Reduced-Order Modeling of Large Linear Passive Multi-Terminal Circuits Using Matrix-Padé Approximation

Roland W. Freund and Peter Feldmann
Bell Laboratories
Lucent Technologies
Murray Hill, New Jersey 07974–0636, USA

Abstract

This paper introduces SyMPVL, an algorithm for the approximation of the symmetric multi-port transfer function of an RLC circuit. The algorithm employs a symmetric block-Lanczos algorithm to reduce the original circuit matrices to a pair of typically much smaller, banded, symmetric matrices. These matrices determine a matrix-Padé approximation of the multi-port transfer function, and can serve as a reduced-order model of the original circuit. They can be "stamped" directly into the Jacobian matrix of a SPICE-type circuit simulator, or can be used to synthesize an equivalent smaller circuit. We also prove stability and passivity of the reduced-order models in the RL, RC, and LC special cases, and report numerical results for SyMPVL applied to example circuits.

1 Introduction

Electronic circuits often contain large linear subnetworks of passive components. Such subnetworks may represent interconnect automatically extracted from layout as large RLC networks, models of IC packages, models of wireless propagation channels, etc.

In general, we are interested only in the behavior of the linear subnetworks at their terminals. This behavior can be fully characterized by a matrix-valued transfer function. In most practical cases, a very closely approximating matrix-valued transfer function can be obtained from a reduced-order model that is several orders of magnitude "smaller" than the original circuit. In [6], we introduced MPVL (**M**atrix **P**adé **V**ia a **L**anczos-type process), an algorithm for the accurate and efficient computation of reduced-order models of large linear circuits. MPVL computes a matrix-Padé approximation of the original circuit's matrix transfer function. MPVL is a general algorithm, applicable to any linear system, and for different number of inputs and outputs.

RLC circuits are described by symmetric matrices and can be characterized in terms of square and symmetric matrix transfer functions. Moreover, being composed entirely of passive components, RLC circuits are always stable and passive. In this paper we introduce SyMPVL, a variant of MPVL, which takes advantage of these special structures of RLC circuits. SyMPVL computes symmetric matrix-Padé approximations of the circuit matrix transfer function, using a symmetric block-Lanczos algorithm. SyMPVL produces increasingly more accurate reduced-order models and, in the important special cases of RL, RC, and LC circuits, guarantees stability and passivity at any order of approximation. The results of the algorithm can be used to synthesize a reduced, equivalent circuit. The work described in this present paper generalizes SyPVL [8], which is an algorithm for computing single-input single-output transfer functions and models.

Alternative approaches, not based on Padé approximation, for generating reduced-order models for RLC circuits also exist. In [16], a block-Arnoldi algorithm is employed. Another approach is PACT, which relies on pole matching and is described in [11].

2 Symmetric Circuit Equations

In this section, we first formulate the equations that describe general RLC circuits, and then we discuss the special cases of RC, RL, and LC circuits. We show that the matrices associated with the circuit equations are always symmetric and, in the special cases, they are also positive semi-definite. These properties are important for the SyMPVL algorithm and for stability and passivity proofs.

2.1 General RLC Circuits

The connectivity of a circuit can be captured by the adjacency matrix, $\mathbf{A}$. Each row of $\mathbf{A}$ corresponds to a circuit element and each column to a circuit node. The column corresponding to the datum (ground) node of the circuit is omitted in order to remove redundancy. By convention, we attach an arbitrary direction to each circuit element, and the corresponding row of $\mathbf{A}$ will contain $+1$ in the column corresponding to the

source node, -1 in the column corresponding to the destination node, and 0 everywhere else. Kirchhoff's laws, which depend only on connectivity, can be expressed as follows:

$$\begin{array}{ll} \text{KCL:} & \mathbf{A}^{\mathrm{T}}\mathbf{i}_b = \mathbf{0}, \\ \text{KVL:} & \mathbf{A}\mathbf{v}_n = \mathbf{v}_b. \end{array} \tag{1}$$

Here $\mathbf{i}_b$ and $\mathbf{v}_b$ are the vectors of branch currents and voltages, respectively, and $\mathbf{v}_n$ is the vector of the non-datum node voltages.

We are interested in analyzing RLC circuits and for simplicity, we assume that the circuit is excited only by current sources. The adjacency matrix and the branch current and voltage vectors can then be partitioned according to circuit-element types:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_i \\ \mathbf{A}_g \\ \mathbf{A}_c \\ \mathbf{A}_l \end{bmatrix}, \quad \mathbf{v}_b = \begin{bmatrix} \mathbf{v}_i \\ \mathbf{v}_g \\ \mathbf{v}_c \\ \mathbf{v}_l \end{bmatrix}, \quad \mathbf{i}_b = \begin{bmatrix} \mathbf{i}_i \\ \mathbf{i}_g \\ \mathbf{i}_c \\ \mathbf{i}_l \end{bmatrix}.$$

Here the subscripts i, g, c, and l stand for branches containing current sources, resistors, capacitors, and inductors, respectively.

The set of circuit equations is completed by adding the *branch constitutive relationships* (BCR's), which describe the physical behavior of the circuit elements. For RLC circuits, the BCR's are:

$$\mathbf{i}_i = -\mathbf{I}_t(t),\ \mathbf{i}_g = \mathcal{G}\mathbf{v}_g,\ \mathbf{i}_c = \mathcal{C}\frac{d}{dt}\mathbf{v}_c,\ \mathbf{v}_l = \mathcal{L}\frac{d}{dt}\mathbf{i}_l. \tag{2}$$

Here $\mathbf{I}_t(t)$ is the vector of current-source values, $\mathcal{G}$ and $\mathcal{C}$ are appropriately-sized diagonal matrices whose diagonal entries are the conductance and capacitance values of each element. Clearly, these values are positive for any physical circuit. The matrix $\mathcal{L}$ has the inductance values on the diagonal and may contain off-diagonal elements that model inductive coupling. Nevertheless, it remains symmetric positive definite.

The modified nodal formulation (MNA) of the circuit equations is obtained by combining the Kirchhoff equations (1) with the BCRs (2), and eliminating as many current unknowns as possible. For the case of RLC circuits only inductor currents need to be left as unknowns. Setting

$$\begin{aligned} \mathbf{G} &= \begin{bmatrix} \mathbf{A}_g^{\mathrm{T}}\mathcal{G}\mathbf{A}_g & \mathbf{A}_l^{\mathrm{T}} \\ \mathbf{A}_l & \mathbf{0} \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} \mathbf{A}_c^{\mathrm{T}}\mathcal{C}\mathbf{A}_c & \mathbf{0} \\ \mathbf{0} & -\mathcal{L} \end{bmatrix}, \\ \mathbf{x} &= \begin{bmatrix} \mathbf{v}_n \\ \mathbf{i}_l \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} \mathbf{A}_i^{\mathrm{T}} \\ \mathbf{0} \end{bmatrix}, \end{aligned} \tag{3}$$

the resulting MNA equations can be summarized compactly in matrix form:

$$\mathbf{G}\mathbf{x} + \mathbf{C}\frac{d}{dt}\mathbf{x} = \mathbf{B}\mathbf{I}_t(t). \tag{4}$$

Note that the matrices $\mathbf{G}$ and $\mathbf{C}$ in (3) are symmetric, and, in general, indefinite. However, in Section 2.2 below, we will discuss some important special cases for which the matrices become positive semi-definite.

We are interested in determining the network functions of the RLC block viewed as a p-terminal component. Since we allowed only current sources in our formulation, it is natural to determine the Z-parameters. By applying the Laplace transform to (4) and assuming zero initial conditions, we obtain

$$\begin{aligned} (\mathbf{G} + s\mathbf{C})\mathbf{X} &= \mathbf{B}\,\mathbf{I}_s(s), \\ \mathbf{V}_i &= \mathbf{B}^{\mathrm{T}}\mathbf{X}. \end{aligned} \tag{5}$$

Here $\mathbf{X}$, $\mathbf{I}_s(s)$, and $\mathbf{V}_i$ represent the Laplace transforms of the unknown vector $\mathbf{x}$, the excitation current $\mathbf{I}_t(t)$, and the vector of voltages across the excitation sources, respectively. Eliminating $\mathbf{X}$ in (5) gives

$$\begin{aligned} \mathbf{V}_i = \begin{bmatrix} \mathbf{A}_i & \mathbf{0} \end{bmatrix}\mathbf{X} &= \mathbf{Z}(s)\mathbf{I}_s(s), \\ \text{where} \quad \mathbf{Z}(s) &= \mathbf{B}^{\mathrm{T}}\left(\mathbf{G} + s\mathbf{C}\right)^{-1}\mathbf{B}. \end{aligned} \tag{6}$$

2.2 Special Cases: RC, RL, LC Circuits

In the case of RC circuits, the matrices $\mathcal{L}$, $\mathbf{A}_l$, and the vector $\mathbf{i}_l$ are empty. The matrices $\mathbf{C}$ and $\mathbf{G}$ in (3) then reduce to

$$\mathbf{G} = \mathbf{A}_g^{\mathrm{T}}\mathcal{G}A_g \quad \text{and} \quad \mathbf{C} = \mathbf{A}_c^{\mathrm{T}}\mathcal{C}\mathbf{A}_c,$$

and thus $\mathbf{C}$ and $\mathbf{G}$ are positive semi-definite.

In the case of RL circuits, the matrices $\mathcal{C}$ and $\mathbf{A}_c$ are empty. The Laplace transform of (4) now becomes

$$\begin{aligned} \mathbf{A}_g^{\mathrm{T}}\mathcal{G}\mathbf{A}_g\mathbf{V}_n + \mathbf{A}_l^{\mathrm{T}}\mathbf{I}_l &= \mathbf{A}_i^{\mathrm{T}}\mathbf{I}_s(s), \\ \mathbf{A}_l\mathbf{V}_n - \mathcal{L}s\mathbf{I}_l &= \mathbf{0}. \end{aligned} \tag{7}$$

By multiplying the first equation in (7) by s and substituting $s\mathbf{I}_l$ from the second, we obtain

$$\underbrace{\mathbf{A}_g^{\mathrm{T}}\mathcal{G}\mathbf{A}_g}_{\mathbf{C}}\mathbf{V}_n s + \underbrace{\mathbf{A}_l^{\mathrm{T}}\mathcal{L}^{-1}\mathbf{A}_l}_{\mathbf{G}}\mathbf{V}_n = s\mathbf{A}_i^{\mathrm{T}}\mathbf{I}_s(s),$$

where the matrices $\mathbf{C}$ and $\mathbf{G}$ are again symmetric positive semi-definite.

In the case of LC circuits, the matrices $\mathcal{G}$ and $\mathbf{A}_g$ are empty. The Laplace transform of (4) is reduced to

$$\begin{aligned} \mathbf{A}_c^{\mathrm{T}}\mathcal{C}\mathbf{A}_c s\mathbf{V}_n + \mathbf{A}_l^{\mathrm{T}}\mathbf{I}_l &= \mathbf{A}_i^{\mathrm{T}}\mathbf{I}_s(s), \\ \mathbf{A}_l\mathbf{V}_n - \mathcal{L}s\mathbf{I}_l &= \mathbf{0}. \end{aligned} \tag{8}$$

By multiplying the first equation in (8) by s and substituting $s\mathbf{I}_L$ from the second, we obtain

$$\underbrace{\mathbf{A}_c^{\mathrm{T}}\mathcal{C}\mathbf{A}_c}_{\mathbf{C}}\mathbf{V}_n s^2 + \underbrace{\mathbf{A}_l^{\mathrm{T}}\mathcal{L}^{-1}\mathbf{A}_l}_{\mathbf{G}} = s\mathbf{A}_i^{\mathrm{T}}\mathbf{I}_s(s). \tag{9}$$

Again, the matrices $\mathbf{C}$ and $\mathbf{G}$ are symmetric positive semi-definite and a change of variables $\sigma = s^2$ brings (9) to the usual form.

3 Matrix-Padé Approximation

Recall that the transfer function $\mathbf{Z}(s)$ of an p-port is defined in (6) where $\mathbf{G}$ and $\mathbf{C}$ are real symmetric $N \times N$ matrices, and $\mathbf{B}$ is a real $N \times p$ matrix. We remark that $\mathbf{Z} : \mathbb{C} \longmapsto (\mathbb{C} \cup \{\infty\})^{p \times p}$ is a matrix-valued rational function. Each pole s of $\mathbf{Z}$ is also an eigenvalue of the matrix pencil $\mathbf{G} + \lambda \mathbf{C}$, i.e., each pole s of $\mathbf{Z}$ satisfies the equation $\det(\mathbf{G} + s\mathbf{C}) = 0$.

3.1 Review of the Case $p = 1$

For the moment, assume that $p = 1$. In this case, $\mathbf{Z}$ is a scalar-valued rational function, and for each n, we can define a scalar-valued Padé approximant $\mathbf{Z}_n$ to $\mathbf{Z}$ as follows. A function of the form

$$\mathbf{Z}_n(s) = \frac{\phi_{n-1}(s)}{\psi_n(s)}, \tag{10}$$

where ϕ_{n-1} and ψ_n are polynomials of degree at most $n-1$ and n, respectively, is called an nth *Padé approximant* to $\mathbf{Z}$ if

$$\mathbf{Z}(s) = \mathbf{Z}_n(s) + \mathcal{O}(s^{q(n)}), \tag{11}$$

where $q(n)$ is as large as possible; see, e.g., [3]. The condition (11) means that the Taylor expansions of $\mathbf{Z}$ and $\mathbf{Z}_n$ about $s = 0$ agree in as many leading Taylor coefficients (the so-called *moments*) as possible. In general, $q(n) = 2n$.

The standard approach to computing $\mathbf{Z}_n$ is based on explicit moment generation. First, one computes the leading $q(n)$ Taylor coefficients of $\mathbf{Z}$, and from these, one then generates the coefficients of the polynomials ϕ_{n-1} and ψ_n in (10). This standard approach to computing $\mathbf{Z}_n$ is employed in the asymptotic waveform evaluation (AWE) technique [13, 14]. However, computing Padé approximants using explicit moment computations is inherently numerically unstable, and indeed, in practice, this approach can be used only for very moderate values of n, such as $n \leq 10$; see [5]. These numerical instabilities can be avoided by exploiting the Lanczos-Padé connection [10] and generating the Padé approximant $\mathbf{Z}_n$ via the Lanczos process [12]. The resulting algorithms for stably computing Padé approximants of transfer function are PVL [4, 5] for general circuits and its special variant SyPVL [8] for RLC circuits. In SyPVL, the Padé approximant $\mathbf{Z}_n$ is computed using the formula

$$\mathbf{Z}_n(s) = \boldsymbol{\rho}_n^{\mathrm{T}} \left(\mathbf{A}_n + s\mathbf{B}_n\right)^{-1} \boldsymbol{\rho}_n, \tag{12}$$

where the $n \times n$ matrices $\mathbf{A}_n$, $\mathbf{B}_n$ and the vector $\boldsymbol{\rho}_n$ are generated from a symmetric version of the Lanczos process.

3.2 The General Case $p \geq 1$

Now we return to the general case of p-ports with $p \geq 1$. One approach to obtaining approximations of $\mathbf{Z}$ is to compute scalar Padé approximants for each of the p^2 entries of $\mathbf{Z}$ by means of p^2 runs of PVL. However, a much more efficient approach is to use the concept of matrix-Padé approximation [3] that generates a matrix-valued approximation $\mathbf{Z}_n$ for all entries of $\mathbf{Z}$ in one run. Moreover, the reduced-order model generated by $\mathbf{Z}_n$ is much smaller than the reduced-order model obtained from p^2 individual PVL runs.

Matrix-Padé approximants can be represented by means of a pair of numerator and denominator matrix polynomials. However, this approach suffers from the same instability mentioned above for the case $p = 1$. Instead, we use an appropriate extension of the formula (12) to the case $p \geq 1$. We say that a matrix-valued function $\mathbf{Z}_n : \mathbb{C} \longmapsto (\mathbb{C} \cup \{\infty\})^{p \times p}$ is an nth *matrix-Padé approximant* to $\mathbf{Z}$ if $\mathbf{Z}_n$ is of the form

$$\mathbf{Z}_n(s) = \boldsymbol{\rho}_n^{\mathrm{T}} \left(\mathbf{A}_n + s\mathbf{B}_n\right)^{-1} \boldsymbol{\rho}_n, \tag{13}$$

where $\boldsymbol{\rho}_n \in \mathbb{C}^{n \times p}$ $\mathbf{A}_n$, $\mathbf{B}_n \in \mathbb{C}^{n \times n}$, and if

$$\mathbf{Z}(s) = \mathbf{Z}_n(s) + \mathcal{O}(s^{q(n)}) \quad \text{with maximal } q(n). \tag{14}$$

In general, we have $q(n) \geq 2\lfloor n/p \rfloor$, with $q(n) > 2\lfloor n/p \rfloor$ if, and only if, so-called *deflation* occurs due to certain linear dependencies.

In the next section, we formulate a Lanczos-type algorithm that generates matrices $\mathbf{A}_n$, $\mathbf{B}_n$, and $\boldsymbol{\rho}_n$ such that the function $\mathbf{Z}_n$ defined in (13) is indeed an nth matrix-Padé approximant to $\mathbf{Z}$.

4 A Symmetric Lanczos-Type Process

In order to compute matrix-Padé approximants to matrix-valued functions $\mathbf{Z}$, we need a Lanczos-type algorithm that can handle multiple starting vectors, namely the p columns of the matrix $\mathbf{B}$ in (6). Such a procedure was recently developed in [1] and further refined in [7]. We use a special symmetric variant of the algorithm described in [7] that exploits the symmetry of the matrices $\mathbf{G}$ and $\mathbf{C}$ in (6). Lanczos-type procedures for multiple starting vectors are necessarily quite involved for two reasons. First, in the course of any such algorithm, linearly dependent vectors may occur that need to be *deflated*. Second, so-called *look-ahead* techniques are required to avoid potential breakdowns due to division by quantities that cannot be excluded to be zero. The algorithm used here is the only existing Lanczos-type process that has both deflation and look-ahead built in.

We now describe the symmetric Lanczos-type procedure. This algorithm could be stated directly in terms of $\mathbf{G}$, $\mathbf{C}$, and $\mathbf{B}$. However, we opted for a formulation that starts from a factorization of $\mathbf{G}$ of the form

$$\mathbf{G} = \mathbf{M}\,\mathbf{J}^{-1}\,\mathbf{M}^{\mathrm{T}}, \quad \text{where} \quad \mathbf{M},\ \mathbf{J} \in \mathbb{R}^{N\times N}. \tag{15}$$

Here $\mathbf{J}$ is assumed to be a "simple" matrix, such as a diagonal matrix. For example, if $\mathbf{M}$ is real symmetric positive definite, then we can choose $\mathbf{J} = \mathbf{I}$ (the $N \times N$ identity matrix). A factorization (15) can be computed via a suitable version of the Bunch-Parlett-Kaufman algorithm if $\mathbf{G}$ is indefinite, or a version of the Cholesky algorithm if $\mathbf{G}$ is symmetric positive definite; see, e.g., [9]. The algorithm is then formulated in terms of $\mathbf{M}$, $\mathbf{M}^{\mathrm{T}}$, $\mathbf{J}$, $\mathbf{C}$, and $\mathbf{B}$. This corresponds to rewriting the transfer function (6) as

$$\mathbf{Z}(s) = \left(\mathbf{M}^{-1}\mathbf{B}\right)^{\mathrm{T}} \left(\mathbf{J} + s\mathbf{M}^{-1}\mathbf{C}\,\mathbf{M}^{-\mathrm{T}}\right)^{-1} \left(\mathbf{M}^{-1}\mathbf{B}\right).$$

The algorithm generates a sequence of vectors, $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n, \ldots$, which are called the *Lanczos vectors*. In the absence of look-ahead steps, the Lanczos vectors are constructed to be $\mathbf{J}$-orthogonal:

$$\mathbf{v}_i^{\mathrm{T}}\,\mathbf{J}\,\mathbf{v}_n = \begin{cases} \delta_n & \text{if } i = n, \\ 0 & \text{if } i \neq n, \end{cases} \quad \text{for all} \quad i, n \geq 1. \tag{16}$$

If look-ahead steps do occur, then the Lanczos vectors are only cluster-wise $\mathbf{J}$-orthogonal, instead of the vector-wise $\mathbf{J}$-orthogonality (16). The Lanczos algorithm is an iterative procedure, with n denoting the iteration counter. At the nth step, the algorithm generates the nth Lanczos vector $\mathbf{v}_n$. For $n \leq p$, the vector $\mathbf{v}_n$ is obtained by $\mathbf{J}$-orthogonalization of the nth column of the starting block $\mathbf{M}^{-1}\mathbf{B}$. For $n > p$, the vector $\mathbf{v}_n$ is obtained by first multiplying a suitable previous vector, $\mathbf{v}_{n-p_c}$, with

$$\mathbf{A} = \mathbf{M}^{-1}\mathbf{C}\,\mathbf{M}^{-\mathrm{T}} \tag{17}$$

and then $\mathbf{J}$-orthogonalizing the resulting vector against a suitable subset of the previous Lanczos vectors. Here $p_c = p_c(n)$ denotes the current block size. Initially, $p_c = p$, and then within the algorithm p_c is reduced by one every time a deflation occurs. Thus, after n steps of the algorithm, the Lanczos vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ have been constructed; in addition, there are p_c "auxiliary" vectors, $\hat{\mathbf{v}}_{n+1}, \hat{\mathbf{v}}_{n+2}, \ldots, \hat{\mathbf{v}}_{p_c}$, that will be turned into Lanczos vectors or deflated in successive iterations.

Next, we present a precise statement of the symmetric Lanczos-type algorithm.

Algorithm 1 (Symmetric Lanczos-type method.)
INPUT:
Matrices $\mathbf{G} = \mathbf{G}^{\mathrm{T}} = \mathbf{M}\,\mathbf{J}\,\mathbf{M}^{\mathrm{T}}$, $\mathbf{C} = \mathbf{C}^{\mathrm{T}} \in \mathbb{R}^{N\times N}$;
A block $\mathbf{B} = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_p\end{bmatrix} \in \mathbb{R}^{N\times p}$.
OUTPUT:
The $p_1 \times p$ matrix $\boldsymbol{\rho}$ where
$p_1 = p - (\#$ of deflations during the first p steps);
The nonzero entries of the $n \times n$ matrices $\mathbf{T}_n$ and $\boldsymbol{\Delta}_n$ where n is the value of the iteration counter at termination.

0) *For* $i = 1, 2, \ldots, p$, *set* $\hat{\mathbf{v}}_i = \mathbf{J}^{-1}\mathbf{M}^{-1}\mathbf{b}_i$.
Set $p_c = p$. (p_c is the current block size.)
Set $\mathcal{I}_{\mathbf{v}} = \emptyset$. ($\mathcal{I}_{\mathbf{v}}$ records deflation.)
Set $\gamma = 1$, $\mathcal{C}_\gamma = \emptyset$, $\mathbf{V}^{(\gamma)} = \emptyset$. (Records clusters.)

For $n = 1, 2, \ldots,$ *do* (Build nth Lanczos vector $\mathbf{v}_n$.) :

1) (Deflate $\hat{\mathbf{v}}_n$ (if necessary) and obtain $\mathbf{v}_n$.)

1a) *Set* $\phi = n - p_c$.

1b) *For all* $i \in \mathcal{C}_\gamma$ (*in ascending order*), *set*

$$\tau_{i,\phi} = \frac{\mathbf{v}_i^{\mathrm{H}}\hat{\mathbf{v}}_n}{\|\mathbf{v}_i\|^2}, \ \hat{\mathbf{v}}_n = \hat{\mathbf{v}}_n - \mathbf{v}_i\tau_{i,\phi}, \ t_{i,\phi} = t_{i,\phi} + \tau_{i,\phi}.$$

1c) *If* $\|\hat{\mathbf{v}}_n\| >$ `dtol`, *then continue with step* 1h).
Otherwise, deflate $\hat{\mathbf{v}}_n$ *by doing the following*:

1d) *If* $p_c = 1$, *then stop.* (In this case, $\mathbf{Z}_n = \mathbf{Z}$.)

1e) *If* $\phi > 0$ *and the deflated vector* $\hat{\mathbf{v}}_n$ *is nonzero, then set* $\mathcal{I}_{\mathbf{v}} = \mathcal{I}_{\mathbf{v}} \cup \{\gamma(\phi)\}$.

1f) *For* $i = n, \ldots, n + p_c - 2$, *set* $\hat{\mathbf{v}}_i = \hat{\mathbf{v}}_{i+1}$. (The auxiliary vector $\hat{\mathbf{v}}_n$ is deflated. The indices of the remaining auxiliary vectors are reduced by one.)
Set $p_c = p_c - 1$. (The current block size is reduced.)

1g) *Go back to step* 1a).

1h) (Normalize $\hat{\mathbf{v}}_n$ to obtain $\mathbf{v}_n$.) *Set*

$$t_{n,n-p_c} = \|\hat{\mathbf{v}}_n\|_2 \quad \textit{and} \quad \mathbf{v}_n = \frac{\hat{\mathbf{v}}_n}{t_{n,n-p_c}}.$$

1i) (Update cluster information.)
Set $\gamma(n) = \gamma$, $\mathbf{V}^{(\gamma)} = \begin{bmatrix}\mathbf{V}^{(\gamma)} & \mathbf{v}_n\end{bmatrix}$, $\mathcal{C}_\gamma = \mathcal{C}_\gamma \cup \{n\}$.
If $\mathcal{C}_\gamma = \{n\}$, *set* $\gamma_{\mathbf{v}} = \gamma\,(\max\{1, n - p_c\})$.

2) (Compute $\boldsymbol{\Delta}^{(\gamma)}$ and check for end of cluster.)

2a) *Form* $\boldsymbol{\Delta}^{(\gamma)} = \left(\mathbf{V}^{(\gamma)}\right)^{\mathrm{T}}\mathbf{J}\,\mathbf{V}^{(\gamma)}$.

2b) (Decide if the current cluster is complete.)
If the matrix $\boldsymbol{\Delta}^{(\gamma)}$ *is singular* (*or in some sense "close" to singular*), *then continue with step* 3).
Otherwise, perform the following updates:

2c) (**J**-orthogonalize the vectors $\hat{\mathbf{v}}_i$, $n+1 \leq i \leq n+p_c-1$, against the vectors in the current cluster.)
For $i = n+1, n+2, \ldots, n+p_c-1$, *set*

$$\left[t_{j,i-p_c}\right]_{j\in\mathcal{C}_\gamma} = \left(\boldsymbol{\Delta}^{(\gamma)}\right)^{-1}\left(\mathbf{V}^{(\gamma)}\right)^{\mathrm{T}}\mathbf{J}\,\hat{\mathbf{v}}_i,$$

$$\hat{\mathbf{v}}_i = \hat{\mathbf{v}}_i - \mathbf{V}^{(\gamma)}\left[t_{j,i-p_c}\right]_{j\in\mathcal{C}_\gamma}.$$

2d) *Set* $\gamma = \gamma + 1$, $\gamma(n+1) = \gamma$, $\mathcal{C}_\gamma = \emptyset$, $\mathbf{V}^{(\gamma)} = \emptyset$.

3) (Obtain new vector $\hat{\mathbf{v}}_{n+p_c}$.)

3a) *Compute* $\mathbf{v} = \mathbf{J}^{-1}\mathbf{M}^{-1}\mathbf{C}\,\mathbf{M}^{-\mathrm{T}}\,\mathbf{v}_n$.

3b) (**J**-orthogonalize $\mathbf{v}$ against previous vectors.)
Set $\boldsymbol{\delta}$ *to the last column of the matrix* $\boldsymbol{\Delta}^{(\gamma-1)}$.
For all $k = \gamma_{\mathbf{v}}, \gamma_{\mathbf{v}}+1, \ldots, \gamma-2$, *set*

$$\begin{aligned} \big[t_{j,n}\big]_{j\in\mathcal{C}_k} &= \big(\boldsymbol{\Delta}^{(k)}\big)^{-1}\,\big(\big[t_{i,j}\big]_{i\in\mathcal{C}_{\gamma-1},j\in\mathcal{C}_k}\big)^{\mathrm{T}}\,\boldsymbol{\delta}, \\ \mathbf{v} &= \mathbf{v} - \mathbf{V}^{(k)}\big[t_{j,n}\big]_{j\in\mathcal{C}_k}. \end{aligned}$$

3c) (**J**-orthogonalization of $\mathbf{v}$ due to inexact deflation.)
For all $k \in \mathcal{I}_{\mathbf{v}}$ *with* $k < \gamma_{\mathbf{v}}$ (*in ascending order*), *set*

$$\begin{aligned} \big[t_{j,n}\big]_{j\in\mathcal{C}_k} &= \big(\boldsymbol{\Delta}^{(k)}\big)^{-1}\,\big(\mathbf{V}^{(k)}\big)^{\mathrm{T}}\mathbf{J}\,\mathbf{v}, \\ \mathbf{v} &= \mathbf{v} - \mathbf{V}^{(k)}\big[t_{j,n}\big]_{j\in\mathcal{C}_k}. \end{aligned}$$

3d) (**J**-orthogonalize $\mathbf{v}$ against cluster $\mathbf{V}^{(\gamma-1)}$.) *Set*

$$\begin{aligned} \big[t_{j,n}\big]_{j\in\mathcal{C}_{\gamma-1}} &= \big(\boldsymbol{\Delta}^{(\gamma-1)}\big)^{-1}\,\big(\mathbf{V}^{(\gamma-1)}\big)^{\mathrm{T}}\mathbf{J}\,\mathbf{v}, \\ \hat{\mathbf{v}}_{n+p_c} &= \mathbf{v} - \mathbf{V}^{(\gamma-1)}\big[t_{j,n}\big]_{j\in\mathcal{C}_{\gamma-1}}. \end{aligned}$$

4) (In the initial iterations, set up $\boldsymbol{\rho}$.) *If* $n \le p_c$, *set*

$$\rho_{n,i} = t_{n,i-p} \quad \textit{for all} \quad n - p_c + p \le i \le p.$$

To get the nth matrix-Padé approximant $\mathbf{Z}_n$ of $\mathbf{Z}$, we need the matrices $\boldsymbol{\rho}$, $\boldsymbol{\Delta}_n$, and $\mathbf{T}_n$, that are produced by n ($\ge p$) steps of the Lanczos-type algorithm. These quantities are defined as follows:

$$\begin{aligned} \mathbf{M}^{-1}\mathbf{B} &= \mathbf{V}_{p_1}\,\boldsymbol{\rho}, \quad \boldsymbol{\rho} = \big[\rho_{i,j}\big]_{1\le i\le p_1,\,1\le j\le p}, \\ \boldsymbol{\Delta}_n &= \mathbf{V}_n^{\mathrm{T}}\mathbf{J}\,\mathbf{V}_n = \mathrm{diag}\left(\boldsymbol{\Delta}^{(1)},\ldots,\boldsymbol{\Delta}^{(\gamma)}\right), \\ \mathbf{T}_n &= (\boldsymbol{\Delta}_n)^{-1}\,\mathbf{V}_n^{\mathrm{T}}\mathbf{J}\,\mathbf{A}\,\mathbf{V}_n = \big[t_{i,j}\big]_{1\le i,j\le n}. \end{aligned} \tag{18}$$

Here $\mathbf{V}_n = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_n\end{bmatrix}$ denotes the matrix that contains the first n Lanczos vectors as columns, and p_1 is defined to be the value of p_c at iteration step $n = p$. In general, $p_1 \le p$, and $p_1 = p$ if none of the vectors in the initial block $\mathbf{M}^{-1}\mathbf{B}$ has been deflated. In term of $\boldsymbol{\rho}$, $\boldsymbol{\Delta}_n$, and $\mathbf{T}_n$, the nth matrix-Padé approximant to $\mathbf{Z}$ is now given as follows:

$$\mathbf{Z}_n(s) = \boldsymbol{\rho}_n^{\mathrm{T}}\left(\boldsymbol{\Delta}_n^{-1} + s\mathbf{T}_n\boldsymbol{\Delta}_n^{-1}\right)^{-1}\boldsymbol{\rho}_n,\ \boldsymbol{\rho}_n = \begin{bmatrix}\boldsymbol{\rho}\\ \mathbf{0}\end{bmatrix}. \tag{19}$$

A rigorous proof of (19) can be found in [7].

5 Stability and Passivity

For RLC circuits, Padé-based reduced-order models are in general not stable and not passive. However, if the order n is large enough so that the reduced-order models are sufficiently accurate, then the reduced-order models defined by $\mathbf{Z}_n$ are almost stable and passive, and can in fact be made stable and passive by a suitable "post-processing" of $\mathbf{Z}_n$. Such post-processing techniques will be described elsewhere.

In this section, we show that the reduced-order models defined by $\mathbf{Z}_n$ are guaranteed to be stable and passive for RC, RL, and LC circuits.

5.1 Stability

Recall from the discussion in Section 2.2 that the symmetric matrices $\mathbf{G}$ and $\mathbf{C}$ in the formula (6) of $\mathbf{Z}$ are positive semi-definite for RC and RL circuits. Furthermore, for LC circuits, $\mathbf{Z}$ is given by (6) with s replaced by s^2, where $\mathbf{G}$ and $\mathbf{C}$ are again positive semi-definite.

Since $\mathbf{G}$ is positive semi-definite, it follows that $\mathbf{J} = \mathbf{I}$ in (15). As a result, the Lanczos vectors generated by Algorithm 1 are actually orthogonal, and thus $\boldsymbol{\Delta}_n = \mathbf{I}_n$ (the $n \times n$ identity matrix) for all n. Therefore, equation (19) reduces to

$$\mathbf{Z}_n(s) = \boldsymbol{\rho}_n^{\mathrm{T}}\left(\mathbf{I}_n + s\mathbf{T}_n\right)^{-1}\boldsymbol{\rho}_n. \tag{20}$$

Furthermore, the third relation in (18) now becomes

$$\mathbf{T}_n = \mathbf{V}_n^{\mathrm{T}}\mathbf{A}\mathbf{V}_n. \tag{21}$$

By (17), the matrix $\mathbf{A}$ is similar to the symmetric positive semi-definite matrix $\mathbf{C}$, and thus all eigenvalues of $\mathbf{A}$ are non-negative. Together with (21), it follows that $\mathbf{T}_n$ is symmetric positive semi-definite.

By (20), all poles of $\mathbf{Z}_n$ are of the form

$$s = -\frac{1}{\lambda}, \quad \text{where } \lambda \text{ is an eigenvalue of } \mathbf{T}_n.$$

Since $\mathbf{T}_n$ is symmetric positive semi-definite, $\lambda \ge 0$, and thus all poles of $\mathbf{Z}_n$ are non-positive. Moreover, it can be shown that a possible pole $s = 0$ is simple. Altogether, this proves that the reduced-order models defined by $\mathbf{Z}_n$ are stable for RC and RL circuits. The case of LC circuits can be handled similarly, using the formula (9) and the fact that the transformation $s \mapsto s^2$ maps the purely imaginary poles of an LC circuit into points on the negative real axis.

5.2 Passivity

We now show that the reduced-order models defined by $\mathbf{Z}_n$ are passive. Again, we only treat the RC and RL cases; LC circuits can be handled similarly after the transformation $s \mapsto s^2$ has been employed.

It is well known (see, e.g., [17, 2]) that the reduced-order model defined by $\mathbf{Z}_n$ is passive if, and only if, the following three conditions are satisfied:

(i) $\mathbf{Z}_n(s)$ has no poles in $\mathbb{C}_+ = \{\, s \in \mathbb{C} \mid \mathrm{Re}\, s > 0 \,\}$ (the right half of the complex plane);

(ii) $\mathbf{Z}_n(\bar{s}) = \overline{\mathbf{Z}_n(s)}$ for all $s \in \mathbb{C}$;

(iii) $\mathrm{Re}\left(\mathbf{x}^{\mathrm{H}}\mathbf{Z}_n(s)\,\mathbf{x}\right) \geq 0$ for all $s \in \mathbb{C}_+$ and $\mathbf{x} \in \mathbb{C}^p$.

It thus remains to verify that the function $\mathbf{Z}_n$ defined in (20) satisfies (i)–(iii). Condition (i) is satisfied in view of the stability of $\mathbf{Z}_n$. Condition (ii) follows trivially since $\mathbf{T}_n$ and $\boldsymbol{\rho}_n$ are are real matrices. Finally, we verify condition (iii). Let $s \in \mathbb{C}_+$. Since $\mathbf{T}_n$ is symmetric positive semi-definite, we have

$$\mathrm{Re}\left(\mathbf{y}^{\mathrm{H}}\left(\mathbf{I} + \bar{s}\,\mathbf{T}_n\right)\,\mathbf{y}\right) = \|\mathbf{y}\|_2^2 + (\mathrm{Re}\,s)\mathbf{y}^{\mathrm{H}}\mathbf{T}_n\mathbf{y} \geq 0 \quad (22)$$

for all $\mathbf{y} \in \mathbb{C}^n$. For any given $\mathbf{x} \in \mathbb{C}^p$, we insert $\mathbf{y} = (\mathbf{I} + s\,\mathbf{T}_n)\,\boldsymbol{\rho}_n\mathbf{x}$ into (22). This gives

$$\begin{aligned} 0 \leq \mathrm{Re}(\mathbf{y}^{\mathrm{H}}(\mathbf{I} + \bar{s}\,\mathbf{T}_n)\mathbf{y}) &= \mathrm{Re}(\mathbf{x}^{\mathrm{H}}\boldsymbol{\rho}_n(\mathbf{I} + s\mathbf{T}_n)^{-1}\boldsymbol{\rho}_n\mathbf{x}) \\ &= \mathrm{Re}(\mathbf{x}^{\mathrm{H}}\mathbf{Z}_n(s)\,\mathbf{x}). \end{aligned}$$

Hence the reduced-order model given by $\mathbf{Z}_n$ is passive.

6 Reduced-Circuit Synthesis

The reduced-order transfer function $\mathbf{Z}_n$ given by (19) can also be interpreted as a time-domain reduced-order model. By introducing a *state* vector $\mathbf{X}(s)$, equation (19) becomes

$$\left(\boldsymbol{\Delta}_n^{-1} + s\mathbf{T}_n\boldsymbol{\Delta}_n^{-1}\right)\mathbf{X}(s) = \boldsymbol{\rho}_n\mathbf{I}(s), \quad \mathbf{V}_n(s) = \boldsymbol{\rho}_n^{\mathrm{T}}\mathbf{X}(s),$$

and transformed back to time domain yields the system of first-order differential algebraic equations

$$\boldsymbol{\Delta}_n^{-1}\mathbf{x}(t) + \mathbf{T}_n\boldsymbol{\Delta}_n^{-1}\frac{d}{dt}\mathbf{x}(t) = \boldsymbol{\rho}_n\mathbf{i}(t), \quad \mathbf{v}_n(t) = \boldsymbol{\rho}_n^{\mathrm{T}}\mathbf{x}(t). \quad (23)$$

This system of only n equations can be used to replace the original, much larger, system (4), which describes the linear circuit. When the linear circuit represents a sub-block of a larger, nonlinear circuit, and the variables $\mathbf{v}(t)$ and $\mathbf{i}(t)$ represent the sub-block's interface currents and voltages, equations (23) together with the equations describing the rest of the nonlinear circuit form a smaller and easier to solve system of nonlinear differential algebraic equations.

In order to use existing circuit simulation tools, it is often useful to synthesize a reduced circuit, which implements exactly the reduced system (23). By means of algebraic manipulations, the equations (23) can be brought to a form that corresponds to an RLC topology, which generalizes either the first or the second Cauer forms. In general, there is no guarantee that the value of the elements will be positive. Nevertheless, when the reduced linear sub-circuit is stable and passive, negative-valued circuit elements will not affect the stability or the accuracy of the simulation. For the case $p = 1$ and for RC circuits, such a synthesis procedure is described in detail in [8].

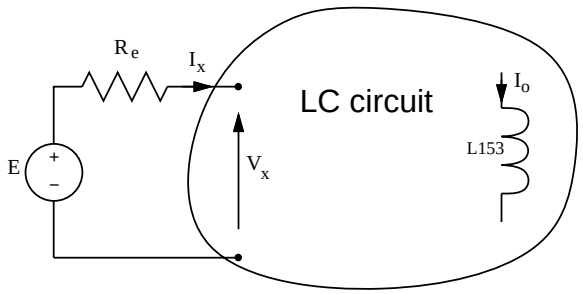

Figure 1: The PEEC circuit

7 Examples

In this section, we report numerical results for SyMPVL applied to three example circuits.

7.1 The PEEC Circuit

Our first example is the circuit resulting from the PEEC modeling of an electromagnetic problem described in [15]. The circuit consists of only inductors, capacitors, and inductive couplings and it is driven by a finite impedance source. We are interested in computing the response of this circuit, in this case the current flowing through one of the inductors.

Writing the KCL equations of the LC two-port and differentiating them with respect to time gives

$$s\mathbf{A}_l^{\mathrm{T}}\mathbf{I}_l + s\mathbf{A}_c^{\mathrm{T}}\mathbf{I}_c + s\mathbf{a}I_x = \mathbf{0}. \quad (24)$$

We substitute the inductor and capacitor equations,

$$s\mathbf{I}_l = \mathcal{L}^{-1}\mathbf{V}_l = \mathcal{L}^{-1}\mathbf{A}_l\mathbf{V} \quad \text{and} \quad \mathbf{I}_c = s\mathcal{C}\mathbf{V}_c = s\mathcal{C}\mathbf{A}_c\mathbf{V},$$

into (24), and obtain the nodal circuit equations

$$\mathbf{A}_l^{\mathrm{T}}\mathcal{L}^{-1}\mathbf{A}_l V + \mathbf{A}_c^{\mathrm{T}}\mathcal{C}\mathbf{A}_c s^2\mathbf{V} + s\mathbf{a}I_x = \mathbf{0}.$$

The output of interest, I_o, is selected among the inductor currents, using the column vector $\mathbf{b}$, i.e., $I_o = \mathbf{b}^{\mathrm{T}}\mathbf{I}_l$. Setting $\mathbf{G} = \mathbf{A}_l^{\mathrm{T}}\mathcal{L}^{-1}\mathbf{A}_l$, $\mathbf{C} = \mathbf{A}_c^{\mathrm{T}}\mathcal{C}\mathbf{A}_c$, and $\mathbf{l}^{\mathrm{T}} = \mathbf{b}^{\mathrm{T}}\mathcal{L}^{-1}\mathbf{A}_l$, we can expressed I_o as

$$I_o = \underbrace{-\mathbf{l}^{\mathrm{T}}(\mathbf{G} + s^2\mathbf{C})^{-1}\mathbf{a}}_{\alpha} \cdot I_x = \frac{\alpha}{R_e + Z_{in}}E.$$

The input impedance of the two-port is defined as the voltage V_x when a unit-valued current is applied:

$$V_x = a^{\mathrm{T}}\mathbf{V} = \underbrace{-\mathbf{a}^{\mathrm{T}}(\mathbf{G} + s^2\mathbf{C})^{-1}\mathbf{a}s}_{Z_{in}} \cdot I_x.$$

Setting $\mathbf{B} = \begin{bmatrix}\mathbf{a} & \mathbf{l}\end{bmatrix}$, the LC two-port is then characterized by the 2×2 matrix transfer function

$$\mathbf{Z}(s) = \mathbf{B}^{\mathrm{T}}(\mathbf{G} + s^2\mathbf{C})^{-1}\mathbf{B}. \quad (25)$$

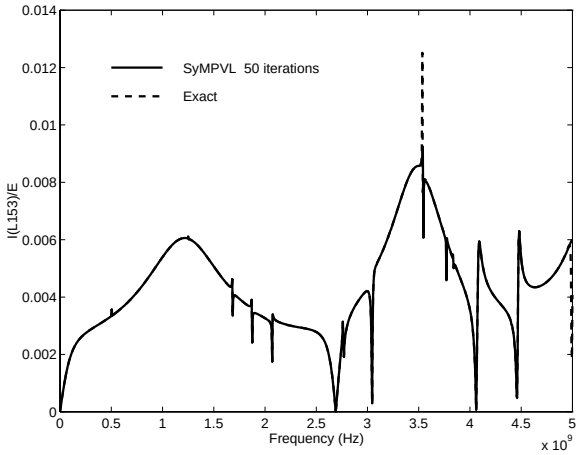

Figure 2: The PEEC circuit transfer function

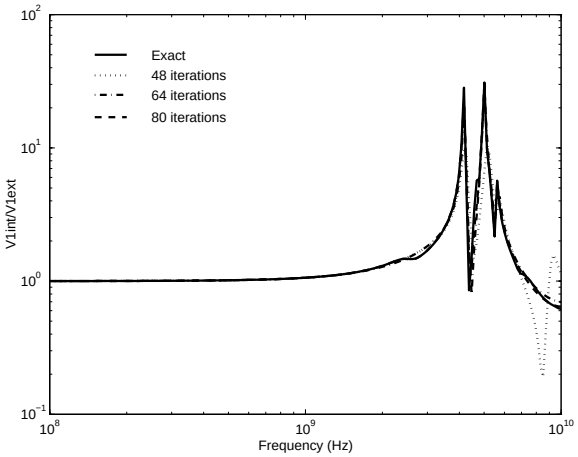

Figure 3: Package: Pin no. 1 external to Pin no. 1 internal

Note that $Z_{in} = -s\mathbf{Z}_{11}(s)$ and $\alpha = -\mathbf{Z}_{21}(s)$.

The SyMPVL algorithm is applied to compute the matrix-Padé approximation of (25). Since the matrix $\mathbf{G}$ is singular (in electrical terms, there is no DC path to the reference voltage (ground) from every node), we employ a frequency shift

$$\mathbf{Z}(s) = \mathbf{B}^{\mathrm{T}} \left(\mathbf{G} + s_0 \mathbf{C} + (s^2 - s_0)\mathbf{C} \right)^{-1} \mathbf{B} \qquad (26)$$

and apply SyMPVL with $\mathbf{G}$ replaced by $\mathbf{G}+s_0\mathbf{C}$ and s replaced by $s^2 - s_0$. An approximation of order $n = 50$ was needed to obtain a good match of the function, as illustrated in Figure 2. This approximation matches $2\lfloor n/m \rfloor = 2\lfloor 50/2 \rfloor = 50$ matrix moments. Running the algorithm 6 more iterations results in a perfect match. Note that this example is an LC circuit, and thus the approximation is stable and passive.

7.2 A Package Model

The next application is the analysis of a 64-pin package model used for an RF integrated circuit. Only eight of the package pins carry signal, the rest being either unused or necessary for supply voltages. The goal is to characterize the package as a 16-port component (8 exterior and 8 interior terminals) and apply SyMPVL model reduction on it. The reduced-order model can than replace the large package subcircuit in a circuit simulator resulting in considerable savings in simulation time.

The package model is described by an RLC circuit with approximatively 4000 circuit elements. The size of the nodal circuit matrices is about 2000.

Figures 3 and 4 show the voltage-to-voltage transfer function between the external terminal of pin no. 1 and the internal terminals of the same pin and the neighboring pin no. 2, respectively. The plots compare reduced models of order 48, 64, and 80 with an exact analysis. The results show that reduced-order models can indeed replace the full package subcircuit with little loss of accuracy. The reduction level depends on the desired accuracy. The most accurate of the models, which gives an almost perfect match of the frequency response, only requires 80 state variables compared to 2000 for the full subcircuit. Since the cost of nonlinear circuit simulation is superlinear in the number of state variables the computational savings can be significant.

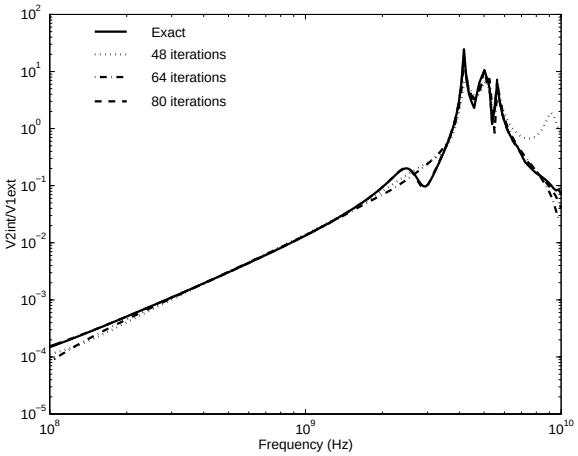

Figure 4: Package: Pin no. 1 external to Pin no. 2 internal

7.3 Synthesized interconnect circuit

The original circuit of this example represents an interconnect parasitic network extracted as an RC cir-

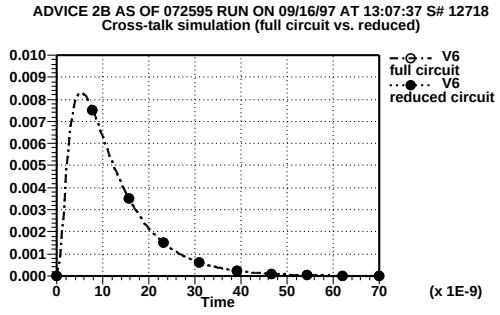

Figure 5: Synthesized vs. full circuit in time domain

cuit. The purpose of the extraction is cross-talk analysis, and therefore the circuit models several capacitively coupled wires. The original circuit consists of 36620 capacitors, 1355 resistors, and 1350 nodes. The circuit is connected with logic gates at 17 ports. In a simulation, the interconnect network will contribute 1350 nodal equations to the nonlinear system that describes the circuit. We applyed the SyMPVL algorithm followed by the synthesis of a reduced circuit. The synthesis resulted in a circuit with 170 capacitors, 459 resistors, and 34 nodes, thus reducing the interconnect contribution from 1350 to only 34 nodal equations. Figure 5 shows the results of simulating the full interconnect compared to the reduced circuits. The waveforms are indistinguishable and the CPU time required for the transient analysis went down from $132s$ to $2.15s$.

8 Concluding Remarks

In this paper, we have introduced SyMPVL as a tool to compute reduced-order models for large passive linear circuit multi-ports. The algorithm is more efficient than the more general MPVL and has the property of generating guaranteed stable and passive models for the special cases of RC, RL, and LC circuits. The reduced-order model can be synthesized exactly as a circuit, with possibly negative-valued components. While the reduced-order models generated by SyMPVL cannot be guaranteed to be stable and passive for general RLC circuits, they can be made stable and passive using suitable "post-processing" techniques.

The authors would like to thank Homer Wang for providing the package model.

References

[1] J.I. Aliaga, D.L. Boley, R.W. Freund, and V. Hernández, "A Lanczos-type algorithm for multiple starting vectors," Numerical Analysis Manuscript No. 96–18, Bell Labs, Murray Hill, NJ, Sep. 1996.

[2] B.D.O. Anderson and S. Vongpanitlerd, *Network Analysis and Synthesis*, Englewood Cliffs, NJ: Prentice-Hall, 1973.

[3] G.A. Baker, Jr. and P. Graves-Morris, *Padé Approximants*, Second Edition, New York, NY: Cambridge University Press, 1996.

[4] P. Feldmann and R.W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," in *Proc. Euro-DAC*, Sep. 1994.

[5] P. Feldmann and R.W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 639–649, May 1995.

[6] P. Feldmann and R.W. Freund, "Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm," in *Proc. 32nd ACM/IEEE DAC*, June 1995.

[7] R.W. Freund, "An extension of the Lanczos-Padé connection to the matrix case," Numerical Analysis Manuscript, Bell Labs, Murray Hill, NJ, in preparation.

[8] R.W. Freund and P. Feldmann, "Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm," in *Tech. Dig. IEEE/ACM ICCAD*, Nov. 1996.

[9] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Third Edition. Baltimore, MD: The Johns Hopkins University Press, 1996.

[10] W.B. Gragg, "Matrix interpretations and applications of the continued fraction algorithm," *Rocky Mountain J. Math.*, vol. 4, pp. 213–225, 1974.

[11] K.J. Kerns, I.L. Wemple, and A.T. Yang, "Stable and efficient reduction of substrate model networks using congruence transformations," in *Tech. Dig. IEEE/ACM ICCAD*, Nov. 1995.

[12] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *J. Res. Nat. Bur. Standards*, vol. 45, pp. 255–282, 1950.

[13] L.T. Pillage and R.A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352–366, Apr. 1990.

[14] V. Raghavan, R.A. Rohrer, L.T. Pillage, J.Y. Lee, J.E. Bracken, and M.M. Alaybeyi, "AWE–inspired," in *Proc. IEEE CICC*, May 1993.

[15] A.E. Ruehli, "Equivalent circuit models for three-dimensional multiconductor systems," *IEEE Trans. Microwave Theory and Tech.*, vol. 22, pp. 216–221, Mar. 1974.

[16] L.M. Silveira, M. Kamon, I. Elfadel, and J. White, "A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits," in *Tech. Dig. IEEE/ACM ICCAD*, Nov. 1996.

[17] M.R. Wohlers, *Lumped and Distributed Passive Networks*, New York, N.Y.: Academic Press, 1969.