Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm*

Roland W. Freund
Bell Laboratories
Lucent Technologies
Room 2C–420
700 Mountain Avenue
Murray Hill, NJ 07974–0636, USA

Peter Feldmann
Bell Laboratories
Lucent Technologies
Room 2C–217
700 Mountain Avenue
Murray Hill, NJ 07974–0636, USA

Abstract. *This paper discusses the analysis of large linear electrical networks consisting of passive components, resistors, capacitors, inductors, transformers, etc. Such networks result in a symmetric formulation of circuit equations. The paper introduces SyPVL, an efficient and numerically stable algorithm for the computation of reduced-order models of large, linear, passive networks. SyPVL represents the specialization of the more general PVL algorithm, to symmetric problems. Besides the gain in efficiency over PVL, SyPVL also preserves the symmetry of the problem, and, as a consequence, can often guarantee the stability of the resulting reduced order models. Moreover, the resulting reduced-order models can be synthesized as actual physical circuits, thus facilitating compatibility with existing analysis tools. The application of SyPVL is illustrated in the paper with a few interconnect analysis examples.*

*Numerical Analysis Manuscript No. 96–13, Bell Laboratories, Murray Hill, New Jersey, May 1996. Available on WWW at `http://cm.bell-labs.com/cs/doc/96`.

1. Introduction

Many of today's trends in VLSI technology, such as decrease in conductor sizes and spacings, increase in the number of layers of metal, etc., concur towards pushing interconnect to become one of the major concerns of VLSI designers. Unfortunately, the supply of CAD solutions for accurate and efficient modeling and analysis of interconnect is extremely inadequate.

At the chip level, interconnect can be modeled by large, lumped, linear networks consisting mainly of capacitors, resistors, and sometimes even inductors (mainly for modeling of large global networks such as power distribution or clock distribution). These networks are typically generated by automatic parasitic extraction programs that often produce vast quantities of data even for modest interconnects. In older VLSI technologies, when the effects of wire resistance could be safely neglected, the processing of extraction program output was simple. One only needed to add up the contributions of each geometric feature into a single number, the total net capacitance. Unfortunately, this simplification is only possible if the effect of wire resistance is negligible. In actuality, quite the contrary is happening: resistive effects increase in importance as wire sizes are reduced. Ignoring wire resistance in the current and future generations of VLSI technology leads to unacceptable errors in chip performance estimation.

The main challenges brought about by the emergence of the interconnect problem are: (1) to capture the vast amount of information produced by extraction programs, (2) to compress it, and (3) to make it available to designers in the form of useful models that can be integrated into the general design methodology. In this paper, we introduce a powerful new algorithm, SyPVL, that can serve as the engine for many of these tasks.

SyPVL is an efficient and numerically stable algorithm for the computation of reduced-order models for large, linear, passive networks. The reduced-order models are based on a suitable Padé approximation. SyPVL represents the specialization of the more general PVL (Padé via Lanczos) algorithm [3] to symmetric problems. We show in this paper that most problems of interest in the analysis and modeling of large linear electrical networks consisting of passive components (resistors, capacitors, inductors, transformers, etc.) have symmetric formulations. Besides being approximatively twice as efficient as PVL, SyPVL also preserves the symmetry of the problem. Due to symmetry, the stability of the resulting reduced-order models can be guaranteed for a number of important special cases, such as RC, RL, and LC networks. Moreover, the reduced-order models can be synthesized as actual physical circuits, thus facilitating compatibility with existing analysis tools.

This paper is organized as follows. First we discuss the formulation of equation systems for circuits consisting of only passive components. We show that the formulation is almost always symmetric. Then, we derive the SyPVL algorithm, and discuss implementation aspects that further increase its efficiency for circuits that have tree or "almost" tree structure. We then describe two synthesis techniques that allow the implementation of the reduced-order model as a significantly smaller circuit. Finally, we present numerical results for a few examples.

2. Formulation

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.

2.1. General RLC circuits

The connectivity of the circuit can be captured using a directional graph. The nodes of the graph correspond to the nodes of the circuit, and the edges of the graph correspond to each of the circuit elements. An arbitrary direction is assigned to graph edges, so one can distinguish between the source and destination nodes. The adjacency matrix of the directional graph describes the connectivity of the circuit. In the sequel, $\mathcal{A}$ always denotes this adjacency matrix. Each row of $\mathcal{A}$ corresponds to a graph edge and, therefore, to a circuit element. Each column of $\mathcal{A}$ corresponds to a graph or circuit node. The column corresponding to the datum (ground) node of the circuit is omitted in order to remove redundancy. A row of $\mathcal{A}$ contains $+1$ in the column corresponding to the source node, -1 in the column corresponding to the destination node, and 0 everywhere else. It is easy to see (e.g., [15]) that Kirchhoff's current and voltage laws, which depend only on the connectivity of the circuit, can be expressed using the adjacency matrix:

$$\begin{aligned} \text{KCL:} \qquad & \mathcal{A}^T i_b = 0, \\ \text{KVL:} \qquad & \mathcal{A} v_n = v_b. \end{aligned} \tag{1}$$

Here, i_b and v_b are the vectors of branch currents and voltages, respectively, and 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 just by current sources. In this case, the matrix $\mathcal{A}$ and the vectors i_b and v_b can be partitioned according to circuit-element types:

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_I \\ \mathcal{A}_G \\ \mathcal{A}_C \\ \mathcal{A}_L \end{bmatrix}, \quad i_b = \begin{bmatrix} i_I \\ i_G \\ i_C \\ i_L \end{bmatrix}, \quad v_b = \begin{bmatrix} v_I \\ v_G \\ v_C \\ v_L \end{bmatrix}. \tag{2}$$

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 equations that describe the physical behavior of the circuit elements, the so-called *branch constitutive relationships* (BCRs). In the case of RLC circuits, the BCRs are as follows:

$$i_I = -I_t(t), \quad i_G = \mathcal{G} v_G, \quad i_C = \mathcal{C} \frac{d}{dt} v_C, \quad v_L = \mathcal{L} \frac{d}{dt} i_L. \tag{3}$$

Here, $I_t(t)$ is the vector of current-source values, and $\mathcal{G}$ and $\mathcal{C}$ are appropriately-sized diagonal matrices whose diagonal entries are the conductance and capacitance values of each element. It is clear that these values are positive for any physical circuit, and consequently, both matrices $\mathcal{G}$ and $\mathcal{C}$ are symmetric positive definite. The inductance matrix $\mathcal{L}$ is also symmetric positive definite. In the absence of inductive coupling, it is a diagonal matrix. Otherwise, $\mathcal{L}$ has off-diagonal entries, which are introduced by inductive coupling.

The modified nodal formulation (MNA) of the circuit equations is obtained by combining the Kirchhoff equations (1) with the BCR's (3), and eliminating as many current unknowns as possible. For the case of RLC circuits, only the inductor currents need to be left as unknowns. The resulting MNA equations are as follows:

$$\begin{aligned} \mathcal{A}_G^T \mathcal{G} \mathcal{A}_G v_n + \mathcal{A}_C^T \mathcal{C} \mathcal{A}_C \frac{d}{dt} v_n + \mathcal{A}_L^T i_L &= \mathcal{A}_I^T I_t(t), \\ \mathcal{A}_L v_n - \mathcal{L} \frac{d}{dt} i_L &= 0. \end{aligned} \tag{4}$$

Rewriting (4) in matrix form gives

$$\mathbf{G} x + \mathbf{C} \frac{d}{dt} x = b I_t(t), \tag{5}$$

where we set

$$\mathbf{G} = \begin{bmatrix} \mathcal{A}_G^T \mathcal{G} \mathcal{A}_G & \mathcal{A}_L^T \\ \mathcal{A}_L & 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathcal{A}_C^T \mathcal{C} \mathcal{A}_C & 0 \\ 0 & -\mathcal{L} \end{bmatrix}, \quad x = \begin{bmatrix} v_n \\ i_L \end{bmatrix}, \quad b = \begin{bmatrix} \mathcal{A}_I^T \\ 0 \end{bmatrix}. \tag{6}$$

An important analysis task for RLC circuits is to treat them as p-terminal components and determine their network functions. Since we allowed only current sources in our formulation, it is easier to determine the Z-parameters. By applying the Laplace transform to (5) and assuming zero initial conditions, we obtain

$$\begin{aligned} (\mathbf{G} + s\mathbf{C}) X &= b I_s(s), \\ V_I &= [\, \mathcal{A}_I \quad 0 \,] X. \end{aligned} \tag{7}$$

Here, X, $I_s(s)$, and V_I denote the Laplace transforms of the unknown vector x, the excitation current $I_t(t)$, and the vector of voltages across the excitation sources, respectively. Eliminating X in (7) gives

$$V_I = [\, \mathcal{A}_I \quad 0 \,] X = \underbrace{b^T (\mathbf{G} + s\mathbf{C})^{-1} b}_{=:\, Z(s)} I_s(s). \tag{8}$$

2.2. Special cases: RC, RL, and LC circuits

The matrices $\mathbf{G}$ and $\mathbf{C}$ in (5) are always symmetric, but, for RLC circuits, they are indefinite in general. However, for the important special cases of RC, RL, and LC circuits, the formulation can be modified so that the matrices $\mathbf{G}$ and $\mathbf{C}$ are positive semi-definite.

First, consider the special case of RC circuits. Here, the matrices $\mathcal{L}$, $\mathcal{A}_L$, and the vector i_L in the MNA equations (4) are empty. Hence, the Laplace transform of (4) becomes

$$\mathbf{G}X + s\mathbf{C}X = bI_s(s), \tag{9}$$

where

$$\mathbf{G} = \mathcal{A}_G^T \mathcal{G} \mathcal{A}_G, \quad \mathbf{C} = \mathcal{A}_C^T \mathcal{C} \mathcal{A}_C, \quad X = V_n, \quad b = \mathcal{A}_I^T. \tag{10}$$

Clearly, the matrices $\mathbf{G}$ and $\mathbf{C}$ are positive semi-definite.

Next, consider the special case of RL circuits. The matrices $\mathcal{C}$ and $\mathcal{A}_C$ in (4) are empty, and hence the Laplace transform of (4) reduces to

$$\begin{aligned} \mathcal{A}_G^T \mathcal{G} \mathcal{A}_G V_n + \mathcal{A}_L^T I_L &= \mathcal{A}_I^T I_s(s), \\ \mathcal{A}_L V_n - \mathcal{L} s I_L &= 0. \end{aligned} \tag{11}$$

By multiplying the first equation in (11) by s and substituting sI_L from the second, we obtain

$$\mathbf{G}V_n + s\mathbf{C}V_n = s\mathcal{A}_I^T I_s(s), \tag{12}$$

where

$$\mathbf{G} = \mathcal{A}_L^T \mathcal{L}^{-1} \mathcal{A}_L, \quad \mathbf{C} = \mathcal{A}_G^T \mathcal{G} \mathcal{A}_G. \tag{13}$$

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

Finally, we consider the case of LC circuits. Here, the matrices $\mathcal{G}$ and $\mathcal{A}_G$ in (4) are empty, and the Laplace transform of (4) reduces to

$$\begin{aligned} \mathcal{A}_C^T \mathcal{C} \mathcal{A}_C s V_n + \mathcal{A}_L^T I_L &= \mathcal{A}_I^T I_s(s), \\ \mathcal{A}_L V_n - \mathcal{L} s I_L &= 0. \end{aligned} \tag{14}$$

By multiplying the first equation in (14) by s and substituting sI_L from the second, we obtain

$$\mathbf{G}V_n + s^2\mathbf{C}V_n = s\mathcal{A}_I^T I_s(s), \tag{15}$$

where

$$\mathbf{G} = \mathcal{A}_L^T \mathcal{L}^{-1} \mathcal{A}_L, \quad \mathbf{C} = \mathcal{A}_C^T \mathcal{C} \mathcal{A}_C. \tag{16}$$

Clearly, the matrices $\mathbf{G}$ and $\mathbf{C}$ are positive semi-definite.

3. The SyPVL algorithm

In this section, we describe the SyPVL algorithm that stably computes Padé approximations of the transfer functions (8), $Z(s)$, for general RLC circuits.

3.1. Padé approximants

Let $\mathbf{G}$ and $\mathbf{C}$ be general, not necessarily symmetric, $N \times N$ matrices, and assume that $\mathbf{G}$ is nonsingular. Let b and c be nonzero vectors of length N. We set

$$Z(s) = c^T \left(\mathbf{G} + s\mathbf{C}\right)^{-1} b, \quad s \in \mathbb{C}. \tag{17}$$

A rational function $Z_q(s)$ of type $(q-1, q)$, i.e.,

$$Z_q(s) = n_{q-1}(s)/d_q(s), \tag{18}$$

where n_{q-1} and d_q are polynomials of degree at most $q-1$ and q, respectively, is called a qth Padé approximant of the function $Z(s)$ if

$$Z_q(s) = Z(s) + \mathcal{O}(s^{2q});$$

see, e.g., [2].

The PVL (Padé via Lanczos) algorithm described in [3] computes these Padé approximants stably by exploiting the well-known connection [10] between Padé approximants and the Lanczos process [11]. More precisely, PVL consists of running the Lanczos process for q steps to obtain a $q \times q$ tridiagonal matrix $\mathbf{T}$. The qth Padé approximant $Z_q(s)$ of $Z(s)$, is then evaluated by means of the representation

$$Z_q(s) = c^T \mathbf{G}^{-1} b \; e_1^T (\mathbf{I} - s\mathbf{T}_q)^{-1} e_1. \tag{19}$$

Here, $e_1 = [1 \quad 0 \quad \cdots \quad 0]^T$ is the first unit vector of length q, and $\mathbf{I}$ is the $q \times q$ identity matrix. The main computational work in PVL are q products of the type

$$\mathbf{G}^{-1}\mathbf{C}v, \quad \text{where } v \text{ is a vector of length } n, \tag{20}$$

and q products of the type

$$\mathbf{C}^T\mathbf{G}^{-T}w, \quad \text{where } w \text{ is a vector of length } n. \tag{21}$$

3.2. Exploiting symmetry

We now return to general RLC circuits. Recall that, in view of (8), the functions $Z(s)$ associated with such circuits are of the form

$$Z(s) = b^T \left(\mathbf{G} + s\mathbf{C}\right)^{-1} b, \quad \text{where} \quad \mathbf{G} = \mathbf{G}^T \quad \text{and} \quad \mathbf{C} = \mathbf{C}^T. \tag{22}$$

It is possible to exploit the symmetry in (22) when using PVL to compute Padé approximants $Z_q(s)$ of (22). The basic observation here is that the underlying Lanczos process simplifies (see, e.g., [7]) when applied to a product of symmetric matrices. The resulting symmetric

Lanczos process requires only half the work and storage, compared to the general Lanczos process. In particular, the symmetric Lanczos process only involves the q products (20), since the need for the products (21) can be eliminated using the symmetry of $\mathbf{G}$ and $\mathbf{C}$.

Next, we state a version of this symmetric Lanczos algorithm tailored to the computation of the "symmetric" transfer functions (22).

Algorithm 1 (Symmetric Lanczos algorithm)

0) Set $v = b$, $v_0 = 0$, and $\delta_0 = 1$.

For $n = 1, 2, \ldots, q$ do:

1) Solve $\mathbf{G}t = v$ for t.
2) Compute $\delta = v^T t$.
   If $\delta = 0$, then stop.
3) Set

$$\begin{aligned} \gamma_n &= \sqrt{|\delta|}, \quad \beta_n = (\operatorname{sign} \delta)\gamma_n, \\ v_n &= \frac{v}{\gamma_n}, \quad t = \frac{t}{\gamma_n}. \end{aligned}$$

4) Set

$$\begin{aligned} v &= \mathbf{C}t - v_{n-1}\beta_n, \\ \alpha_n &= t^T v, \\ v &= v - v_n \alpha. \end{aligned}$$

After running Algorithm 1, we can set up the $q \times q$ tridiagonal matrix

$$\mathbf{T}_q = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \gamma_2 & \alpha_2 & \beta_3 & \ddots & \vdots \\ 0 & \gamma_3 & \alpha_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_q \\ 0 & \cdots & 0 & \gamma_q & \alpha_q \end{bmatrix}. \tag{23}$$

Using the results on PVL for general functions (17), it follows that the qth Padé approximant $Z_q(s)$ of the symmetric transfer function (22) is given by

$$Z_q(s) = b^T \mathbf{G}^{-1} b \; e_1^T (\mathbf{I} - s\mathbf{T}_q)^{-1} e_1, \tag{24}$$

with $\mathbf{T}_q$ the matrix defined in (23).

We refer to the resulting computational procedure of obtaining Padé approximants of symmetric transfer function (22) via Algorithm 1 and formula (24) as the SyPVL (Symmetric PVL) algorithm, in order to distinguish it from the PVL algorithm for general transfer functions.

Note that $\beta_n = (\operatorname{sign}\delta)\gamma_n = \pm\gamma_n$ for all n, and hence the tridiagonal matrix is symmetric up to possible $\pm$ pairs γ_n and β_n. For general RLC circuits, in cannot be excluded that $\beta_n = -\gamma_n$ occurs, and thus, the matrix $\mathbf{T}_q$ can indeed be nonsymmetric. However, for the special cases of RC, RL, and LC circuits, we can guarantee that $\mathbf{G}$ is positive definite, and this implies that always $\delta \geq 0$ in Algorithm 1. It follows that $\beta_n = \gamma_n$ for all n, and thus, for the special cases of RC, RL, and LC circuits, we have that the tridiagonal matrix

$$\mathbf{T}_q = \begin{bmatrix} \alpha_1 & \gamma_2 & 0 & \cdots & 0 \\ \gamma_2 & \alpha_2 & \gamma_3 & \ddots & \vdots \\ 0 & \gamma_3 & \alpha_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \gamma_q \\ 0 & \cdots & 0 & \gamma_q & \alpha_q \end{bmatrix} \tag{25}$$

is always real and symmetric.

3.3. Efficient solution of linear systems

Recall that Step 1 of Algorithm 1 requires the solution of linear systems with $\mathbf{G}$ as coefficient matrix. Often the matrix $\mathbf{G}$ has a special structure that can be exploited in the solution process of these linear systems. For example, for RC circuits the matrix $\mathbf{G}$ has the form

$$\mathbf{G} = \mathcal{A}_G^T \mathbf{G} \mathcal{A}_G.$$

In many applications of interest the graph associated with the adjacency matrix $\mathcal{A}_G$ forms a tree. In this case, the adjacency matrix is square (the number of edges equals the number of nodes minus one), and furthermore, it can be shown that, when properly ordered, it is also triangular. Therefore the matrix $\mathbf{G}$ is already decomposed in its Cholesky factors, and thus systems with $\mathbf{G}$ can be solved trivially by means of backsolves.

In some applications, the structure is "almost" a tree, in the sense that a "small" number of co-tree branches exists. If the number of the co-tree branches is not excessive, the Sherman-Morrison formula [8] is the most efficient way to handle the small-rank modification to the factored matrix. This approach is equivalent to the one used in RICE [14].

3.4. Stability of reduced-order models

An RLC circuit is stable if all the poles λ of the symmetric transfer function $Z(s)$ defined in (22) lie in the negative half-plane

$$\mathbb{C}_- = \{ s \in \mathbb{C} \mid \Re s \leq 0 \}$$

and if the poles λ on the imaginary axis, i.e., $\Re\lambda = 0$, are simple. A well-known disadvantage of Padé-based reduced-order modeling is that the resulting reduced circuits need not be stable, even if the original RLC circuit is stable.

However, for the special case of RC, RL, and LC circuits, we can guarantee that the reduced-order models associated with the Padé approximant $Z_q(s)$ are stable if the circuit associated with $Z(s)$ is stable. In fact, for stable RC and RL circuits, it is well known that the poles of $Z(s)$ are all real and negative, i.e., we have

$$\lambda_{\min} \leq \lambda \leq \lambda_{\max} < 0 \quad \text{for all poles } \lambda \text{ of } Z(s). \tag{26}$$

Furthermore, the poles λ of $Z(s)$ are just given as the inverses of the eigenvalues of the matrix $-\mathbf{G}^{-1}\mathbf{C}$. Now, one of the many properties of the symmetric Lanczos Algorithm 1 is that, if $\mathbf{G}$ is positive definite (which is true for RC and RL circuits), the zeros of $\mathbf{T}_q$ interlace the eigenvalues of the matrix $-\mathbf{G}^{-1}\mathbf{C}$. Together with (26), it follows that the poles μ of $Z_q(s)$ satisfy

$$\lambda_{\min} \leq \mu \leq \lambda_{\max} < 0 \quad \text{for all poles } \mu \text{ of } Z_q(s).$$

Hence, the reduced circuits associated with $Z_q(s)$ are stable.

The case of LC circuits can be treated similarly. Here the poles are all purely imaginary, simple, and complex conjugate. Setting $\sigma = s^2$, this case can be reduced to the previously-solved case.

4. Synthesis

Digital CMOS circuits are composed of logic cells (clusters of tightly-coupled MOS devices) connected to each other by conductors, as shown in Figure 4.1 (a). The conductors are implemented using several polysilicon and metal layers. In current technologies, the interconnecting conductors can no longer be accurately modeled in terms of their capacitance only. Resistance and coupling capacitances have an increasingly important contribution to signal propagation. As a consequence, parasitic extraction programs are used to analyze the layout and generate a model of the interconnection network. The raw result of these extraction programs may be a large network of resistors and capacitors reflecting the intricate topology of the interconnections. The digital circuit-design methodology relies on the heavy use of high-level approximate analysis tools that, at best, will perform inefficiently in the

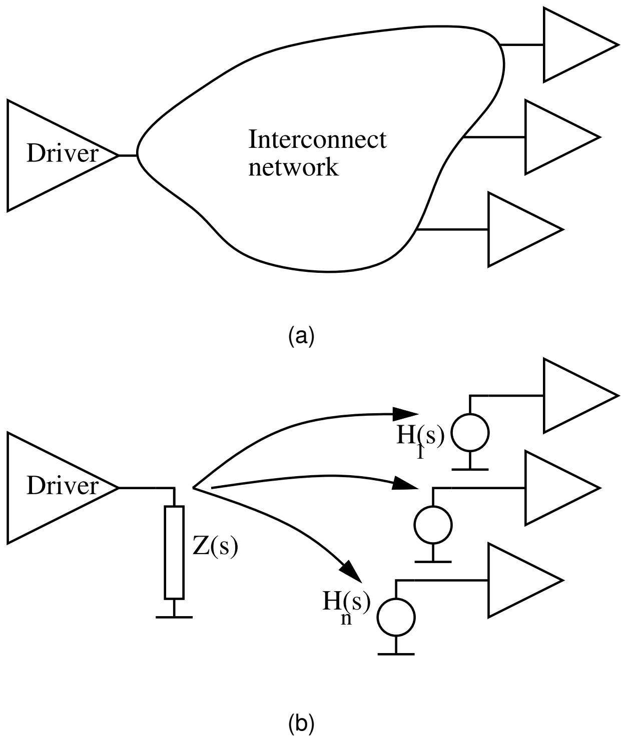

Figure 4.1: Interconnection network and reduced model

presence of these large RC networks. There is therefore an urgent need to simplify the large RC interconnect networks. One such possible simplified model compatible with event driven digital circuit analysis tools is illustrated in Figure 4.1 (b).

4.1. Driving-point impedance

We are interested in modeling the impedance seen by the driver circuit. For the case of CMOS technology, the loading of the driven nonlinear logic cells can be safely approximated by linear capacitors. The driving-point impedance is obtained by analyzing an RC network configured as a one-port. Therefore, the driving-point impedance can be characterized by either the impedance $Z(s)$ or the admittance $Y(s)$ of the port. In the most common case, the resistive leakage to ground is negligible and therefore the DC impedance, $Z(0)$, will be infinite. We employ a network-reduction technique based on a Padé approximation that is linked to a Taylor expansion around a point. In our case, we are interested in maximum accuracy at low frequency and thus the most natural choice for an expansion point is $s = 0$. Consequently, it is natural to approximate the admittance of the port, which is always well defined for $s = 0$. The admittance of a port is the current flowing through the port when a unit voltage is applied to it.

The RC circuit is characterized by the following equations:

$$\begin{array}{ll} \text{KVL:} & \mathcal{A}_V v_n = v_V, \quad \mathcal{A}_G v_n = v_G, \quad \mathcal{A}_C v_n = v_C, \\ \text{KCL:} & \mathcal{A}_V^T i_V + \mathcal{A}_G^T i_G + \mathcal{A}_C^T i_C = 0, \\ \text{BCR:} & v_V = 1, \quad i_G = \mathcal{G} v_G, \quad i_C = \mathcal{C}\dfrac{d}{dt} v_C. \end{array} \tag{27}$$

From here we obtain the modified nodal circuit equations:

$$\begin{aligned} \mathcal{A}_V v_n &= 1, \\ \mathcal{A}_V^T i_V + \mathcal{A}_G^T \mathcal{G} \mathcal{A}_G v_n + \mathcal{A}_C^T \mathcal{C} \mathcal{A}_C \frac{d}{dt} v_n &= 0. \end{aligned} \tag{28}$$

We assume that only one voltage source attached to node 1, and hence $\mathcal{A}_V = e_1^T$. Writing the modified nodal circuit equations in matrix form and then applying the Laplace transform, we obtain

$$\begin{bmatrix} 0 & e_1^T \\ e_1 & \mathcal{A}_G^T \mathcal{G} \mathcal{A}_G \end{bmatrix} \begin{bmatrix} i_V \\ v_n \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{A}_G^T \mathcal{C} \mathcal{A}_G \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_V \\ v_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{29}$$

and

$$\mathbf{G}X + s\mathbf{C}X = e_1, \tag{30}$$

where

$$\mathbf{G} = \begin{bmatrix} 0 & e_1^T \\ e_1 & \mathcal{A}_G^T \mathcal{G} \mathcal{A}_G \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{A}_G^T \mathcal{C} \mathcal{A}_G \end{bmatrix}.$$

The current of the one port is $i_V = e_1^T X$ and thus

$$Y(s) = e_1^T (\mathbf{G} + s\mathbf{C})^{-1} e_1. \tag{31}$$

Unfortunately, the situation when $Y(0) = 0$ creates certain difficulties for the underlying Lanczos process. In order to avoid them, we chose a different formulation of the problem.

We observe that the equation of the source, $\mathcal{A}_V v_n = 1$, in fact means $v_1 = 1$. Therefore, we can eliminate the node voltage v_1 and the corresponding KCL of node 1 from the circuit equations. We partition the adjacency matrices accordingly:

$$\mathcal{A}_G = [\, a_1 \quad \mathcal{A}'_G \,], \quad \mathcal{A}_C = [\, 0 \quad \mathcal{A}'_C \,]. \tag{32}$$

Note that the first column of $\mathcal{A}_C$ is 0 since no capacitor is connected to node 1. By (29) and (32), we now have

$$\begin{aligned} 0 &= e_1 i_V + [\, a_1 \quad \mathcal{A}'_G \,]^T \mathcal{G} [\, a_1 \quad \mathcal{A}'_G \,] v_n + [\, 0 \quad \mathcal{A}'_G \,]^T \mathcal{C} [\, 0 \quad \mathcal{A}'_C \,] \frac{d}{dt} v_n \\ & e_1 i_V + \begin{bmatrix} a_1^T \mathcal{G} a_1 & a_1^T \mathcal{G} \mathcal{A}'_G \\ \mathcal{A}'^T_G \mathcal{G} a_1 & \mathcal{A}'^T_G \mathcal{G} \mathcal{A}'_G \end{bmatrix} \begin{bmatrix} v_1 \\ v'_n \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{A}'^T_C \mathcal{C} \mathcal{A}'_C \end{bmatrix} \frac{d}{dt} \begin{bmatrix} v_1 \\ v'_n \end{bmatrix}. \end{aligned}$$

Taking the Laplace transform of these equations, we obtain

$$\begin{aligned} I_V + a_1^T \mathcal{G} a_1 + a_1^T \mathcal{G} \mathcal{A}'_G V'_n &= 0, \\ \mathcal{A}'^T_G \mathcal{G} a_1 + \mathcal{A}'^T_G \mathcal{G} \mathcal{A}'_G V'_n + \mathcal{A}'^T_C \mathcal{C} \mathcal{A}'_C s V'_n &= 0. \end{aligned}$$

Setting

$$l = -\mathcal{A}'^T_G \mathcal{G} a_1, \quad \mathbf{G} = \mathcal{A}'^T_G \mathcal{G} \mathcal{A}'_G, \quad \mathbf{C} = \mathcal{A}'^T_C \mathcal{C} \mathcal{A}'_C, \quad \alpha = a_1^T \mathcal{G} a_1$$

and dropping the prime, i.e., $V_n = V'_n$, the above Laplace-domain equations can be stated as follows:

$$\begin{aligned} I_V &= -\alpha + l^T V_n, \\ l &= \mathbf{G} V_n + \mathbf{C} s V_n. \end{aligned} \tag{33}$$

From (33), we obtain

$$Y(s) = -I_V = \alpha - l^T (\mathbf{G} + s\mathbf{C})^{-1} l. \tag{34}$$

One can now apply the SyPVL algorithm to reduce the second term in (34). Let $\mathbf{G} = \mathbf{L}\mathbf{L}^T$ be the Cholesky factorization of $\mathbf{G}$, where $\mathbf{L}$ denotes the lower triangular Cholesky factor. It follows that

$$\begin{aligned} l^T (\mathbf{G} + s\mathbf{C})^{-1} l &= (\mathbf{L}^{-1} l)^T \left(\mathbf{I} + s \mathbf{L}^{-1} \mathbf{C} \mathbf{L}^{-T} \right)^{-1} \mathbf{L}^{-1} l \\ &= b^T \left(\mathbf{I} - s\mathbf{A} \right)^{-1} b \\ &\approx b^T b \cdot e_1^T \left(\mathbf{I} - s\mathbf{T} \right)^{-1} e_1, \end{aligned} \tag{35}$$

where $b = \mathbf{L}^{-1} l$ and $\mathbf{A} = -\mathbf{L}^{-1} \mathbf{C} \mathbf{L}^{-T}$. We need to bring this reduced expression to a form suitable for easy "unstamping". We start by factoring $-\mathbf{T} = \mathbf{U}\mathbf{U}^T$ into a product of an upper triangular matrix $\mathbf{U}$ and its transpose $\mathbf{U}^T$. To see that such a factorization is possible, we first note that the matrix $-\mathbf{T}$ is symmetric positive definite. Furthermore, it is easy to verify that the matrix $\mathbf{U}$ is just the Cholesky factor resulting from the Cholesky algorithm applied to the matrix $-\mathbf{T}$ with the order of its rows and columns reversed. Using the factorization $-\mathbf{T} = \mathbf{U}\mathbf{U}^T$, we can rewrite the second factor in (35) as follows:

$$e_1^T \left(\mathbf{I} - s\mathbf{T} \right)^{-1} e_1 = (\mathbf{U}^{-1} e_1)^T \left(\mathbf{U}^{-1} \mathbf{U}^{-T} + s\mathbf{I} \right)^{-1} \mathbf{U}^{-1} e_1. \tag{36}$$

Since $\mathbf{U}$ is upper triangular, the vector $\mathbf{U}^{-1} e_1$ in (36) remains a scalar multiple of e_1. We now bring the matrix $\mathbf{U}^{-1}\mathbf{U}^{-T}$ in (36) to tridiagonal form by using an orthogonal similarity transformation based on Givens rotations. Such a transformation is a standard first step in the QR algorithm for computing eigenvalues of symmetric matrices, and is described, e.g., in [8]. Let

$$\mathbf{H} = \mathbf{Q} \left(\mathbf{U}^{-1} \mathbf{U}^{-T} \right) \mathbf{Q}^T \tag{37}$$

be the tridiagonal matrix resulting from this transformation. Note that $\mathbf{Q}$ is a product of Givens rotations, and thus it is orthogonal, i.e., $\mathbf{Q}^{-1} = \mathbf{Q}^T$. Using (37), we can rewrite (36) as follows:

$$(\mathbf{U}^{-1} e_1)^T \left(\mathbf{U}^{-1} \mathbf{U}^{-T} + s\mathbf{I} \right)^{-1} \mathbf{U}^{-1} e_1 = (\mathbf{Q}\mathbf{U}^{-1} e_1)^T \left(\mathbf{H} + s\mathbf{I} \right)^{-1} \mathbf{Q}\mathbf{U}^{-1} e_1. \tag{38}$$

Finally, we need to bring the reduced tridiagonal matrix $\mathbf{H}$ to a form that corresponds to an RC-ladder resistance matrix. This can be done by multiplying $\mathbf{H} = [\, h_{j,k} \,]_{j,k=1,2,\ldots,n}$ from the left and right by a diagonal matrix

$$\mathbf{D} = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_n \end{bmatrix}. \tag{39}$$

The values of d_k, $k = 1, 2, \ldots, n$, are determined by the equations

$$\begin{aligned} & d_{n-1} d_n h_{n-1,n} + d_n^2 h_{n,n} = 0, \\ & d_{k-1} d_k h_{k-1,k} + d_k^2 h_{k,k} + d_k d_{k+1} h_{k,k+1} = 0, \quad k = n-1, n-2, \ldots, 2, \\ & d_1^2 h_{1,1} + d_1 d_2 h_{1,2} = g_0. \end{aligned} \tag{40}$$

This system of equations in matrix form is

$$\begin{bmatrix} h_{1,1} & h_{1,2} & 0 & \cdots & 0 \\ h_{2,1} & h_{2,2} & h_{2,3} & \ddots & \vdots \\ 0 & h_{3,2} & h_{3,3} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & h_{n-1,n} \\ 0 & \cdots & 0 & h_{n,n-1} & h_{n,n} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} g_0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{41}$$

which is just

$$\mathbf{H} d_d = g_0 e_1. \tag{42}$$

The final form is

$$\tilde{Y}(s) = \alpha - b^T b \cdot (\mathbf{DQU}^{-1} e_1)^T \left(\mathbf{DHD} + s\mathbf{D}^2 \right)^{-1} \mathbf{DQU}^{-1} e_1. \tag{43}$$

The values of the resistors (the R's) can be directly read off from the reduced tridiagonal conductance matrix $\mathbf{DHD}$ and the capacitors (the C's) from the reduced capacitance matrix $\mathbf{D}^2$.

5. Examples

In the section, we present two interconnect analysis examples.

The first example represents the extracted RC network model of a data bus consisting of about 5000 resistors and capacitors. The two neighboring lines are also modeled explicitly in this network. SyPVL was applied to synthesize a reduced RC circuit model for the driving-point impedance of this network. Figure 5.2 show the results of the time-domain

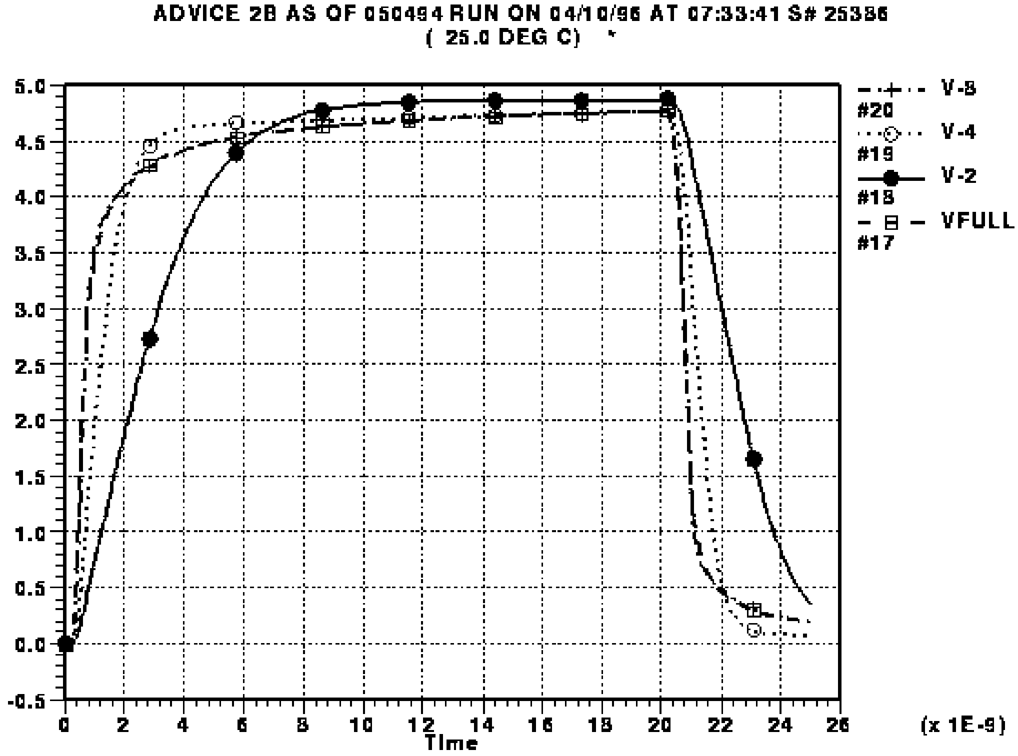

Figure 5.2: Bus line example

simulation of this network driven by a buffer fully modeled at the transistor level. The graphs compare the voltage waveforms at the output of the driver, obtained first by simulation of the detailed, 5000 element RC network, and then by replacing it with RC line models of order 2, 4, and 8. We observe that using a model of order only two causes unacceptably large errors in the results, but as we increase the order of the approximation we regain full accuracy. The use of an RC line of length 8, computed using SyPVL, gives results that are almost indistinguishable from those obtained by simulating the full circuit. Recall that the incremental cost of obtaining a larger order model for a large network is practically negligible.

The second example is a somewhat larger circuit representing a long and intricate interconnection network modeled by over 30,000 resistors and capacitors. Surprisingly, the driving-point impedance of this line can be modeled with near perfect accuracy using a very low order model. Figure 5.3 shows the results of nonlinear, time-domain simulation of a buffer circuit driving this line. There are practically no observable differences in the voltage waveforms at the driving point when the 30,000 element network is replaced by an RC line of length two with parameters determined using SyPVL. One reason for the success of such a low-order model is the fact that coupling interactions with neighboring lines were approximated as capacitances to ground.

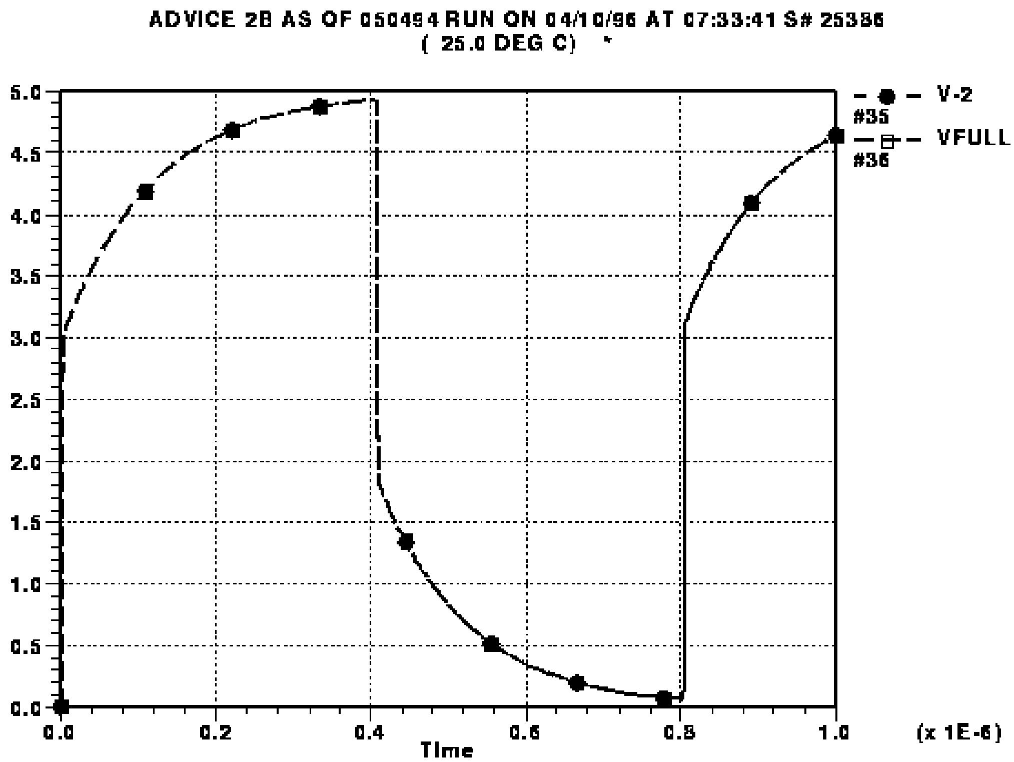

Figure 5.3: Larger interconnect example

6. Concluding remarks

In this paper, we have introduced SyPVL, as the most appropriate tool to compute reduced-order models (based on the Padeápproximation) for large passive linear circuits. The algorithm has the property of generating guaranteed stable approximations for a wide class of circuits of interest. Moreover, the stable approximations can be used to "synthesize" actual circuits that implement the model. We described such a synthesis procedure that can implement the one-port impedance of an arbitrary RC circuit in the form of a ladder structure.

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. 95-11, AT&T Bell Laboratories, Murray Hill, NJ, Nov. 1995.

[2] G. A. Baker, Jr. and P. Graves-Morris, *Padé Approximants, Part I: Basic Theory.* Reading, MA: Addison-Wesley, 1981.

[3] 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.

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

[5] R. W. Freund, "The look-ahead Lanczos process for large nonsymmetric matrices and related algorithms," in *Linear Algebra for Large Scale and Real-Time Applications*, Dordrecht, The Netherlands: Kluwer Academic Publishers, 1993.

[6] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal, "An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices," *SIAM J. Sci. Comput.*, vol. 14, pp. 137–158, Jan. 1993.

[7] R. W. Freund and N. M. Nachtigal, "Software for simplified Lanczos and QMR algorithms," *Appl. Numer. Math.*, vol. 19, pp. 319–341, 1995.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Second Edition. Baltimore, MD: The Johns Hopkins University Press, 1989.

[9] N. Gopal, D. P. Neikirk, and L. T. Pillage, "Evaluating RC-interconnect using moment-matching approximations," *1991 IEEE International Conference on Computer-Aided Design*, Nov. 1991.

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

[11] 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.

[12] 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.

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

[14] C. L. Ratzlaff, N. Gopal, and L. T. Pillage, "RICE: rapid interconnect circuit evaluator," *28th ACM/IEEE Design Automation Conference Proceedings 1991*, June 1991.

[15] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York, N.Y.: Van Nostrand Reinhold, 1983.

[16] M. E. Van Valkenburg, *Introduction to Modern Network Synthesis*. New York, N.Y.: John Wiley & Sons. Inc., 1960.