. e., from solving

$$0 = \mathbf{A}_C\mathbf{C}(\mathbf{A}_C^T\mathbf{e})\mathbf{A}_C\frac{d\mathbf{e}}{d\mathbf{v}_{\text{pin}}} + \mathbf{A}_{\text{pin}}\frac{d\mathbf{q}_{\text{pin}}}{d\mathbf{v}_{\text{pin}}} \tag{18a}$$

$$0 = \mathbf{I}_{n_{\text{pin}}} - \mathbf{A}_{\text{pin}}^T\frac{d\mathbf{e}}{d\mathbf{v}_{\text{pin}}} \tag{18b}$$

where $\mathbf{e} = \mathbf{e}(\mathbf{v}_{\text{pin},i})$ solves (15) for $\mathbf{v}_{\text{pin}} = \mathbf{v}_{\text{pin},i}$.

Parameterized nonlinear capacitors can be dealt with as described in Sect. III-B

D. Inductive structures

Inductors are current driven, magnetic flux defining elements. Hence, the element functions $\mathbf{\Phi}_L$ map currents to magnetic fluxes. Inductors are dynamic elements. A change in the magnetic flux induces a voltage drop. As we see in (1b), the branch voltage of inductor branches are given by the time derivative of the corresponding magnetic flux.

As for the network elements resistor and capacitor, to set up the linear system (2) at the core of each integration scheme, the element function $\mathbf{\Phi}_L$ and the Jacobian $\mathbf{L}$ have to be evaluated at different current values.

Again, a homogeneous structure containing linear and nonlinear inductors can be combined to one metaelement, replying a magentic flux when supplied with an electric current at its terminals.

Extracting a table model: Prescribing currents $\boldsymbol{\imath}_{\text{pin}} \in \mathbb{R}^{n_{\text{pin}}}$ at the pins, the distribution of fluxes in a magnetic network is described by

$$0 = \mathbf{A}_L\boldsymbol{\jmath}_L - \mathbf{A}_{\text{pin}}\boldsymbol{\imath}_{\text{pin}}, \tag{19a}$$
$$0 = \mathbf{\Phi}_L(\boldsymbol{\jmath}_L) - \mathbf{A}_L^T\boldsymbol{\varphi}, \tag{19b}$$

because also in a magentic network laws by Kirchhoff apply, saying, e. g., that the magnetic fluxes $\boldsymbol{\varphi}$ add to zero in each node. The fluxes $\boldsymbol{\varphi}_{\text{pin}}$ at the terminals, i. e., the quantities that are replied to surrounding are selected by

$$0 = -\mathbf{A}_{\text{pin}}^T\boldsymbol{\varphi} + \boldsymbol{\varphi}_{\text{pin}}. \tag{19c}$$

Analog to the procedure for resistive and capacitive structures (Sects. III-A & III-C) we tabulate data for purely inductive structures:

$$\{\boldsymbol{\imath}_{\text{pin},i}, \boldsymbol{\varphi}_{\text{pin},i}, \mathbf{L}_{\text{pin},i}\} \quad \text{for } i = 1, \ldots, l, \tag{20}$$

where $\boldsymbol{\imath}_{\text{pin},i} \in \mathbb{R}^{n_{\text{pin}}}$, $\boldsymbol{\varphi}_{\text{pin},i} \in \mathbb{R}^{n_{\text{pin}}}$, $\mathbf{L}_{\text{pin},i} \in \mathbb{R}^{n_{\text{pin}} \times n_{\text{pin}}}$.

During a process of scanning the input-output behaviour the $l \in \mathbb{N}$ pin currents $\boldsymbol{\imath}_{\text{pin},\cdot}$ are specified. Then the system (19) is solved for $\boldsymbol{\jmath}_L$, $\boldsymbol{\varphi}$ and $\boldsymbol{\varphi}_{\text{pin}}$. The Jacobians $\mathbf{L}_{\text{pin},i}$ for $i = 1, \ldots, l$ are computed as

$$\mathbf{L}_{\text{pin},i} = \frac{d\boldsymbol{\varphi}_{\text{pin}}}{d\boldsymbol{\imath}_{\text{pin}}}\Big|_{\boldsymbol{\imath}_{\text{pin},i}} = \mathbf{A}_{\text{pin}}^T\left(\mathbf{A}_L\mathbf{L}(\boldsymbol{\jmath}_{L,i})^{-1}\mathbf{A}_L^T\right)^{-1}\mathbf{A}_{\text{pin}}, \tag{21}$$

where $\mathbf{L}(\boldsymbol{\jmath}_{L,i})$ are the derivatives of the element function $\mathbf{\Phi}_L$, evaluated at $\boldsymbol{\jmath}_{L,i}$, part of the solution of (19) for $\boldsymbol{\imath}_{\text{pin}} = \boldsymbol{\imath}_{\text{pin},i}$.

E. Instantiating table models for dynamic structures

In the previous sections III-C and III-D we modeled the static behaviour of elements, that cause the dynamics in a network. Analog to the instantiation of the meta model $\boldsymbol{\tau}_R$ in the current balance equations (11), we can include meta models for purely capacitive and purely inductive structures, with models described by $\boldsymbol{\tau}_C$ and $\boldsymbol{\tau}_L$, respectively. Summing up, the MNA equations (1) for a system containing the classical elements resistor R, capacitor C, inductor L and sources, and homogeneous substructures that are described by the meta models we extract are given by

$$0 = \mathbf{A}_C\frac{d}{dt}\mathbf{q}_C(\mathbf{A}_C^T\mathbf{e}) + \mathbf{A}_R\mathbf{r}(\mathbf{A}_R^T\mathbf{e}) + $$
$$+ \mathbf{A}_L\boldsymbol{\jmath}_L + \mathbf{A}_V\boldsymbol{\jmath}_V + \mathbf{A}_I\boldsymbol{\imath}(t) + \tag{22a}$$
$$+ \boxed{\mathbf{A}_{TC}\frac{d}{dt}\boldsymbol{\tau}_C(\mathbf{A}_{TC}^T\mathbf{e})} + \boxed{\mathbf{A}_{TR}\boldsymbol{\tau}_R(\mathbf{A}_{TR}^T\mathbf{e})} + \boxed{\mathbf{A}_{TL}\boldsymbol{\jmath}_{TL}},$$

$$0 = \frac{d}{dt}\mathbf{\Phi}_L(\boldsymbol{\jmath}_L) - \mathbf{A}_L^T\mathbf{e}, \tag{22b}$$

$$0 = \boxed{\frac{d}{dt}\boldsymbol{\tau}_L(\boldsymbol{\jmath}_{TL}) - \mathbf{A}_{TL}^T\mathbf{e}}, \tag{22c}$$

$$0 = \mathbf{v}(t) - \mathbf{A}_V^T\mathbf{e}, \tag{22d}$$

where the boxed terms refer to the contribution of the meta models. Note, that we introduce additional unknowns $\boldsymbol{\jmath}_{TL}$ in (22a) and an additional equation (22c) accounting for the currents caused by meta models describing purely inductive structures.

Repeating the consideration that led to the description of the systems (2) for the network equations (22), it becomes clear that indeed, we only need the evaluations of the meta model functions $\boldsymbol{\tau}_\Omega$ and its derivative $\mathbf{T}_\Omega$ (for $\Omega \in \{R, C, L\}$) during numerical integration of the network equations. For each meta model and for each state during integration evaluations of $\boldsymbol{\tau}_\Omega$ and $\mathbf{T}_\Omega$ are computed by interpolation from the tabulated data that was collected during the training phase. Recall that the contribution to the Jacobian in the time integration $\mathbf{T}_\Omega$ is not calculated from numerical differentiation based on $\boldsymbol{\tau}_\Omega$ but on the tabulated data directly.

IV. NUMERICAL RESULTS

For testing purposes the presented approach for the extraction and the instantiation of table models has been implemented in MATLAB, where, for the time being for interpolation the available functions interp{1,2,3,n} have been used.

For the time integration of the dynamical systems (22), not an Euler scheme, but CHORAL [23], a Rosenbrock-Wanner based scheme of local order 3 with adaptive stepsize selection, tailored to for the integration of differential algebraic network problems was implemented and used.

In the following we present results from different testcases.

A. Realistic ESD system

The presented meta-modeling approach is motivated by a realistic problem from semiconductor industry. We use this problem from electro static discharge (ESD) analysis as a first example to show the usability of the method.

ESD analysis is used to obtain knowledge on how fast electrical charge on the pins of a package can be discharged, and whether the metal interconnect can handle the expected current densities [7]. The interconnect and resistance networks are typically modeled by resistors, and diodes are used to connect different parts of the networks. The resulting networks may contain up to millions of linear and nonlinear elements and hence there is need for reduced order models that enable fast but accurate simulation.

The ESD circuit at hand consists of 35804 nodes, of which 1 is a pin, hence a connection to the outside world. The ESD problem is a purely resistive structure comprising 202766 linear resistors and 947 diodes, modeled by a nonlinear voltage-current relation

$$\jmath_d(u) = \frac{1}{rs} \left(\text{hyp}(u - vzp, \delta) - \text{hyp}(-vzp, \delta) \right)$$
$$- \frac{1}{rdx} \left(\text{hyp}(-x - vzm, \delta) - \text{hyp}(-vzm, \delta) \right)$$

with $\text{hyp}(x, \epsilon) = \frac{1}{2} \left(x + \sqrt{x^2 + 4\epsilon^2} \right)$. This model is subject to parameterization by rs, rdx, vzp, vzm, δ and ϵ, we do not specify here.
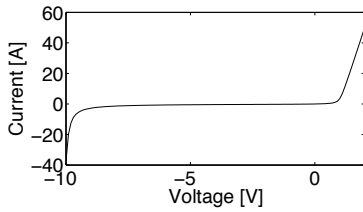

Fig. 2. Voltage-current haracteristic of diode.

However, the voltage-current relation of each single diode follows the characteristics depicted in Fig. 2. Motivated by this characteristic, we choose the training input

$$\mathbf{v}_{\text{pin}} \in \{-10.0 : 0.2 : -9.4, \, 0.0, \, 0.6 : 0.2 : 1.2\},$$

using the `MATLAB`-notation *start:step:end*.

Despite the situation given in (22), during a design process we are not interested to instantiate the ESD circuit in a circuit with dynamic elements. Instead we are interested in scaning the voltage at the ESD circuits' pins, given a current is flowing because voltage peaks could cause damage of the circuit. In other words: we connect the ESD circuit with a current source and measure the voltage at the subblock's pin.

Summing up, during the training phase, the ESD circuit is connected to a voltage source, in the application, it is connected to a current source. Once the table has been constructed for the nine different inputs above, we see from Tab. II that the calculation of the voltage reply for varying the magnitude of the current source at the block's input in the range $-5.0 : 0.1 : 5.0$ with the table model takes much less time than carrying out the calculation with the full problem, where the maximum relative error encountered here amounts to 0.08 % (the speedup is about 300x for the simulation (12x when including the extraction time)).

TABLE II
ESD CIRCUIT: ELAPSED TIME [SEC]

| full system | reduced system |
	extraction time: 1 min 57 sec
24 min 1 sec	4.5 sec

B. Nonlinear transmission line

An extended version of the nonlinear transmission line [12] frequently used in papers about nonlinear MOR [13], [8], serves as first test example. This circuit, as displayed in Fig. 3 basically consists of a series of N blocks with a capacitive path to ground at the end of each block, completed with a current source. Every block contains M pairs made up from a diode and a linear resistor. Therefore, the dimension of the problem amounts to $n = N \cdot M + 1$.
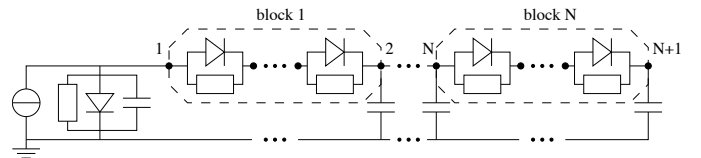

Fig. 3. Adapted transmission line [12].

In the following, we chose $N = 10$ and $M = 100$, creating a system of dimension $n = 1001$. Furthermore, as in [12], the diode-current is modeled as $\imath_d(x) = \exp(40 \cdot x) - 1$, where x is the voltage across its terminals. The linear resistors as well as the capacitors have unit value 1. The current source provides a current $\imath(t) = \frac{1}{2} \left(\cos \left(2\pi f t \right) + 1 \right)$.

As the top-level circuit from Fig. 3 comprises N times the same block, a reduced system of dimension $r = N + 1 = 11$ can be obtained by replacing each block by a table model. For this purely resistive block a table is constructed once as described above. Then each individual block is replaced by this macro model.

For setting up the table, we chose input voltages v_{pin} from the set $\{-0.01, 0.00, +0.01\}$. Note this is an obvious choice, i.e., no heuristics are needed. For the example at hand, a speedup of about factor 15 becomes evident from Tab. III, with a very good matching as can be seen from Fig. 4.

TABLE III
TRANSMISSION LINE: ELAPSED TIME [SEC]

| | full system | reduced system |
		extraction time: 0.59
$f = 0.1$	27.96	1.96
$f = 1.0$	279.99	16.44

C. Nonlinear parameterized transmission line

Introducing a parameter p in the modeling of the diode current, i.e. $\imath_d(x) = \exp(p \cdot x) - 1$ we transform the transmission line from Fig. 3 to a nonlinear parameterized problem. A table model representing the basic block structure is now constructed from scanning the subsystem with both varying input voltages and parameter settings. In the following, we chose the
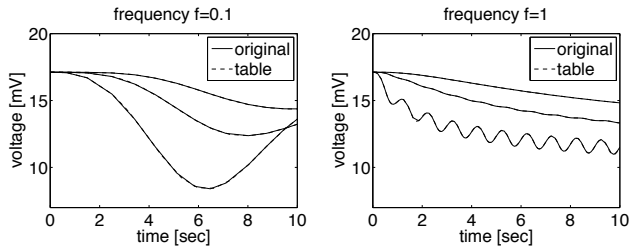
Fig. 4.   Transmission line: voltages at node 2,3,4

Fig. 6.   Varactor transmission line

pin voltage $v_{\text{in}} = \{-0.02, -0.01, 0.00, +0.01, +0.02\}$ and the parameter $p = \{35, 55\}$, leading to a table with 10 grid points.

From Fig. 5 no difference in the trajectories produced with the same parameters for the original and the reduced system can be see. For the example at hand a speedup of about factor 6 becomes obvious from Tab. IV.
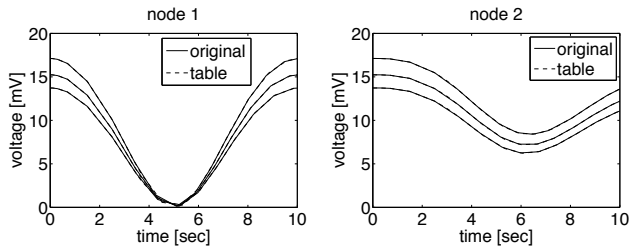

Fig. 5.   Parameterized transmission line: $p = \{40, 45, 50\}$ (top to down)

TABLE IV
PARAMETERIZED TRANSMISSION LINE: ELAPSED TIME [SEC]

	full system	reduced system extraction time: 1.45
$p = 40$	28.04	4.20
$p = 45$	28.49	4.21
$p = 50$	26.73	4.21

D. Circuit with nonlinear capacitors

The circuit shown in Fig. 6 is again a kind of transmission line. The central block, describing the capacitive effects of the line, is made up from M pairs of a linear capacitor and a so-called varactor diode in series, i.e. a nonlinear capacitor used in several practical applications [24], [25]. This block is repeated N times in the overall circuit that comprises also resistors and inductors.

For x being the voltage across the terminals of the varactor diode the charge is modeled according to

$$q_d(x) = \frac{C_0 \cdot U_d}{1-n} \cdot \left(1 - \left(1 - \frac{x}{U_d}\right)^{1-n}\right),$$

with constant parameters $C_0 = 2$, $U_d = 0.45$ and $n = 0.5$

For the case study we chose $M = 100$ and $N = 10$, resulting in a full system of dimension $n = 2021$.

For the capacitive block a compact model is derived by sweeping $v_{\text{pin}} = \{0.0, \pm 0.3, \pm 0.6\}$. For testing, the block is instantiated $N = 10$ times and we choose a voltage source:
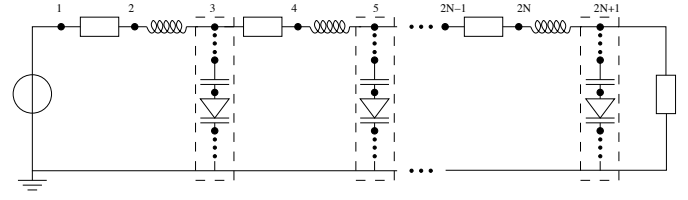
$$v(t) = \sin(2\pi \cdot 10^5 \cdot t) + 0.4 \cdot \sin(2\pi \cdot 10^7 \cdot t).$$

Inspecting Fig. 7 no difference in the trajectories produced with the full system and the reduced one is evident. Furthermore, from Tab. V a speedup of around 100 is recordable.

TABLE V
VARACTOR TRANSMISSION LINE: ELAPSED TIME [SEC]

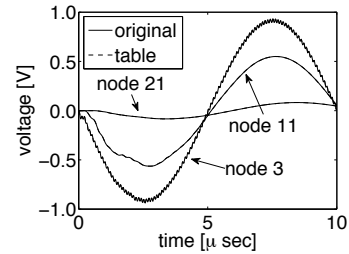| full system	reduced system extraction time: 1.59 sec
3 h 19 min 16 sec	1 min 59 sec

Fig. 7.   Varactor diode circuit: voltages at node 3,11,21

V. CONCLUDING REMARKS

We have presented a method for the reduction of large-scale parameterized nonlinear systems. The method, that is based on interpolation the input-output behavior of nonlinear dynamical systems, was successfully applied to large-scale realistic electrical networks. Numerical results confirm that speed-ups in transient simulation of factor up to 50 can be achieved, without loss of accuracy. Furthermore, the method is easy to implement and robust with respect to choosing the training inputs. Additionally, since the reduced order models are available as table models, they can easily be used by existing circuit simulators.

A possible disadvantage of the presented method is that it will suffer from the curse of dimensionality: as the number of parameters and terminals increases, the costs for model building and evaluation may grow. Note that also other methods like TPWL and POD suffer from this. Adaptive sampling [26] and sparse grid methods may help to limit the costs and are under current investigation. In the experiments shown here also with a rather naive choice of the training inputs, the models were still robust.

Extensions to mixed static/dynamic circuits and advanced interpolation methods that can deal with large numbers of inputs and outputs are subject to future research.

REFERENCES

[1] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.

[2] S. X. D. Tan and L. He, *Advanced model order reduction techniques in VLSI design*. Cambridge University Press, 2007.

[3] W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, Eds., *Model Order Reduction: Theory, Research Aspects and Applications*, ser. Mathematics in Industry. Springer, 2008, vol. 13.

[4] A. Odabasioglu, M. Celik, and T. Pileggi, "PRIMA: Passive Reduced-order Interconnect Macromodeling Algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 8, pp. 645–654, 1998.

[5] K. J. Kerns and A. T. Yang, "Stable and efficient reduction of large, multiport networks by pole analysis via congruence transformations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 7, pp. 734–744, 1997.

[6] ——, "Preservation of passivity during rlc network reduction via split congruence transformations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 7, pp. 582–591, 1998.

[7] J. Rommes and W. H. A. Schilders, "Efficient methods for large resistor networks," *IEEE Trans. CAD Int. Circ. Syst.*, vol. 29, no. 1, pp. 28–39, January 2010.

[8] M. Striebel and J. Rommes, "Model order reduction of nonlinear systems: status, open issues, and applications," Technische Universität Chemnitz, Tech. Rep. CSC/08-07, 2008.

[9] A. Verhoeven, J. ter Maten, M. Striebel, and R. Mattheij, "Model order reduction for nonlinear IC models," TU Eindhoven, CASA-Report 07-41, 2007, to appear in IFIP2007 conference proceedings.

[10] P. Astrid and A. Verhoeven, "Application of least squares mpe technique in the reduced order modeling of electrical circuits," in *Proceedings of the 17th Int. Symp. MTNS*, 2006, pp. 1980–1986.

[11] S. Chaturantabut and D. C. Sorensen, "Nonlinear model reduction via discrete empirical interpolation," *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2737–2764, 2010.

[12] M. J. Rewieński and J. White, "A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices," *IEEE Trans. CAD Int. Circ. Syst.*, vol. 22, no. 2, pp. 155–170, February 2003.

[13] B. N. Bond and L. Daniel, "A piecewise-linear moment-matching approach to parameterized model-order reduction for highly nonlinear systems," *IEEE Trans. CAD Int. Circ. Syst.*, vol. 26, no. 12, pp. 2116–2129, December 2007.

[14] ——, "Stable reduced models for nonlinear descriptor systems through piecewise-linear approximation and projection," *IEEE Trans. CAD Int. Circ. Syst.*, vol. 28, no. 10, pp. 1467–1480, October 2009.

[15] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. John Hopkins University Press, 1996.

[16] T. McConaghy and G. Gielen, "Double-strength caffeine: fast template-free symbolic modeling of analog circuits via implicit canonical form functions and explicit introns," in *Proceedings of Design Automation and Test in Europe (DATE)*, 2006, pp. 269–274.

[17] B. D. Smedt and G. G. E. Gielen, "Watson: design space boundary exploration and model generation for analog and rfic design," *IEEE Trans. Comp.-Aided Design of Integrated Circuits and Systems*, vol. 22, pp. 213–224, 2003.

[18] P. Meijer, "Table models for device modelling." in *Proc. ISCAS.*, vol. 3, 1988, pp. 2593–2596.

[19] M. Günther, U. Feldmann, and E. J. W. ter Maten, "Modelling and discretization of circuit problems," in *[27]*. Elsevier B.V., 2005, pp. 523–650.

[20] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I – Nonstiff Problems*. Springer, 2000.

[21] M. Rewieński, "A perspective on fast-SPICE simulation technology," in *Advanced Simulation and Verification of Electronic and Biological Systems*. Springer, 2005, pp. 23–42.

[22] J. M. Kolyer and D. Watson, *ESD: From A To Z*. Springer, 1996.

[23] M. Günther, "Simulating digital circuits numerically – a charge-oriented ROW approach," *Numer. Math*, vol. 79, pp. 203–212, 1998.

[24] O. O. Yildirim, D. S. Ricketts, and D. Ham, "Reflection soliton oscillator," *IEEE Trans. Microw. Theory. Tech.*, vol. 57, no. 10, pp. 2344–2453, October 2009.

[25] D. W. Porterfield, T. W. Crowe, R. F. Bradley, and N. R. Erickson, "A high-power, fixed-tuned, millimeter-wave balanced frequency doubler," *IEEE Trans. Microw. Theory. Tech.*, vol. 47, no. 4, pp. 419–425, April 1999.

[26] W. Hendrickx, D. Gorissen, and T. Dhaene, "Grid enabled sequential design and adaptive metamodeling," in *Proceedings of the 2006 Winter Simulation Conference*, 2006, pp. 872–881.

[27] P. G. Ciarlet, W. H. A. Schilders, and E. J. W. ter Maten, Eds., *Numerical Methods in Electromagnetics*, ser. Handbook of Numerical Analysis. Elsevier, North Holland, 2005, vol. XIII.

Michael Striebel received the Diploma degree in mathematics form Ulm University, Ulm, Germany in 2002 and the Ph.D. degree in mathematics from Bergische Universiät Wuppertal, Wuppertal, Germany in 2006. He is currently a researcher Bergische Universität Wuppertal, Germany, and works on model order reduction and multirate time integration, related to circuit design and simulation and quantum physics.

Joost Rommes received the M.Sc. degree in computational science (cum laude), the M.Sc. degree in computer science (cum laude), and the Ph.D. degree in mathematics from Utrecht University, Utrecht, The Netherlands, in 2002, 2003, and 2007, respectively. He is currently a researcher at NXP Semiconductors, Eindhoven, The Netherlands, and works on model order reduction, specialized eigenvalue methods, and algorithms for problems related to circuit design and simulation.