A partial Padé-via-Lanczos method for reduced-order modeling ⋆

Zhaojun Bai [1]

Department of Computer Science, University of California at Davis, One Shields Avenue, Davis, California 95616, U.S.A.
e-mail: bai@cs.ucdavis.edu

Roland W. Freund

Bell Laboratories, Lucent Technologies, Room 2C–525, 700 Mountain Avenue, Murray Hill, New Jersey 07974–0636, U.S.A.
e-mail: freund@research.bell-labs.com

Abstract

The classical Lanczos process can be used to efficiently generate Padé approximants of the transfer function of a given single-input single-output time-invariant linear dynamical system. Unfortunately, in general, the resulting reduced-order models based on Padé approximation do not preserve the stability, and possibly passivity, of the original linear dynamical system. In this paper, we describe the use of partial Padé approximation for reduced-order modeling. Partial Padé approximants have a number of prescribed poles and zeros, while the remaining degrees of freedom are used to match the Taylor expansion of the original transfer function in as many leading coefficients as possible. We present an algorithm for computing partial Padé approximants via suitable rank-1 updates of the tridiagonal matrices generated by the Lanczos process. Numerical results for two circuit examples are reported.

Key words: Lanczos algorithm; Linear dynamical system; Transfer function; Stability; Passivity; Partial Padé approximation; VLSI circuit simulation

⋆ Numerical Analysis Manuscript No. 99–3–20, Bell Laboratories, Murray Hill, NJ, December 1999. Available online from http://cm.bell-labs.com/cs/doc/99.

[1] Most of this work was done while this author was on sabbatical at Bell Laboratories, Lucent Technologies, Murray Hill, New Jersey.

1 Introduction

It has long been known that Padé approximation is a useful tool for generating reduced-order models of linear dynamical systems; see, e.g., [14] and the references given there. In recent years, there has been renewed interest in and extensive research into Padé-based reduced-order modeling. These recent developments were mainly triggered by the landmark paper [28] that demonstrated the potential of using Padé approximation in the simulation of large electronic VLSI circuits. Some of the recent research in this area focused on the efficient and numerically stable computation of Padé-based reduced-order models. In particular, it is now widely accepted that Krylov-subspace methods, such as the Lanczos algorithm [26] or the Arnoldi process [4] should be employed, in order to avoid the inherent numerical ill-conditioning of generating Padé approximants directly from the Taylor coefficients of the transfer function of the linear dynamical system. We refer the reader to [15,16] for recent surveys of reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation.

On the other hand, it is also well known that, when applied to stable linear dynamical systems, reduced-order modeling techniques based on Padé approximation in general do not preserve the stability of the original system; see, e.g., [1,14,31]. For some applications, such as the use of Padé-based reduced-order models for the efficient computation of the frequency response of large-scale linear dynamical systems, the possible occurrence of unstable poles is not an issue [12]. However, often reduced-order modeling is used to replace large linear subsystems of a stable, possibly nonlinear, system by smaller approximate models, with the goal to reduce the complexity of the simulation of the overall system. In this context, it is crucial that the reduced-order models of the linear subsystems are stable, in order to ensure stability of the simulation of the overall system. In circuit simulation, reduced-order modeling is often applied to large linear subsystems that represent networks consisting of only resistors, inductors, and capacitors. These RLC networks are stable and passive, and again, for the stability of the overall simulation, it is crucial that reduced-order models preserve the passivity of the original RLC network; see, e.g., [29]. Unfortunately, except for the special cases of RC, RL, and LC networks [18,19], Padé-based reduced-order models of RLC networks are not passive in general.

The purpose of this paper is to explore the use of partial Padé approximation for the construction of stable, and possibly passive, reduced-order models. True Padé approximants are rational functions of a given order where all available degrees of freedom are used to match the Taylor expansion of the function to be approximated in as many leading coefficients as possible; see, e.g., [8]. Partial Padé approximants [9], on the other hand, have a number of prescribed poles and zeros, while only the remaining degrees of freedom are used to match the Taylor expansion of the function to be approximated in as many leading coefficients as possible. Our main motivation for studying the use of partial Padé approximation for reduced-order modeling is based on the observation that, typically, the instability, and possibly non-passivity, of reduced-order models based on true Padé approximants is only due to a small number of unstable poles and zeros. By prescribing a small number of stable poles and zeros, while preserving as much of the approximation property of the Padé approximant as possible, we obtain a new reduced-order model based on partial Padé approximation. Often, this new model is stable and possibly passive. In particular, we present an algorithm that generates partial Padé approximants via rank-1 updates of the tridiagonal matrices generated by the Lanczos process. Due to the use of only rank-1 updates, the algorithm is limited to partial Padé approximants whose number, m, of prescribed poles and zeros is bounded by n, where n is the order of the true Padé approximant. However, we stress that, for all practical purposes, this does not pose a limitation at all. First, we never encountered a situation where the true Padé approximant had more than n unstable poles and zeros, which would then require to prescribe $m > n$ poles and zeros. Second, a partial Padé approximant with m prescribed poles and zeros matches the true Padé approximant in its first $2n-m$ Taylor coefficients. So even if $m > n$ would occur, the approximation property of the resulting partial Padé approximant would be too weak to be of practical interest.

We remark that there is some related earlier work; see [1,2,31]. However, the techniques proposed there all involve explicit matching of the Taylor coefficients of the transfer function, and hence they are inherently numerically unstable. In [24], it is proposed to use an implicitly restarted Lanczos method to remedy the possible instability of Padé-based reduced-order models. However, the implicit restarts modify some of the data that describes the given linear dynamical system. Consequently, as pointed out in [15], the reduced-order model generated by this process no longer matches leading Taylor coefficients of the transfer function of the given system, and this is undesirable in some applications, such as circuit simulation, where the leading Taylor coefficients have some physical meaning.

For the special case of RLC networks, it is actually possible to generate provably stable and passive reduced-order models by means of projection onto Krylov subspaces; see [17,27,32]. However, the transfer functions of these projected reduced-order models match only half as many Taylor coefficients of the original transfer function as the corresponding Padé approximant derived from the same Krylov subspace. Moreover, these projection techniques require a very specific formulation of the equations that characterize a given RLC network. For example, in [5], there is an example of a simple RLC network for which the projected reduced-order model is unstable if another formulation of the network equations is used. In contrast, Padé approximation and also partial Padé approximation yield identical results, independent of the chosen formulation of the network equations. Finally, we would like to stress that the techniques described in this paper are not restricted to RLC networks and can be employed for reduced-order modeling of general single-input single-output time-invariant linear dynamical systems.

The remainder of the paper is organized as follows. In Section 2, we collect some facts about transfer functions of single-input single-output time-invariant linear dynamical systems. In Section 3, we briefly review Padé approximants of transfer functions and their computation by means of the Lanczos process. In Sections 4 and 5, we show how partial Padé approximants can be obtained via suitable rank-1 updates of the Lanczos tridiagonal matrix. In Section 6, we present a statement of the overall computational procedure for generating partial Padé approximants via the Lanczos process. In Section 7, we report the results of numerical experiments for two circuit examples. Finally, in Section 8, we make some concluding remarks.

Throughout this article, we use boldface letters to denote vectors and matrices. The $n \times n$ identity matrix is denoted by $\mathbf{I}_n$ and the $n \times m$ zero matrix by $\mathbf{0}_{n\times m}$; if the actual dimensions of these matrices are apparent from the context, we omit these indices and simply write $\mathbf{I}$ and $\mathbf{0}$. For square matrices $\mathbf{M}$, we denote by $\lambda(\mathbf{M})$ the set of all eigenvalues of $\mathbf{M}$. The sets of real and complex numbers are denoted by $\mathbb{R}$ and $\mathbb{C}$, respectively. For $s \in \mathbb{C}$, $\mathrm{Re}(s)$ is the real part of s and $\mathrm{Im}(s)$ is the imaginary part of s. We use $\mathbb{C}_+ := \{\, s \in \mathbb{C} \mid \mathrm{Re}(s) > 0 \,\}$ to denote the open right half of the complex plane. Finally, $\mathcal{R}_{m,n}$ denotes the set of rational functions with real numerator polynomial of degree at most m and real denominator polynomial of degree at most n.

2 Transfer functions and some properties

In this section, we collect some facts about transfer functions of single-input single-output time-invariant linear dynamical systems.

2.1 Time-invariant linear dynamical systems

We consider single-input single-output time-invariant linear dynamical systems given by state-variable descriptions of the form

$$\begin{aligned} \mathbf{C}\,\frac{d}{dt}\mathbf{x}(t) &= -\mathbf{G}\,\mathbf{x}(t) + \mathbf{b}\,u(t), \\ y(t) &= \mathbf{c}^{\mathrm{T}}\,\mathbf{x}(t), \end{aligned} \tag{1}$$

where $\mathbf{C}$, $\mathbf{G} \in \mathbb{R}^{N \times N}$ and $\mathbf{b}$, $\mathbf{c} \in \mathbb{R}^N$. In (1), the function $u(t)$ represents the input of the system, $y(t)$ is the output, and $\mathbf{x}(t)$ is the N-dimensional vector of state variables. The matrices $\mathbf{C}$ and $\mathbf{G}$ are allowed to be singular, and we only assume that the pencil $\mathbf{G} + s\,\mathbf{C}$ is *regular*, i.e., the matrix $\mathbf{G} + s\,\mathbf{C}$ is singular only for finitely many values of $s \in \mathbb{C}$. Note that the first equation in (1) is a system of differential-algebraic equations if $\mathbf{C}$ is singular and a system of ordinary differential equations if $\mathbf{C}$ is nonsingular.

The input-output behavior of the linear dynamical system (1) is described by its Laplace-domain *transfer function*,

$$H : \mathbb{C} \mapsto \mathbb{C} \cup \{\infty\}, \quad H(s) := \mathbf{c}^{\mathrm{T}} \left(\mathbf{G} + s\,\mathbf{C}\right)^{-1} \mathbf{b}, \tag{2}$$

see, e.g., [10]. In Section 3 below, we will consider Padé approximants of transfer functions H of the form (2), and their computation via the Lanczos process. To this end, we will need the following representation of H in terms of a single matrix $\mathbf{A}$, instead of the two matrices $\mathbf{G}$ and $\mathbf{C}$ in the definition of H. Let $s_0 \in \mathbb{R}$ be any fixed expansion point such that the matrix $\mathbf{G} + s_0\,\mathbf{C}$ is nonsingular. Let

$$\mathbf{G} + s_0\,\mathbf{C} = \mathbf{L}\,\mathbf{U}, \quad \text{where} \quad \mathbf{L},\, \mathbf{U} \in \mathbb{R}^{N \times N}, \tag{3}$$

be any formal factorization of $\mathbf{G} + s_0\,\mathbf{C}$. For example, (3) can be chosen as a "true" LU factorization where $\mathbf{L}$ and $\mathbf{U}$ are triangular matrices, possibly permuted due to pivoting, or the "trivial" factorization given by $\mathbf{L} = \mathbf{G} + s_0\,\mathbf{C}$ and $\mathbf{U} = \mathbf{I}$. Using (3), the representation (2) of H can be rewritten as follows:

$$\begin{aligned} &H(s) = \mathbf{l}^{\mathrm{T}} \left(\mathbf{I} - (s - s_0)\,\mathbf{A}\right)^{-1} \mathbf{r}, \\ &\text{where} \quad \mathbf{A} := -\mathbf{L}^{-1}\,\mathbf{C}\,\mathbf{U}^{-1}, \quad \mathbf{r} := \mathbf{L}^{-1}\,\mathbf{b}, \quad \mathbf{l} := \mathbf{U}^{-\mathrm{T}}\,\mathbf{c}. \end{aligned} \tag{4}$$

2.2 Stability

If the linear dynamical system (1) describes an actual physical system, such as a functioning electronic circuit, then it will necessarily be stable. Roughly speaking, stability means that for bounded inputs $u(t)$, the state-variable vector $\mathbf{x}(t)$ of (1) will remain bounded for all times t; see, e.g., [3, Chapter 3.7] or [10, Chapter 8]. For time-invariant linear dynamical systems (1), stability can be defined via the transfer function.

Definition 1 *The transfer function H of a single-input single-output time-invariant linear dynamical system is said to be stable (in the sense of Lya-*

punov) if H has no poles in $\mathbb{C}_+$ and if any pole of H on the imaginary axis is simple.

Note that for functions H given by (4), any pole p_j of H is of the form

$$p_j = s_0 + \frac{1}{\lambda_j}, \quad \text{where} \quad \lambda_j \in \lambda(\mathbf{A}). \tag{5}$$

However, in general, not every p_j of the form (5) is a pole of H. Indeed, the poles of H are given by (5) if, and only if, the triple

$$\mathbf{A} \in \mathbb{R}^{N \times N}, \quad \mathbf{r}, \mathbf{l} \in \mathbb{R}^N \tag{6}$$

in (4) is a minimal realization of H. Here, for a given transfer function H, a representation (4) is called a *minimal realization* if the state-space dimension N in (6) is minimal. If (4) is a minimal realization, then the stability of H can be characterized completely in terms of $\lambda(\mathbf{A})$; see, e.g., [3, Theorem 3.7.2]. Next, we state this result.

Theorem A *Let H be a transfer function given by* (4)*, and assume that* (4) *is a minimal realization. Then* :

a) *The poles of H are given by*

$$p_j = s_0 + \frac{1}{\lambda_j}, \quad \textit{where} \quad \lambda_j \in \lambda(\mathbf{A}), \quad j = 1, 2, \ldots, N.$$

b) *The transfer function H is stable if, and only if, the following two conditions are satisfied* :
  (i) $\mathrm{Re}\,(p_j) \leq 0$ *for all* $j = 1, 2, \ldots, N$;
  (ii) *If* $\mathrm{Re}\,(p_j) = 0$*, then λ_j occurs only in 1×1 blocks in the Jordan canonical form of* $\mathbf{A}$.

2.3 Passivity

Next, we define passivity, which is a stronger condition than stability. Roughly speaking, a system is passive if it does not generate energy. For example, any RLC network is passive. For time-invariant linear dynamical systems (1), passivity is equivalent to positive realness of the associated transfer function H; see, e.g., [3] or [33, Chapter 4]. Based on this equivalence, in this paper, we will use the following definition of passivity.

Definition 2 *The transfer function H of a single-input single-output time-invariant linear dynamical system is said to be passive if*

(i) *H has no poles in $\mathbb{C}_+$;*
(ii) *$H(\overline{s}) = \overline{H(s)}$ for all $s \in \mathbb{C}$;*
(iii) $\mathrm{Re}\,(H(s)) \geq 0$ *for all $s \in \mathbb{C}_+$.*

Note that, for linear dynamical systems (1), condition (ii) is always satisfied since the data $\mathbf{C}$, $\mathbf{G}, \mathbf{b}$, $\mathbf{c}$ in (1) is assumed to be real. Furthermore, using the representation (4) of H, condition (i) can be checked via computing the eigenvalues of the matrix $\mathbf{A}$ in (4). Now assume that (i) is satisfied. Then, by the Maximum Modulus Theorem, condition (iii) is satisfied only if

$$\mathrm{Re}\,(H(\mathtt{i}\,\omega)) \geq 0 \quad \text{for all} \quad \omega \in \mathbb{R}. \tag{7}$$

In [6], it is shown how (7) can be checked via computing the eigenvalues of a certain matrix pencil derived from the representation (4) of H.

In the following theorem, we collect some well-known necessary conditions for passivity.

Theorem B (Necessary conditions for passivity.)

a) *If H is passive, then H is stable.*
b) *If H is passive, then H has no poles and zeros in $\mathbb{C}_+$, and any possible pole or zero of H on the imaginary axis is simple.*

3 Padé approximation via the Lanczos process

In this section, we briefly review the concept of Padé approximation of a given transfer function and the numerical computation of these Padé approximants via the Lanczos process.

3.1 Padé approximants

Let H be the transfer function given by (4). Note that H is a rational function. More precisely, $H \in \mathcal{R}_{N-1,N}$ where N is the state-space dimension of (1). In circuit simulation, N can be extremely large, and then H is replaced by an approximation $H_n \in \mathcal{R}_{n-1,n}$ with state-space dimension $n \ll N$. A widely-used and often the only viable choice of H_n is Padé approximation.

Expanding the transfer function (4) about s_0, we have

$$H(s) = \sum_{j=0}^{\infty} \mu_j (s - s_0)^j, \quad \text{where} \quad \mu_j := \mathbf{l}^{\mathrm{T}}\, \mathbf{A}^j\, \mathbf{r}. \tag{8}$$

A function $H_n \in \mathcal{R}_{n-1,n}$ is said to be an *n-th Padé approximant* of H (about the expansion point s_0) if (8) and the corresponding expansion of H_n agree in the first $2n$ terms, i.e.,

$$H_n(s) = \sum_{j=0}^{2n-1} \mu_j (s - s_0)^j + \mathcal{O}\left((s - s_0)^{2n}\right). \tag{9}$$

For an overview of Padé approximants, we refer the reader to [8].

Note that equation (9) represents $2n$ conditions for the $2n$ degrees of freedom that describe any function $H_n \in \mathcal{R}_{n-1,n}$. In particular, (9) defines a unique n-th Padé approximant H_n if, and only if, the so-called n-th *moment matrix*

$$\mathbf{M}_n := [\, \mu_{j+k-2} \,]_{j,k=1,2,\ldots,n} \quad \text{is nonsingular.} \tag{10}$$

In this paper, for simplicity, we assume that (10) is satisfied for all n.

3.2 Computation via the Lanczos process

The standard approach to computing H_n is to generate the coefficients of the numerator and denominator polynomials of H_n via the solution of systems of linear equations with coefficient matrix $\mathbf{M}_n$. However, in general, due to the typical ill-conditioning of $\mathbf{M}_n$, this approach is feasible only for very moderate values of n, such as $n \le 10$; see [12] for examples. Fortunately, these numerical difficulties can easily be avoided by exploiting the well-known connection [23] between Padé approximants H_n and the Lanczos process [26]. Next, we state this connection.

The starting point is the representation (4) of H. We apply the nonsymmetric Lanczos algorithm to the matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ from (4) and with $\mathbf{r}, \mathbf{l} \in \mathbb{R}^N$ from (4) as right, respectively left, starting vector. We remark that our assumption on the nonsingularity of the moment matrices (10) guarantees that no breakdowns occur in the Lanczos process; see, e.g., [20,23]. After n steps, the Lanczos algorithm has generated right and left Lanczos vectors,

$$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n+1} \quad \text{and} \quad \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{n+1}, \tag{11}$$

such that, for all $j = 1, 2, \ldots, n+1$,

$$\begin{aligned} \operatorname{span}\{\, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_j \,\} &= \operatorname{span}\{\, \mathbf{r}, \mathbf{A}\,\mathbf{r}, \ldots, \mathbf{A}^{j-1}\,\mathbf{r} \,\}, \\ \operatorname{span}\{\, \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_j \,\} &= \operatorname{span}\{\, \mathbf{l}, \mathbf{A}^{\mathrm{T}}\,\mathbf{l}, \ldots, \left(\mathbf{A}^{\mathrm{T}}\right)^{j-1}\,\mathbf{l} \,\}, \end{aligned} \tag{12}$$

and

$$\mathbf{w}_j^{\mathrm{T}}\,\mathbf{v}_k = 0 \quad \text{for all} \quad j \neq k = 1, 2, \ldots, n+1. \tag{13}$$

The conditions (12) and (13) determine the Lanczos vectors only up to a scaling. We use the scaling $\|\mathbf{v}_j\|_2 = \|\mathbf{w}_j\|_2 = 1$ for all j.

The Lanczos vectors are generated by means of three-term recurrences. For the right Lanczos vectors in (11), these recurrences can be stated compactly in matrix form as follows:

$$\mathbf{A}\,\mathbf{V}_n = \mathbf{V}_n\,\mathbf{T}_n + \rho_{n+1}\,\mathbf{v}_{n+1}\,\mathbf{e}_n^{\mathrm{T}}. \tag{14}$$

Here, $\mathbf{V}_n := [\,\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n\,]$, $\mathbf{e}_n$ is the n-th unit vector of length n, and $\mathbf{T}_n$ is the $n \times n$ tridiagonal matrix

$$\mathbf{T}_n = \begin{bmatrix} \alpha_1 & \rho_2 & & \\ \gamma_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \rho_n \\ & & \gamma_n & \alpha_n \end{bmatrix}, \quad \text{where} \quad \rho_j,\, \gamma_j \neq 0 \quad \text{for all} \quad j. \tag{15}$$

The left Lanczos vectors in (11) satisfy an equation similar to (14).

The Padé-Lanczos connection then states that the n-th Padé approximant H_n to the transfer function H in (4) is given by

$$H_n(s) = (\mathbf{l}^{\mathrm{T}}\mathbf{r})\;\mathbf{e}_1^{\mathrm{T}}\left(\mathbf{I} - (s - s_0)\,\mathbf{T}_n\right)^{-1}\mathbf{e}_1, \tag{16}$$

where $\mathbf{e}_1$ denotes the first unit vector of length n. For a proof of (16), we refer the reader to [12] or [23]. Computing H_n via (16) has been advocated in [11,12,21], and following [11,12], this approach is now known as the PVL (Padé via Lanczos) method.

3.3 Poles and zeros of H_n

Using Cramer's rule, the representation (16) of H_n can be rewritten as

$$H_n(s) = (\mathbf{l}^{\mathrm{T}}\mathbf{r})\,\frac{\det\left(\mathbf{I}_{n-1} - (s - s_0)\,\mathbf{T}_n'\right)}{\det\left(\mathbf{I}_n - (s - s_0)\,\mathbf{T}_n\right)}, \tag{17}$$

where $\mathbf{T}_n'$ is the $(n-1) \times (n-1)$ tridiagonal matrix obtained by deleting the first row and first column of the matrix $\mathbf{T}_n$ in (15). The representation (17) shows that the poles and zeros of H_n can be obtained via the eigenvalues of $\mathbf{T}_n$ and $\mathbf{T}'_n$. More precisely, the poles are given by $p_j = s_0 + 1/\lambda_j$, $\lambda_j \in \lambda(\mathbf{T}_n)$, and the zeros by $z_j = s_0 + 1/\lambda'_j$, $\lambda'_j \in \lambda(\mathbf{T}'_n)$. Note that common poles and zeros, $p_j = z_j$, which then would cancel out, cannot occur in (17). This is a consequence of the fact that all sub- and superdiagonal elements of $\mathbf{T}_n$ in (15) are nonzero. We thus have the following result.

Lemma 1 *Let $1 \le n \le N$ and H_n be the n-th Padé approximant of H. Then, the poles p_j and the zeros z_j of H_n are given by*

$$\begin{aligned} p_j &= s_0 + \frac{1}{\lambda_j}, \quad \textit{where} \quad \lambda_j \in \lambda(\mathbf{T}_n), \\ \textit{and} \quad z_j &= s_0 + \frac{1}{\lambda'_j}, \quad \textit{where} \quad \lambda'_j \in \lambda(\mathbf{T}'_n). \end{aligned}$$

Remark 1 The Lanczos process is intimately connected with formally orthogonal polynomials; see, e.g., [20,23] and the references given there. More precisely, each pair of right and left Lanczos vectors (11) can be expressed as

$$\mathbf{v}_j = \xi_j \, \psi_{j-1}(\mathbf{A})\, \mathbf{r} \quad \text{and} \quad \mathbf{w}_j = \eta_j \, \psi_{j-1}(\mathbf{A}^{\mathrm{T}})\, \mathbf{l}, \tag{18}$$

where ψ_{j-1} is a monic polynomial of degree $j-1$ and $\xi_j, \eta_j \neq 0$ are suitable scaling factors. In view of (18), the bi-orthogonality (13) of the Lanczos vectors is equivalent to the formal orthogonality,

$$\langle \psi_j, \psi_k \rangle := \mathbf{l}^{\mathrm{T}} \, \psi_j(\mathbf{A})\, \psi_k(\mathbf{A})\, \mathbf{r} = 0 \quad \text{for all} \quad j \neq k = 0, 1, \ldots, n,$$

of the polynomials $\psi_0, \psi_1, \ldots, \psi_n$. It is well known [23] that ψ_n is the characteristic polynomial of $\mathbf{T}_n$. Thus, the denominator polynomial in the representation (17) of H_n is the *reverse* of ψ_n with respect to s_0, i.e.,

$$\det(\mathbf{I}_n - (s - s_0)\, \mathbf{T}_n) = (s - s_0)^n \, \psi_n(1/(s - s_0)).$$

Similarly, the numerator polynomial in (17) is the reverse of the characteristic polynomial of $\mathbf{T}'_n$ with respect to s_0.

Remark 2 Using the connection with polynomials outlined in Remark 1, it is possible to relate our results on partial Padé approximants to the work [22,25] on the calculation of Gauss quadratures with some prescribed knots.

4 Partial Padé approximation

In this section, we first define partial Padé approximants and then establish a connection to the Lanczos process.

In the following, let k, ℓ, and n be given integers with

$$0 \le k \le n, \quad 0 \le \ell \le n-1, \quad 1 \le m := k + \ell \le n, \tag{19}$$

and let

$$0 \ne \varphi_1, \varphi_2, \dots, \varphi_k, \theta_1, \theta_2, \dots, \theta_\ell \in \mathbb{C}, \tag{20}$$

be given complex numbers. We assume that the m numbers (20) are pairwise distinct and that each of the sets

$$\mathcal{P} := \{\, \varphi_1, \varphi_2, \dots, \varphi_k \,\} \quad \text{and} \quad \mathcal{D} := \{\, \theta_1, \theta_2, \dots, \theta_\ell \,\} \tag{21}$$

is closed under complex conjugation:

$$\varphi_j \in \mathcal{P} \;\Rightarrow\; \overline{\varphi_j} \in \mathcal{P} \quad \text{and} \quad \theta_j \in \mathcal{D} \;\Rightarrow\; \overline{\theta_j} \in \mathcal{D}. \tag{22}$$

We stress that, in (19), either $k = 0$ or $\ell = 0$ is allowed, which means that either $\mathcal{P}$ or $\mathcal{D}$ can be the empty set. In analogy to the representations of the poles and zeros of H_n given in Lemma 1, we use the numbers (20) to define prescribed poles and zeros

$$\begin{aligned} \hat{p}_j &:= s_0 + \frac{1}{\varphi_j}, \quad j = 1, 2, \dots, k, \\ \text{and} \quad \hat{z}_j &:= s_0 + \frac{1}{\theta_j}, \quad j = 1, 2, \dots, \ell, \end{aligned} \tag{23}$$

respectively. Now consider approximants of the form

$$\hat{H}_n \in \mathcal{R}_{n-1,n} \quad \text{with prescribed poles } \hat{p}_1, \dots, \hat{p}_k \text{ and zeros } \hat{z}_1, \dots, \hat{z}_\ell. \tag{24}$$

The functions (24) have $2n - k - \ell = 2n - m$ degrees of freedom, and therefore, for a given expansion point $s_0 \in \mathbb{R}$, one would expect that such functions can be used to match the first $2n - m$ coefficients in the expansion (8) of the original transfer function H. This leads to the following definition. A function $\hat{H}_n$ of the form (24) is called an *n-th partial Padé approximant* of H (about the expansion point s_0) if (8) and the corresponding expansion of $\hat{H}_n$ agree in the first $2n-m$ terms, i.e.,

$$\hat{H}_n(s) = \sum_{j=0}^{2n-m-1} \mu_j (s-s_0)^j + \mathcal{O}\left((s-s_0)^{2n-m}\right). \tag{25}$$

We remark that the general concept of partial Padé approximation (not only of transfer functions) was introduced and studied by Brezinski [9]. The special case that only poles are prescribed, i.e., $\ell = 0$ is usually referred to as Padé-type approximation; see, e.g., [8].

In [9], it is shown how to generate partial Padé approximants directly from the coefficients $\mu_0, \mu_1, \ldots, \mu_{2n-m-1}$ in the expansion (8) of H. However, as in the case of Padé approximants H_n, this approach suffers from inherent ill-conditioning. Next, we propose a different approach for constructing the partial Padé approximant $\hat{H}_n$ via the Lanczos process. We stress that, in principle, it is also possible to construct $\hat{H}_n$ using any pair of bases for the Krylov subspaces (12). However, since for the computation of H_n, the Lanczos vectors appear to be the bases of choice, we see no reason to employ any other bases for the construction of $\hat{H}_n$.

Let $\mathbf{T}_n$ be the $n \times n$ tridiagonal Lanczos matrix. Recall from (16) that $\mathbf{T}_n$ defines the n-th Padé approximant H_n. Let m be the integer defined in (19). We consider rank-1 updates of $\mathbf{T}_n$ of the form

$$\widehat{\mathbf{T}}_n = \mathbf{T}_n + \mathbf{z}\,\mathbf{e}_n^{\mathrm{T}}, \quad \text{where} \quad \mathbf{z} = \begin{bmatrix} \mathbf{0} \\ \mathbf{z}_m \end{bmatrix} \quad \text{and} \quad \mathbf{z}_m \in \mathbb{R}^m. \tag{26}$$

Here, $\mathbf{e}_n$ denotes the n-th unit vector of length n. Therefore, (26) means that $\widehat{\mathbf{T}}_n$ and the tridiagonal matrix $\mathbf{T}_n$ differ only in the trailing m entries of their last columns. For example, when $n = 8$ and $m = 5$, $\widehat{\mathbf{T}}_8$ is a matrix of the form

$$\widehat{\mathbf{T}}_8 = \begin{bmatrix} \times & \times & & & & & & \\ \times & \times & \times & & & & & \\ & \times & \times & \times & & & & \\ & & \times & \times & \times & & & \times \\ & & & \times & \times & \times & & \times \\ & & & & \times & \times & \times & \times \\ & & & & & \times & \times & \times \\ & & & & & & \times & \times \end{bmatrix}.$$

In analogy to (16), we now set

$$\hat{H}_n(s) := (\mathbf{l}^{\mathrm{T}}\mathbf{r})\, \mathbf{e}_1^{\mathrm{T}} \left(\mathbf{I} - (s-s_0)\,\widehat{\mathbf{T}}_n\right)^{-1} \mathbf{e}_1. \tag{27}$$

The following lemma shows that the function (27) is a candidate for the n-th partial Padé approximant.

Lemma 2 *For any choice of the vector* $\mathbf{z}_m \in \mathbb{R}^m$ *in* (26)*, the associated function* $\hat{H}_n$ *defined in* (27) *satisfies*

$$\hat{H}_n(s) = H_n(s) + \mathcal{O}\left((s - s_0)^{2n-m}\right). \tag{28}$$

PROOF. Expanding both $\hat{H}_n$ in (27) and H_n in (16) about s_0, we see that (28) holds true if

$$\mathbf{e}_1^{\mathrm{T}}\, \widehat{\mathbf{T}}_n^j\, \mathbf{e}_1 = \mathbf{e}_1^{\mathrm{T}}\, \mathbf{T}_n^j\, \mathbf{e}_1 \quad \text{for all} \quad j = 0, 1, \ldots, 2n - m - 1. \tag{29}$$

Thus, it only remains to show (29). To this end, note that, since $\mathbf{T}_n$ is tridiagonal and $\mathbf{e}_1$ is the first unit vector, for all $i = 1, 2, \ldots, n-1$, the trailing $n - i$ entries of the column vector $\mathbf{T}_n^{i-1}\, \mathbf{e}_1$ are guaranteed to be zero. Together with (26), we get

$$\widehat{\mathbf{T}}_n\, \mathbf{T}_n^{i-1}\, \mathbf{e}_1 = \mathbf{T}_n^i\, \mathbf{e}_1 + \mathbf{z}\left(\mathbf{e}_n^{\mathrm{T}}\, \mathbf{T}_n^{i-1}\, \mathbf{e}_1\right) = \mathbf{T}_n^i\, \mathbf{e}_1, \quad 1 \le i \le n - 1. \tag{30}$$

Using induction on i, we deduce from (30) that

$$\widehat{\mathbf{T}}_n^i\, \mathbf{e}_1 = \mathbf{T}_n^i\, \mathbf{e}_1 \quad \text{for all} \quad i = 0, 1, \ldots, n - 1. \tag{31}$$

Similarly, for all $j = 1, 2, \ldots, n-m$, the trailing $n-j$ entries of the row vector $\mathbf{e}_1^{\mathrm{T}}\, \mathbf{T}_n^{j-1}$ are guaranteed to be zero, and together with (26), we get

$$\mathbf{e}_1^{\mathrm{T}}\, \mathbf{T}_n^{j-1}\, \widehat{\mathbf{T}}_n = \mathbf{e}_1^{\mathrm{T}}\, \mathbf{T}_n^j + \left(\mathbf{e}_1^{\mathrm{T}}\, \mathbf{T}_n^{j-1}\, \mathbf{z}\right)\, \mathbf{e}_n^{\mathrm{T}} = \mathbf{e}_1^{\mathrm{T}}\, \mathbf{T}_n^j, \quad 1 \le j \le n - m. \tag{32}$$

Using induction on j, we deduce from (32) that

$$\mathbf{e}_1^{\mathrm{T}}\, \widehat{\mathbf{T}}_n^j = \mathbf{e}_1^{\mathrm{T}}\, \mathbf{T}_n^j \quad \text{for all} \quad j = 0, 1, \ldots, n - m. \tag{33}$$

By multiplying (33) and (31), it follows that

$$\mathbf{e}_1^{\mathrm{T}}\, \widehat{\mathbf{T}}_n^{j+i}\, \mathbf{e}_1 = \left(\mathbf{e}_1^{\mathrm{T}}\, \widehat{\mathbf{T}}_n^j\right)\left(\widehat{\mathbf{T}}_n^i\, \mathbf{e}_1\right) = \left(\mathbf{e}_1^{\mathrm{T}}\, \mathbf{T}_n^j\right)\left(\mathbf{T}_n^i\, \mathbf{e}_1\right) = \mathbf{e}_1^{\mathrm{T}}\, \mathbf{T}_n^{j+i}\, \mathbf{e}_1$$

for all $j + i = 0, 1, \ldots, n - m + n - 1 = 2n - m - 1$, which is just the claim (29). $\square$

In view of (9), (25), and (28), the function $\hat{H}_n$ defined in (27) is indeed an n-th partial Padé approximant if it has the prescribed poles and zeros (23). Note that, in analogy to (17), the function (27) has the representation

$$\hat{H}_n(s) = (\mathbf{l}^{\mathrm{T}}\mathbf{r})\, \frac{\det(\mathbf{I}_{n-1} - (s - s_0)\, \widehat{\mathbf{T}}'_n)}{\det(\mathbf{I}_n - (s - s_0)\, \widehat{\mathbf{T}}_n)}, \tag{34}$$

where $\widehat{\mathbf{T}}'_n$ is the $(n-1) \times (n-1)$ matrix obtained by deleting the first row and the first column of $\widehat{\mathbf{T}}_n$. By (34), $\hat{H}_n$ has the prescribed poles and zeros (23) if, and only if, the matrices $\widehat{\mathbf{T}}_n$ and $\widehat{\mathbf{T}}'_n$ have the prescribed eigenvalues $\mathcal{P}$ and $\mathcal{D}$, respectively. Therefore, we have established the following theorem.

Theorem 3 *Let $\mathbf{T}_n$ be the $n \times n$ Lanczos tridiagonal matrix and $\widehat{\mathbf{T}}_n$ be its rank-1 update (26). If the vector $\mathbf{z}_m \in \mathbb{R}^m$ in (26) is chosen such that*

$$\mathcal{P} \subset \lambda\big(\widehat{\mathbf{T}}_n\big) \quad \textit{and} \quad \mathcal{D} \subset \lambda\big(\widehat{\mathbf{T}}'_n\big), \tag{35}$$

then the function $\hat{H}_n$ in (27) is an n-th partial Padé approximant of H.

By Theorem 3, computing an n-th partial Padé approximant via the Lanczos process reduces to constructing the vector $\mathbf{z}_m$ in (26) such that (35) is satisfied. In the next section, we show that such a vector can be obtained by solving a suitable linear system.

5 Computing the vector $\mathbf{z}_m$

Let $\mathcal{P}$ and $\mathcal{D}$ be the sets (21) of prescribed eigenvalues of $\widehat{\mathbf{T}}_n$ and $\widehat{\mathbf{T}}'_n$, respectively.

Clearly, $\varphi_j \in \mathcal{P}$ is an eigenvalue of $\widehat{\mathbf{T}}_n$ if, and only if, there is an associated left eigenvector $\mathbf{q}_j$, i.e.,

$$\mathbf{q}_j\, \widehat{\mathbf{T}}_n = \varphi_j\, \mathbf{q}_j, \quad \mathbf{q}_j \in \mathbb{C}^{1\times n}, \quad \mathbf{q}_j \neq \mathbf{0}. \tag{36}$$

Using (26), we can re-write (36) as follows:

$$\varphi_j\, \mathbf{q}_j - \mathbf{q}_j\, \mathbf{T}_n = (\mathbf{q}_j\, \mathbf{z})\; \mathbf{e}_n^{\mathrm{T}}, \quad \mathbf{q}_j \neq \mathbf{0}. \tag{37}$$

Therefore, $\widehat{\mathbf{T}}_n$ has the prescribed eigenvalues $\mathcal{P}$ if, and only if, the column vector $\mathbf{z}$ in (26) can be chosen such that, for each $j = 1, 2, \ldots, k$, there exists a row vector $\mathbf{q}_j$ satisfying (37). The next proposition gives a condition for such a vector $\mathbf{z}$.

Proposition 4 *For each $j = 1, 2, \ldots, k$, let $\mathbf{q}_j$ be a solution of*

$$\mathbf{q}_j \, \mathbf{T}_n = \varphi_j \, \mathbf{q}_j, \quad \mathbf{q}_j \in \mathbb{C}^{1 \times n}, \quad \|\mathbf{q}_j\|_2 = 1, \tag{38}$$

if $\varphi_j \in \lambda(\mathbf{T}_n)$, and set

$$\mathbf{q}_j \ := (\varphi_j \, \mathbf{I} - \mathbf{T}_n)^{-1} \, \mathbf{e}_n^{\mathrm{T}} \tag{39}$$

otherwise. Moreover, for $j = 1, 2, \ldots, k$, set

$$d_j := \begin{cases} 0 & \text{if } \varphi_j \in \lambda(\mathbf{T}_n), \\ 1 & \text{otherwise.} \end{cases} \tag{40}$$

Then, $\widehat{\mathbf{T}}_n$ has the prescribed eigenvalues $\mathcal{P}$ if, and only if, the vector $\mathbf{z}$ in (26) satisfies the system of linear equations

$$\mathbf{Q}_k \, \mathbf{z} = \mathbf{d}_k, \quad \text{where} \quad \mathbf{Q}_k := \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_k \end{bmatrix} \quad \text{and} \quad \mathbf{d}_k := \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{bmatrix}. \tag{41}$$

PROOF. Let $\mathbf{q}_j$ be any vector satisfying (37). First, we show that

$$\mathbf{q}_j \, \mathbf{z} = 0 \quad \Longleftrightarrow \quad \varphi_j \in \lambda(\mathbf{T}_n). \tag{42}$$

Indeed, if $\mathbf{q}_j \, \mathbf{z} = 0$, then (37) reduces to $\mathbf{q}_j \, \mathbf{T}_n = \varphi_j \, \mathbf{q}_j$, $\mathbf{q}_j \neq \mathbf{0}$, and thus $\varphi_j \in \lambda(\mathbf{T}_n)$ with associated left eigenvector $\mathbf{q}_j$. Conversely, let $\varphi_j \in \lambda(\mathbf{T}_n)$ and assume that $\mathbf{q}_j \, \mathbf{z} \neq 0$. By (37), it follows that

$$\left(\varphi_j \, \mathbf{I} - \mathbf{T}_n^{\mathrm{T}}\right) \mathbf{q}_j^{\mathrm{T}} = d_j \, \mathbf{e}_n, \quad \text{where} \quad d_j := \mathbf{q}_j \, \mathbf{z} \neq 0,$$

and thus

$$\mathbf{e}_n \in \operatorname{range}\left(\varphi_j \, \mathbf{I} - \mathbf{T}_n^{\mathrm{T}}\right). \tag{43}$$

On the other hand, by (15), $\varphi_j \, \mathbf{I} - \mathbf{T}_n^{\mathrm{T}}$ is an unreduced lower Hessenberg matrix, which is also singular since $\varphi_j \in \lambda(\mathbf{T}_n)$. This implies

$$\operatorname{rank}\left(\varphi_j \, \mathbf{I} - \mathbf{T}_n^{\mathrm{T}}\right) = n - 1 \quad \text{and} \quad \operatorname{rank}\left[\left(\varphi_j \, \mathbf{I} - \mathbf{T}_n^{\mathrm{T}}\right) \quad \mathbf{e}_n\right] = n,$$

which is a contradiction to (43). Thus $\mathbf{q}_j\,\mathbf{z} = 0$ and the proof of (42) is complete.

Next, note that, in view of (15) and (26), $\widehat{\mathbf{T}}_n$ is an unreduced upper Hessenberg matrix. This implies that all eigenvalues of $\widehat{\mathbf{T}}_n$ have geometric multiplicity 1, and thus the left eigenvector $\mathbf{q}_j$ is uniquely determined by (36) up to a nonzero normalization factor. By (42), we can fix the normalization of $\mathbf{q}_j$ such that

$$\mathbf{q}_j\,\mathbf{z} = d_j := \begin{cases} 0 & \text{if } \varphi_j \in \lambda(\mathbf{T}_n), \\ 1 & \text{otherwise.} \end{cases} \tag{44}$$

With the normalization (44), it follows that a solution $\mathbf{q}_j$ of (37) is given by (38) if $\varphi_j \in \lambda(\mathbf{T}_n)$ and by (39) otherwise. Finally, with the vectors $\mathbf{q}_j$, $j = 1, 2, \ldots, k$, given by (38) or (39), the matrix $\widehat{\mathbf{T}}_n$ has indeed the prescribed eigenvalues if, and only if, the vector $\mathbf{z}$ satisfies the normalization conditions (44). However, (44) is just the linear system (41). □

For the prescribed eigenvalues $\mathcal{D}$ of $\lambda(\mathbf{T}'_n)$, we can proceed in analogous fashion. This leads to the following proposition.

Proposition 5 *For each $j = 1, 2, \ldots, \ell$, let $\mathbf{r}_j$ be a solution of*

$$\mathbf{r}_j\,\mathbf{T}'_n = \theta_j\,\mathbf{r}_j, \quad \mathbf{r}_j \in \mathbb{C}^{1\times n-1}, \quad \|\mathbf{r}_j\|_2 = 1, \tag{45}$$

if $\theta_j \in \lambda(\mathbf{T}'_n)$, and set

$$\mathbf{r}_j \;:= (\theta_j\,\mathbf{I} - \mathbf{T}'_n)^{-1}\,\mathbf{e}_{n-1}^{\mathrm{T}} \tag{46}$$

otherwise. Moreover, for $j = 1, 2, \ldots, \ell$, set

$$f_j := \begin{cases} 0 & \textit{if } \theta_j \in \lambda(\mathbf{T}'_n), \\ 1 & \textit{otherwise.} \end{cases} \tag{47}$$

Then, $\widehat{\mathbf{T}}'_n$ has the prescribed eigenvalues $\mathcal{D}$ if, and only if, the vector $\mathbf{z}$ in (26) *satisfies the system of linear equations*

$$\mathbf{R}_\ell\,\mathbf{z} = \mathbf{f}_\ell, \quad \textit{where} \quad \mathbf{R}_\ell := \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_\ell \end{bmatrix} \quad \textit{and} \quad \mathbf{f}_\ell := \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_\ell \end{bmatrix}. \tag{48}$$

The proof of Proposition 5 is completely analogous to that of Proposition 4 and can thus be omitted.

By combining the linear systems (41) and (48), we obtain $m = k + \ell$ equations for the vector $\mathbf{z}$. Furthermore, recall from (26) that the first $n - m$ entries of $\mathbf{z}$ are zero and that only the trailing part $\mathbf{z}_m$ needs to be computed. For the case $m \leq n - 1$, the resulting linear system for $\mathbf{z}_m$ is as follows:

$$\begin{bmatrix} \mathbf{Q}_k(:, n-m+1:n) \\ \mathbf{R}_\ell(:, n-m:n-1) \end{bmatrix} \mathbf{z}_m = \begin{bmatrix} \mathbf{d}_k \\ \mathbf{f}_\ell \end{bmatrix}. \tag{49}$$

Here, $\mathbf{M}(:, i:j)$ denotes the submatrix consisting of the i-th column to j-th column of the matrix $\mathbf{M}$. For the case $m = n$, the linear system for $\mathbf{z}_m$ is as follows:

$$\begin{bmatrix} & \mathbf{Q}_k & \\ \mathbf{0}_{\ell \times 1} & & \mathbf{R}_\ell \end{bmatrix} \mathbf{z}_m = \begin{bmatrix} \mathbf{d}_k \\ \mathbf{f}_\ell \end{bmatrix}. \tag{50}$$

Note that both (49) and (50) are systems of m linear equations for the m entries of $\mathbf{z}_m$. Therefore, both these linear systems have unique solutions, provided their coefficient matrices are nonsingular.

We remark that, although the coefficient matrices of (49) and (50) are complex in general, the solution vector $\mathbf{z}_m$ is always real. This follows from assumption (22) on the sets $\mathcal{P}$ and $\mathcal{D}$. Indeed, let $\varphi_j \in \mathcal{P}$ and $\varphi_{j+1} = \overline{\varphi_j} \in \mathcal{P}$ be a pair of complex conjugate prescribed eigenvalues of $\widehat{\mathbf{T}}_n$. It then follows from (38)–(40) that

$$\mathbf{q}_{j+1} = \overline{\mathbf{q}_j} \quad \text{and} \quad d_{j+1} = d_j. \tag{51}$$

Multiplying the j-th and $j+1$-st row of (41) from the left by

$$\frac{1}{2} \begin{bmatrix} 1 & 1 \\ -\mathtt{i} & \mathtt{i} \end{bmatrix}$$

and using (51), we obtain the equivalent real equations

$$\begin{bmatrix} \mathrm{Re}(\mathbf{q}_j) \\ \mathrm{Im}(\mathbf{q}_j) \end{bmatrix} \mathbf{z} = \begin{bmatrix} d_j \\ 0 \end{bmatrix}.$$

This means that, in (41) and thus also in (49), respectively (50), we can replace the pair of rows corresponding to each pair of complex conjugate numbers in $\mathcal{P}$ by an equivalent real pair of rows. Similarly, one can turn the system (48) into a real system. All together, this means that the system (49), respectively (50), can always be made real, and thus the solution vector $\mathbf{z}_m$ is real.

In summary, we have the following algorithm for computing the vector $\mathbf{z}$ in (26).

Algorithm 1 (Computing the vector $\mathbf{z}$)

INPUT: *n-th Lanczos tridiagonal matrix $\mathbf{T}_n$,*
sets $\mathcal{P}$ and $\mathcal{D}$ satisfying (19)–(22)*.*

OUTPUT: *A vector $\mathbf{z} \in \mathbb{R}^n$ such that the matrices $\widehat{\mathbf{T}}_n = \mathbf{T}_n + \mathbf{z}\,\mathbf{e}_n^{\mathrm{T}}$ and $\widehat{\mathbf{T}}'_n$ have the prescribed eigenvalues $\mathcal{P}$ and $\mathcal{D}$, respectively.*

1) *For $j = 1, 2, \ldots, k$ do:*
   If $\varphi_j \in \lambda(\mathbf{T}_n)$,
      compute $\mathbf{q}_j$ as a solution of (38)*,*
   else
      compute $\mathbf{q}_j$ by (39)*.*
2) *For $j = 1, 2, \ldots, \ell$ do:*
   If $\theta_j \in \lambda(\mathbf{T}'_n)$,
      compute $\mathbf{r}_j$ as a solution of (45)*,*
   else
      compute $\mathbf{r}_j$ by (46)*.*
3) *Compute $\mathbf{d}_k$ by* (40) *and* (41)*.*
4) *Compute $\mathbf{f}_\ell$ by* (47) *and* (48)*.*
5) *If $m = k + \ell < n$ do:*
   If the coefficient matrix of (49) *is singular: stop.*
   Otherwise, solve (49) *for the vector $\mathbf{z}_m$ and set $\mathbf{z} = \begin{bmatrix} \mathbf{0} \\ \mathbf{z}_m \end{bmatrix}$.*
   If $m = k + \ell = n$ do:
   If the coefficient matrix of (50) *is singular: stop.*
   Otherwise, solve (50) *for the vector $\mathbf{z}_m$ and set $\mathbf{z} = \mathbf{z}_m$.*

We conclude this section with remarks about the two special cases of partial Padé approximants with all poles, respectively all zeros, prescribed.

Remark 3 Consider the case that all poles $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n$ of the partial Padé approximant $\hat{H}_n$ are prescribed. In this case, $k = n$ and $\ell = 0$ in (19), and the linear system (50) reduces to $\mathbf{Q}_n\,\mathbf{z} = \mathbf{d}_n$. In view of Definition 1, this special case can always be used to generate a reduced-order transfer function $\hat{H}_n$ that is guaranteed to be stable. Indeed, all one needs to do is prescribe pairs of complex conjugate poles with $\mathrm{Re}(\hat{p}_j) < 0$ for all $j = 1, 2, \ldots, n$. On the other hand, the associated n-th partial Padé approximant $\hat{H}_n$ then only matches $2n - n = n$ Taylor coefficients of the original transfer function H, instead of the $2n$ coefficients matched by the n-th Padé approximant H_n. For example, suppose $n - 2$ of the poles $p_j = s_0 + 1/\lambda_j$ of H_n are pairwise distinct and satisfy $\mathrm{Re}(p_j) < 0$, while the other 2 poles, say p_{n-1} and p_n, violate the stability of H_n. Then, we can choose the elements of the set $\mathcal{P}$ in (21) as follows:

$$\varphi_j = \begin{cases} \lambda_j & \text{for } j = 1, 2, \ldots, n-2, \\ \dfrac{1}{\hat{p}_j - s_0} & \text{for } j = n-1, n. \end{cases}$$

Here, $\hat{p}_{n-1}$ and $\hat{p}_n$ is any pair of prescribed real or complex conjugate poles with strictly negative real part. For this choice of $\mathcal{P}$, Algorithm 1 generates a vector $\mathbf{z}$ such that the associated reduced-order transfer function (27), $\hat{H}_n$, is guaranteed to be stable.

Remark 4 If $k = 0$ and $\ell = n - 1$, then all the zeros of $\hat{H}_n$ are prescribed. In this case, the linear system (49) reduces to $\mathbf{R}_{n-1}\,\mathbf{z}_{n-1} = \mathbf{f}_{n-1}$, and the associated function (27), $\hat{H}_n$, matches $2n - (n-1) = n+1$ Taylor coefficients of the original transfer function H.

6 PVLπ algorithm

In this section, we combine the PVL method sketched in Section 3.2 with the update procedure for obtaining partial Padé approximants described in Sections 4 and 5. The resulting computational procedure is called the PVLπ algorithm. It consists of the basic PVL algorithm to generate the true Padé approximant H_n and additional post-processing, with the goal to remove unstable poles, and possibly unstable zeros, of H_n.

A sketch of the PVLπ algorithm is as follows.

Algorithm 2 (Sketch of the PVLπ algorithm)

INPUT: *Expansion point $s_0 \in \mathbb{R}$,*
data $\mathbf{A}$, $\mathbf{r}$, $\mathbf{l}$ of the function $H(s) = \mathbf{l}^{\mathrm{T}} \left(\mathbf{I} - (s - s_0)\,\mathbf{A}\right)^{-1} \mathbf{r}$.

OUTPUT: *An n-th Padé approximant H_n or partial Padé approximant $\hat{H}_n$ of H.*

1) *Run n steps of the Lanczos process (applied to the matrix $\mathbf{A}$ with right and left starting vectors $\mathbf{r}$ and $\mathbf{l}$) to obtain the tridiagonal Lanczos matrix $\mathbf{T}_n$, and set*

$$H_n(s) = (\mathbf{l}^{\mathrm{T}}\mathbf{r})\,\mathbf{e}_1^{\mathrm{T}} \left(\mathbf{I} - (s - s_0)\,\mathbf{T}_n\right)^{-1} \mathbf{e}_1. \tag{52}$$

Here, n is chosen such that some appropriate stopping criterion is satisfied.

2) *Compute the eigenvalues* $\lambda(\mathbf{T}_n)$ *and* $\lambda(\mathbf{T}'_n)$*, and from these the poles and zeros*

$$p_j = s_0 + \frac{1}{\lambda_j}, \quad \lambda_j \in \lambda(\mathbf{T}_n), \quad \textit{and} \quad z_j = s_0 + \frac{1}{\lambda'_j}, \quad \lambda'_j \in \lambda(\mathbf{T}'_n),$$

of H_n.

3) *Check the stability, and possibly passivity, of* H_n.
   If H_n *is stable, and possibly passive, then*: *stop.*
4) *Choose* $0 \le k \le n$ *and* $0 \le \ell \le n-1$ *with* $k + \ell \le n$. *Prescribe* k *poles* $\{\hat{p}_j\}_{j=1}^{k}$ *and* ℓ *zeros* $\{\hat{z}_j\}_{j=1}^{\ell}$ *such that the numbers*

$$\varphi_j := \frac{1}{\hat{p}_j - s_0}, \quad 1 \le j \le k, \quad \textit{and} \quad \theta_j := \frac{1}{\hat{z}_j - s_0}, \quad 1 \le j \le \ell,$$

satisfy (20)–(22).

5) *Use Algorithm* 1 *to compute the vector* $\mathbf{z}$.
6) *Set* $\widehat{\mathbf{T}}_n = \mathbf{T}_n + \mathbf{z}\,\mathbf{e}_n^{\mathrm{T}}$ *and*

$$\hat{H}_n(s) = (\mathbf{l}^{\mathrm{T}}\mathbf{r})\,\mathbf{e}_1^{\mathrm{T}}\left(\mathbf{I} - (s - s_0)\,\widehat{\mathbf{T}}_n\right)^{-1}\mathbf{e}_1. \tag{53}$$

7) *Compute the poles, and possibly zeros, of* $\hat{H}_n$ *and check the stability, and possibly passivity, of* $\hat{H}_n$.
   If $\hat{H}_n$ *is stable, and possibly passive, then*: *stop.*
   Otherwise, return to Step 4) *and choose another set of prescribed poles and zeros.*

Next, we make some remarks on the various steps of Algorithm 2.

Remark 5 The PVL algorithm [11,12] essentially consists of Steps 1) and 2).

Remark 6 In Step 1), a simple stopping criterion is to check the convergence of the dominant poles of H_n, as suggested in [12]. A more sophisticated stopping criterion based on directly estimating the error $|H(s) - H_n(s)|$ was recently proposed in [7].

Remark 7 To test for stability in Steps 3) and 7), one just needs to check if all poles of H_n and $\hat{H}_n$, respectively, satisfy the conditions given in part b) of Theorem A.

Remark 8 If the given function H is passive, then in Steps 3) and 7) one also needs to check the passivity of H_n and $\hat{H}_n$, respectively. The necessary conditions for passivity given in Theorem B only involve the poles and zeros, and thus they can easily be checked once the eigenvalues $\lambda(\mathbf{T}_n)$ and $\lambda(\mathbf{T}'_n)$, respectively $\lambda(\widehat{\mathbf{T}}_n)$ and $\lambda(\widehat{\mathbf{T}}'_n)$, have been computed. If these necessary conditions are not satisfied, the function H_n, respectively $\hat{H}_n$, is not passive. Otherwise, one proceeds to check the eigenvalue-based characterization of passivity recently proposed in [6]. More precisely, in [6], it is shown that the function H_n given by (52) is passive if, and only if, the following three conditions are satisfied:

(i) H_n is stable;
(ii) $(\mathbf{l}^{\mathrm{T}}\mathbf{r})\ \det(\mathbf{M}+\lambda_0\,\mathbf{N}) \geq 0$ for a given $\lambda_0 \geq 0$;
(iii) The matrix pencil $\mathbf{M}+\lambda\,\mathbf{N}$ has either no real positive eigenvalues or if any real positive eigenvalue has even multiplicity.

Here, the matrix pencil $\mathbf{M}+\lambda\,\mathbf{N}$ is given by

$$\mathbf{M} := (\mathbf{I}+s_0\,\mathbf{T}_n)^2\left(\mathbf{I}-\mathbf{e}_1\,\mathbf{e}_1^{\mathrm{T}}\right)+(\mathbf{I}+s_0\,\mathbf{T}_n)\,\mathbf{e}_1\,\mathbf{e}_1^{\mathrm{T}}, \quad \mathbf{N} := \mathbf{T}_n^2\left(\mathbf{I}-\mathbf{e}_1\,\mathbf{e}_1^{\mathrm{T}}\right).$$

Analogously, for checking the passivity of $\hat{H}_n$, one simply needs to replace H_n by $\hat{H}_n$ and $\mathbf{T}_n$ by $\widehat{\mathbf{T}}_n$ in the above conditions (i)–(iii).

Remark 9 The post-processing Steps 3) to 7) are all performed on $n \times n$ matrices and they involve $\mathcal{O}(n^3)$ arithmetic operations. Since it is expected that $n \ll N$, the overall cost of these post-processing steps should not be significant.

Remark 10 We are not aware of any systematic choice of the prescribed poles and zeros in Step 4) that would guarantee stability, and possibly passivity, of the associated partial Padé approximant. For the numerical experiments reported in Section 7, we chose the prescribed poles and zeros by suitable reflections of the unstable poles and zeros with respect to the imaginary axis, and this lead to stable and passive models.

Remark 11 In general, the modified matrix (26), $\widehat{\mathbf{T}}_n$, is no longer tridiagonal if $m > 2$. However, it is easy to reduce $\widehat{\mathbf{T}}_n$ again to tridiagonal form. Indeed, all one needs to do is run n steps of the Lanczos process applied to the matrix $\widehat{\mathbf{T}}_n$ with right starting vector $\tilde{\mathbf{v}}_1 = \mathbf{e}_1$ and left starting vector $\tilde{\mathbf{w}}_1 = \mathbf{e}_1$. Again, for simplicity, we assume that no breakdown occurs in the Lanczos process. Then, after n steps, the algorithm has generated a tridiagonal matrix $\widetilde{\mathbf{T}}_n \in \mathbb{R}^{n\times n}$ and two matrices $\widetilde{\mathbf{V}}_n$, $\widetilde{\mathbf{W}}_n \in \mathbb{R}^{n\times n}$ of right and left Lanczos vectors such that

$$\widetilde{\mathbf{W}}_n^{\mathrm{T}}\,\widehat{\mathbf{T}}_n\,\widetilde{\mathbf{V}}_n = \widetilde{\mathbf{W}}_n^{\mathrm{T}}\,\widetilde{\mathbf{V}}_n\,\widetilde{\mathbf{T}}_n \quad \text{and} \quad \widetilde{\mathbf{W}}_n^{\mathrm{T}}\,\widetilde{\mathbf{V}}_n = \operatorname{diag}\left(1, \tilde{\delta}_2, \ldots, \tilde{\delta}_n\right). \tag{54}$$

Furthermore, the matrices $\widetilde{\mathbf{V}}_n$, $\widetilde{\mathbf{W}}_n$, and $\widetilde{\mathbf{W}}_n^{\mathrm{T}}\,\widetilde{\mathbf{V}}_n$ are all nonsingular. Note that, by the second relation in (54), we have

$$\widetilde{\mathbf{W}}_n^{\mathrm{T}}\,\mathbf{e}_1 = \widetilde{\mathbf{W}}_n^{\mathrm{T}}\,\tilde{\mathbf{v}}_1 = \mathbf{e}_1 \quad \text{and} \quad \widetilde{\mathbf{V}}_n^{\mathrm{T}}\,\mathbf{e}_1 = \widetilde{\mathbf{V}}_n^{\mathrm{T}}\,\tilde{\mathbf{w}}_1 = \mathbf{e}_1. \tag{55}$$

Using (54) and (55), we can rewrite the formula (53) of $\hat{H}_n$ as follows:

$$\begin{aligned}
\hat{H}_n(s) &= (\mathbf{l}^{\mathrm{T}}\mathbf{r})\, \mathbf{e}_1^{\mathrm{T}} \left(\mathbf{I} - (s-s_0)\, \widehat{\mathbf{T}}_n\right)^{-1} \mathbf{e}_1 \\
&= (\mathbf{l}^{\mathrm{T}}\mathbf{r}) \left(\mathbf{e}_1^{\mathrm{T}} \widetilde{\mathbf{V}}_n\right) \left(\widetilde{\mathbf{W}}_n^{\mathrm{T}} \widetilde{\mathbf{V}}_n - (s-s_0)\, \widetilde{\mathbf{W}}_n^{\mathrm{T}} \widehat{\mathbf{T}}_n \widetilde{\mathbf{V}}_n\right)^{-1} \left(\widetilde{\mathbf{W}}_n^{\mathrm{T}} \mathbf{e}_1\right) \\
&= (\mathbf{l}^{\mathrm{T}}\mathbf{r})\, \mathbf{e}_1^{\mathrm{T}} \left(\mathbf{I} - (s-s_0)\, \widetilde{\mathbf{T}}_n\right)^{-1} \left(\widetilde{\mathbf{W}}_n^{\mathrm{T}} \widetilde{\mathbf{V}}_n\right)^{-1} \mathbf{e}_1 \\
&= (\mathbf{l}^{\mathrm{T}}\mathbf{r})\, \mathbf{e}_1^{\mathrm{T}} \left(\mathbf{I} - (s-s_0)\, \widetilde{\mathbf{T}}_n\right)^{-1} \mathbf{e}_1.
\end{aligned}$$

This shows that, in (53), we can simply replace $\widehat{\mathbf{T}}_n$ by the tridiagonal matrix $\widetilde{\mathbf{T}}_n$ obtained from n steps of the Lanczos process applied to $\widehat{\mathbf{T}}_n$ with $\mathbf{e}_1$ as both right and left starting vector.

7 Numerical examples

In this section, we present two circuit examples that demonstrate the effectiveness of the PVLπ method.

7.1 The PEEC circuit

Our first example is a circuit resulting from the so-called PEEC discretization [30] of an electromagnetic problem. This circuit has often been used as a test problem for reduced-order modeling techniques. The circuit is an RLC network consisting of only inductors, capacitors, and inductive couplings, and a single resistive source that drives the circuit. In this case, the transfer function H represents the current flowing through one of the inductors. The circuit is stable with all poles of H in the left half of the complex plane. However, since the circuit is mostly an LC network, most of the poles are close to the imaginary axis. Due to this proximity, during the computation of a reduced-order model, numerical and approximation errors can easily produce unstable poles $p_j \in \mathbb{C}_+$. Indeed, running the PVL algorithm (with expansion point $s_0 = 2\pi \times 10^9$) for $n = 60$ iterations produces an almost exact transfer function in the relevant frequency range $s = 2\pi\, \mathtt{i}\, \omega$, $0 \le \omega \le 5 \times 10^9$. The corresponding curves for $|H(s)|$ and $|H_{60}(s)|$ are shown in Figure 1. However, the Padé approximant H_{60} in not stable due to 15 unstables poles $p_j \in \mathbb{C}_+$. More precisely, 8 of these 15 are truly unstable poles, while the other 7 are nearly stable poles, as shown in Figure 2.

In order to produce a stable reduced-order model, we ran the PVLπ Algorithm 2 to force all the 15 unstable poles, $p_j = \mathrm{Re}(p_j) + \mathtt{i}\ \mathrm{Im}(p_j)$, into the left

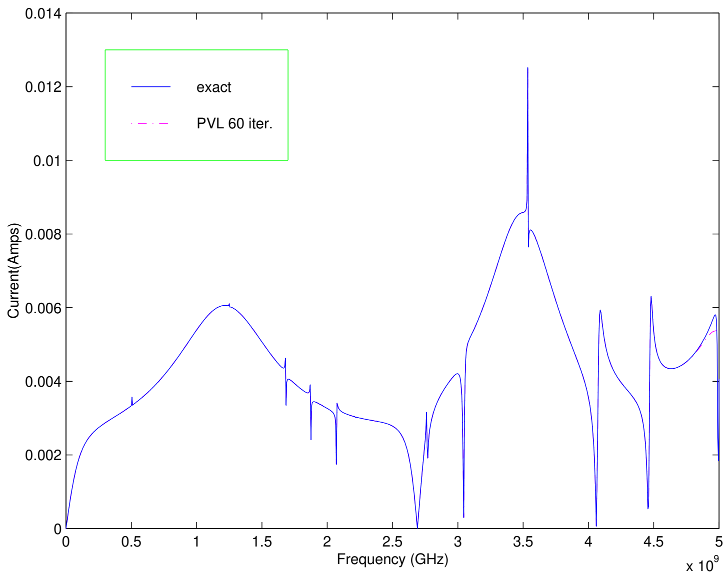

Fig. 1. The PEEC transfer function, exact and 60 PVL iterations

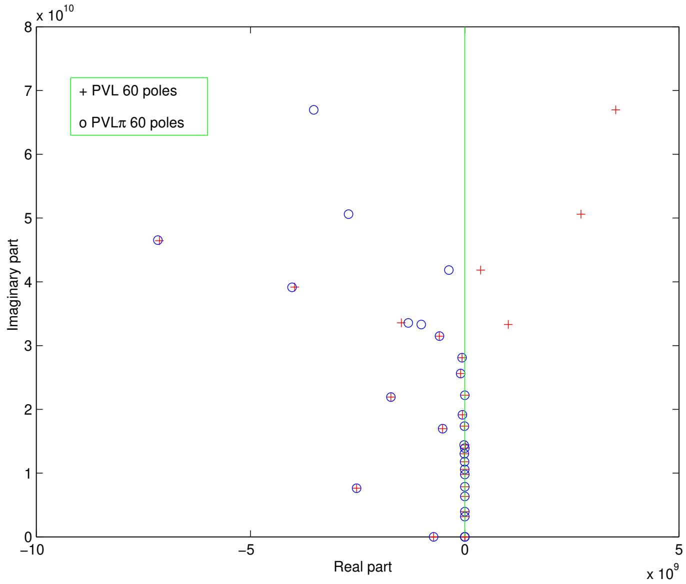

Fig. 2. The PEEC circuit, PVL poles and PVLπ poles

half-plane. This is done by setting, in Step 4) of Algorithm 2, $k = 15$, $\ell = 0$, and

$$\varphi_j = \frac{1}{\hat{p}_j - s_0}, \quad \text{where} \quad \hat{p}_j = -\operatorname{Re}(p_j) + \mathtt{i}\ \operatorname{Im}(p_j),$$

for all 15 unstable poles p_j of H_{60}. Figure 2 shows the poles of H_{60} and the poles of the modified reduced-order transfer function $\hat{H}_{60}$ generated by PVLπ. Note that $\hat{H}_{60}$ has all its poles in the left half plane, and thus $\hat{H}_{60}$ is stable. Figure 3 shows the curves $|H(s)|$ and $\left|\hat{H}_{60}(s)\right|$ for the relevant frequency range. In Figure 4, we plot both the PVL error curve $|H(s) - H_{60}(s)|$ and the

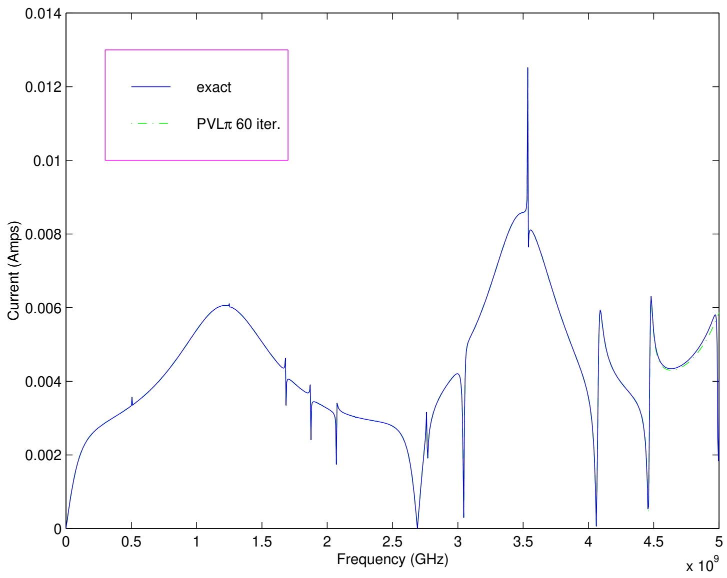

Fig. 3. The PEEC transfer function, exact and 60 PVLπ iterations

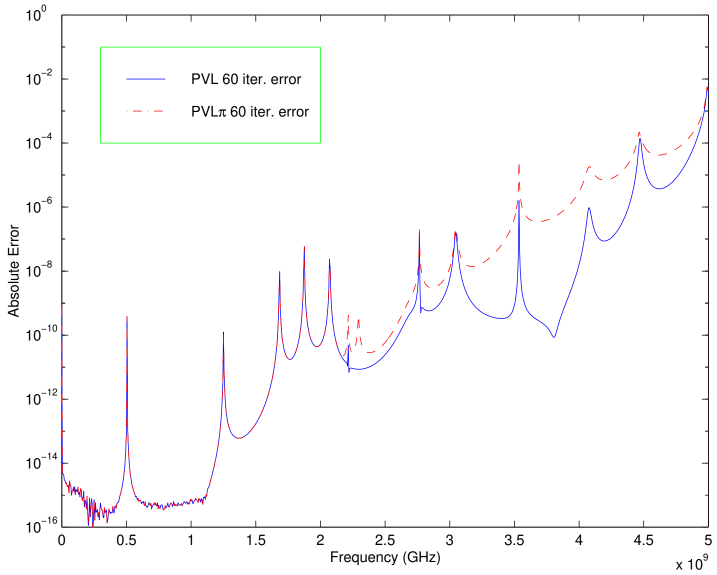

Fig. 4. PVL and PVLπ error curves for the PEEC transfer function

PVLπ error curve $\left|H(s) - \hat{H}_{60}(s)\right|$ for the relevant frequency range. These error curves show that the accuracy of the stable reduced-order transfer function $\hat{H}_{60}$ remains satisfactory.

7.2 A package model

Our second example is a circuit that arose in the analysis of a 64-pin package model used for an RF integrated circuit. The package model is described by approximately 4000 RLC circuit elements, resulting in matrices $\mathbf{C}$ and $\mathbf{G}$

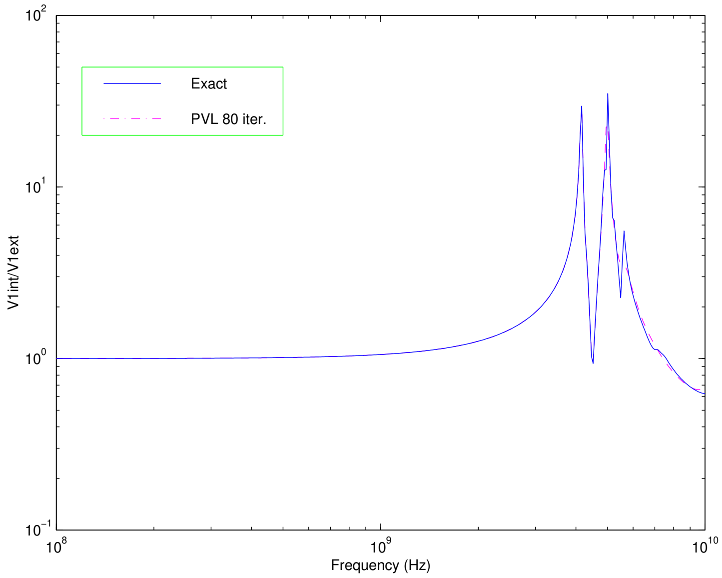

Fig. 5. The package transfer function, exact and 80 PVL iterations

in (2) of size about 2000. Here, we are interested in approximating the passive transfer function H that represents the input impedance of one of the pins of the package. The PVL algorithm (with expansion point $s_0 = 5\pi \times 10^9$) requires $n = 80$ iterations to generate a Padé approximant that approximates the exact transfer function H sufficiently well in the relevant frequency range $s = 2\pi\,\mathtt{i}\,\omega$, $0 \leq \omega \leq 10^{10}$. Figure 5 shows the corresponding curves for $|H(s)|$ and $|H_{80}(s)|$. However, the Padé approximant is neither stable nor passive due to 2 unstable poles $p_j \in \mathbb{C}_+$ and 4 unstable zeros $z_j \in \mathbb{C}_+$.

In order to produce a stable and passive reduced-order model, we ran the PVLπ Algorithm 2 to force the 2 unstable poles, $p_j = \mathrm{Re}(p_j) + \mathtt{i}\ \mathrm{Im}(p_j)$, and the 4 unstable zeros, $z_j = \mathrm{Re}(z_j) + \mathtt{i}\ \mathrm{Im}(z_j)$, into the left half-plane. This is done by setting, in Step 4) of Algorithm 2, $k = 2$, $\ell = 4$, and

$$\varphi_j = \frac{1}{\hat{p}_j - s_0}, \quad \text{where} \quad \hat{p}_j = -0.1\ \mathrm{Re}(p_j) + \mathtt{i}\,10\ \mathrm{Im}(p_j),$$
$$\theta_j = \frac{1}{\hat{z}_j - s_0}, \quad \text{where} \quad \hat{z}_j = -0.1\ \mathrm{Re}(z_j) + \mathtt{i}\,10\ \mathrm{Im}(z_j),$$

for all unstable poles and zeros of H_{80}. The resulting partial Padé approximant $\hat{H}_{80}$ produced by PVLπ now has all its poles and zeros in the left complex half-plane. Thus, $\hat{H}_{80}$ is stable and satisfies the necessary conditions for passivity stated in part b) of Theorem B. Furthermore, using the eigenvalue-based passivity test proposed in [6], we verified that $\hat{H}_{80}$ is indeed passive. Figure 6 shows the curves $|H(s)|$ and $\left|\hat{H}_{80}(s)\right|$ for the relevant frequency range. These curves clearly illustrate that no accuracy has been lost by enforcing stability and passivity by means of the PVLπ post-processing.

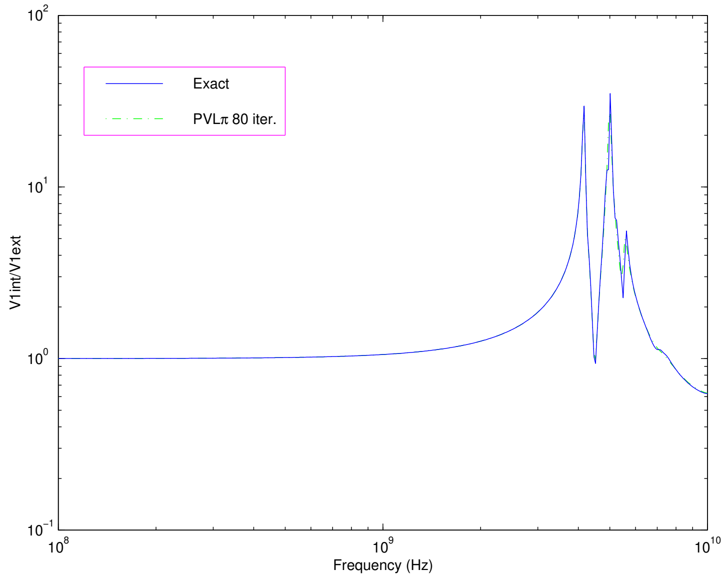

Fig. 6. The package transfer function, exact and 80 PVLπ iterations

8 Concluding remarks

We have introduced the PVLπ algorithm for generating reduced-order models based on partial Padé approximation via the Lanczos process. The algorithm can be viewed as a variant of PVL with added post-processing to remove possible unstable poles and zeros of the PVL reduced-order model.

There are still two important open problems. First, we are not aware of any systematic way of choosing the prescribed poles and zeros of the partial Padé approximants so that stability, and possibly passivity, of the corresponding reduced-order model can always be guaranteed. While we obtained stable and passive models by simply prescribing reflections of the unstable PVL poles and zeros with respect to the imaginary axis, there is a definite need for automating the selection of the prescribed poles and zeros. One possibility is to use an optimization procedure that minimizes some suitable measure of distance to stability, and possibly passivity. Such an optimization procedure consists of an outer iteration for the choice of the prescribed poles and zeros, while PVLπ is employed to generate the associated partial Padé approximant within each outer iteration. Work in this direction is in progress and will be reported elsewhere.

Second, this paper only treats the case of scalar transfer functions. In [13], the PVL algorithm has been extended to the general case of matrix-valued transfer functions of multi-input multi-output linear dynamical systems. It is also possible to extend PVLπ to this more general case, by using the concept of partial matrix-Padé approximation. Such an extension would be beyond the scope of this paper, and instead, this will be done in some future report.

Acknowledgement

We would like to thank Peter Feldmann for providing the circuit examples for our numerical tests. The authors are grateful to the referees and the editor for their constructive comments that helped us to improve the presentation of the paper.

References

[1] F.J. Alexandro, Jr., Stable partial Padé approximations for reduced-order transfer functions, IEEE Trans. Automat. Control 29 (1984) 159–162.

[2] D.F. Anastasakis, N. Gopal, L.T. Pillage, Enhancing the stability of asymptotic waveform evaluation for digital interconnect circuit applications, IEEE Trans. Computer-Aided Design 13 (1994) 729–736.

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

[4] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9 (1951) 17–29.

[5] Z. Bai, P. Feldmann, R.W. Freund, Stable and passive reduced-order models based on partial Padé approximation via the Lanczos process, Numerical Analysis Manuscript No. 97–3–10, Bell Laboratories, Murray Hill, NJ, 1997. Available online from `http://cm.bell-labs.com/cs/doc/97`.

[6] Z. Bai, R.W. Freund, Eigenvalue-based characterization and test for positive realness of transfer functions, IEEE Trans. Automat. Control 45 (2000), to appear.

[7] Z. Bai, Q. Ye, Error estimation of the Padé approximation of transfer functions via the Lanczos process, Electron. Trans. Numer. Anal. 7 (1998) 1–17.

[8] G.A. Baker, Jr., P. Graves-Morris, Padé Approximants, 2nd edition, Cambridge University Press, New York, NY, 1996.

[9] C. Brezinski, Partial Padé approximation, J. Approx. Theory 54 (1988) 210–233.

[10] C.-T. Chen, Linear System Theory and Design, Oxford University Press, New York, NY, 1984.

[11] P. Feldmann, R.W. Freund, Efficient linear circuit analysis by Padé approximation via the Lanczos process, in: Proceedings of EURO-DAC '94 with EURO-VHDL '94, IEEE Computer Society Press, Los Alamitos, CA, 1994, pp. 170–175.

[12] P. Feldmann, R.W. Freund, Efficient linear circuit analysis by Padé approximation via the Lanczos process, IEEE Trans. Computer-Aided Design 14 (1995) 639–649.

[13] P. Feldmann, R.W. Freund, Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm, in: Proceedings of the 32nd Design Automation Conference, ACM, New York, NY, 1995, pp. 474–479.

[14] L. Fortuna, G. Nunnari, A. Gallo, Model Order Reduction Techniques with Applications in Electrical Engineering, Springer-Verlag, London, UK, 1992.

[15] R.W. Freund, Circuit simulation techniques based on Lanczos-type algorithms, in: Systems and Control in the Twenty-First Century (C.I. Byrnes, B.N. Datta, D.S. Gilliam, and C.F. Martin, Eds.), Birkhäuser, Boston, MA, 1997, pp. 171–184.

[16] R.W. Freund, Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation, in: Applied and Computational Control, Signals, and Circuits, Vol. 1 (B.N. Datta, Ed.), Birkhäuser, Boston, MA, 1999, pp. 435–498.

[17] R.W. Freund, Passive reduced-order models for interconnect simulation and their computation via Krylov-subspace algorithms, in: Proceedings of the 36th Design Automation Conference, ACM, New York, NY, 1999, pp. 195–200.

[18] R.W. Freund, P. Feldmann, Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm, in: Technical Digest of the 1996 IEEE/ACM International Conference on Computer-Aided Design, IEEE Computer Society Press, Los Alamitos, CA, 1996, pp. 280–287.

[19] R.W. Freund, P. Feldmann, Reduced-order modeling of large linear passive multi-terminal circuits using matrix-Padé approximation, in: Proceedings of the Design, Automation and Test in Europe Conference 1998, IEEE Computer Society Press, Los Alamitos, CA, 1998, pp. 530–537.

[20] R.W. Freund, M.H. Gutknecht, N.M. Nachtigal, An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices, SIAM J. Sci. Comput. 14 (1993) 137–158.

[21] K. Gallivan, E.J. Grimme, P. Van Dooren, Asymptotic waveform evaluation via a Lanczos method, Appl. Math. Lett. 7 (1994) 75–80.

[22] G.H. Golub, J. Kautsky, Calculation of Gauss quadratures with multiple free and fixed knots, Numer. Math. 41 (1983) 147–163.

[23] W.B. Gragg, Matrix interpretations and applications of the continued fraction algorithm, Rocky Mountain J. of Math. 5 (1974) 213–225.

[24] E.J. Grimme, D.C. Sorensen, P. Van Dooren, Model reduction of state space systems via an implicitly restarted Lanczos method, Numer. Algorithms 12 (1996) 1–31.

[25] J. Kautsky, S. Elhay, Gauss quadratures and Jacobi matrices for weight functions not of one sign, Math. Comp. 43 (1984) 543–550.

[26] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bur. Standards 45 (1950) 255–282.

[27] A. Odabasioglu, M. Celik, L.T. Pileggi, PRIMA: passive reduced-order interconnect macromodeling algorithm, in: Technical Digest of the 1997 IEEE/ACM International Conference on Computer-Aided Design, IEEE Computer Society Press, Los Alamitos, CA, 1997, pp. 58–65.

[28] L.T. Pillage, R.A. Rohrer, Asymptotic waveform evaluation for timing analysis, IEEE Trans. Computer-Aided Design 9 (1990) 352–366.

[29] R.A. Rohrer and H. Nosrati, Passivity considerations in stability studies of numerical integration algorithms, IEEE Trans. Circuits and Systems 28 (1981) 857–866.

[30] A.E. Ruehli, Equivalent circuit models for three-dimensional multiconductor systems, IEEE Trans. Microwave Theory Tech. 22 (1974) 216–221.

[31] Y. Shamash, Stable reduced-order models using Padé-type approximations, IEEE Trans. Automat. Control 19 (1974) 615–616.

[32] L.M. Silveira, M. Kamon, I. Elfadel, J. White, A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits, in: Technical Digest of the 1996 IEEE/ACM International Conference on Computer-Aided Design, IEEE Computer Society Press, Los Alamitos, CA, 1996, pp. 288–294.

[33] M.E. Van Valkenburg, Introduction to Modern Network Synthesis, John Wiley & Sons. Inc., New York, NY, 1960.