Passive reduced-order modeling via Krylov-subspace methods*

Roland W. Freund
Bell Laboratories
Room 2C–525
700 Mountain Avenue
Murray Hill, New Jersey 07974–0636, U.S.A.
freund@research.bell-labs.com

Keywords: linear system, transfer function, passivity, Lanczos process, projection

Abstract

In recent years, Krylov-subspace methods have become popular tools for computing reduced-order models of large-scale time-invariant linear dynamical systems. These techniques produce in a Padé sense optimal reduced-order models. However, when applied to passive systems, the Padé models do not preserve passivity in general. In this paper, we describe a Krylov-subspace technique that generates passive reduced-order models. Results of numerical experiments with a passive system arising in VLSI circuit simulation are reported.

1 Introduction

In recent years, there has been a surge of interest in reduced-order modeling of very large-scale linear dynamical systems. Much of the recent work in this area has been driven by the need for efficient reduced-order modeling techniques in the simulation of VLSI circuits. Today's integrated electronic circuits are extremely complex, with up to hundreds of millions of devices. Prototyping of such circuits is no longer possible, and instead, computational methods are used to simulate and analyze the behavior of the electronic circuit at the design stage. This allows to correct the design before the circuit is actually fabricated in silicon.

The simulation of electronic circuits involves the numerical solution of very large-scale, sparse, in general nonlinear, systems of time-dependent differential-algebraic equations (DAEs); see, e.g., [1, 2] and the references given there. These systems can be so large that time integration becomes inefficient or even prohibitive. On the other hand, electronic circuits often contain large linear subcircuits of passive components that are described by large time-invariant linear dynamical systems. In particular, such linear subcircuits may result from extracted RLC models of the circuit's wiring, the so-called interconnect, models of the circuit's package, or models of wireless propagation channels. By replacing the large linear dynamical systems corresponding to linear subcircuits by reduced-order models of much smaller state-space dimensions, the size of the system of DAEs describing the whole circuit can be reduced significantly, so that time integration becomes feasible; see, e.g., [3, 4, 5] and the references given there.

In circuit simulation, Krylov-subspace techniques, such as suitable variants of the Lanczos or Arnoldi process, have emerged as the methods of choice for generating reduced-order models of large linear subcircuits; for a recent survey, we refer the reader to [6]. For example, Lanczos-type methods produce reduced-order models that are in a Padé sense optimal approximations to the original linear dynamical system. Despite the success of Krylov-subspace methods, there are still some important open issues, such as the preservation of passivity. In circuit simulation, reduced-order modeling is mostly applied to passive linear subcircuits, and preservation of passivity is crucial in order to ensure the stability of the simulation of the whole circuit; see, e.g., [7, 8]. Unfortunately, "optimal" Padé reduced-order models, in general, do not preserve passivity.

In this paper, we describe reduced-order modeling techniques based on Krylov subspaces that preserve passivity. While the resulting passive reduced-order models are no longer optimal in the Padé sense, they still satisfies a somewhat weaker approximation property.

The remainder of the paper is organized as follows. In Section 2, we review some basic material on linear dynamical systems. In Section 3, we introduce block Krylov subspaces and review the construction of suitable basis vectors. In Section 4, we define reduced-order models based on projection, discuss their approximation properties, and describe their actual computation. In Section 5, we establish passivity of these reduced-order models. In Section 6, numerical results for a passive system arising in VLSI circuit simulation are reported. In Section 7, we make some concluding remarks.

*Numerical Analysis Manuscript No. 00–3–02, Bell Laboratories, Murray Hill, New Jersey, March 2000. Available on WWW at http://cm.bell-labs.com/cs/doc/00.

Throughout this article, we use boldface letters to denote vectors and matrices. All vectors and matrices are assumed to be real. As usual, $\mathbf{M}^{\mathrm{T}} = [\, m_{kj} \,]$ denotes the transpose of the matrix $\mathbf{M} = [\, m_{jk} \,]$, and $\mathbf{M} \geq \mathbf{0}$ means that $\mathbf{M}$ is symmetric positive semi-definite. We use $\mathbf{I}_n$ to denote the $n \times n$ identity matrix and $\mathbf{0}_{n \times m}$ to denote the $n \times m$ zero matrix; we will omit these indices whenever the actual dimensions of $\mathbf{I}$ and $\mathbf{0}$ are apparent from the context. 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. Finally, $\mathbb{C}_+ := \{\, s \in \mathbb{C} \mid \mathrm{Re}(s) > 0 \,\}$ is the open right half of the complex plane.

2 Reduced-order modeling of linear dynamical systems

In this section, we review some basic material on linear dynamical systems and reduced-order models.

2.1 Linear dynamical systems

We consider multi-input multi-output time-invariant linear dynamical systems of the form

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

where $\mathbf{G}$, $\mathbf{C} \in \mathbb{R}^{N \times N}$, $\mathbf{B} \in \mathbb{R}^{N \times m}$, and $\mathbf{L} \in \mathbb{R}^{N \times p}$ are given matrices. Note that m and p denote the number of inputs and outputs, respectively, the components of the given vector-valued function $\mathbf{u}\colon [0, \infty) \mapsto \mathbb{R}^m$ are the inputs, and $\mathbf{y}\colon [0, \infty) \mapsto \mathbb{R}^p$ is the unknown function of outputs. The components of the unknown vector-valued function $\mathbf{x}\colon [0, \infty) \mapsto \mathbb{R}^N$ are the state variables, and N is the state-space dimension. In general, the matrices $\mathbf{C}$ and $\mathbf{G}$ in (1) are allowed to be singular. However, we assume that the matrix pencil $\mathbf{G} + s\,\mathbf{C}$ is regular.

2.2 Reduced-order models

A reduced-order model of (1) is a linear dynamical system of the same form as (1), but of smaller state-space dimension $n < N$. Thus, a reduced-order model of state-space dimension n is of the form

$$\begin{aligned} \mathbf{C}_n\,\frac{d\mathbf{z}}{dt} &= -\mathbf{G}_n\,\mathbf{z} + \mathbf{B}_n\,\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{L}_n^{\mathrm{T}}\,\mathbf{z}(t). \end{aligned} \tag{2}$$

Here, $\mathbf{G}_n$, $\mathbf{C}_n \in \mathbb{R}^{n \times n}$, $\mathbf{B}_n \in \mathbb{R}^{n \times m}$, and $\mathbf{L}_n \in \mathbb{R}^{n \times p}$ are matrices that should be chosen such that the input-output mapping $\mathbf{u}(t) \mapsto \mathbf{y}(t)$ of (2) somehow approximates the input-output mapping of the system (1).

In frequency domain, the input-output relation of the original system (1), respectively the reduced-order model (2), is given by the transfer function

$$\mathbf{H}(s) := \mathbf{L}^{\mathrm{T}}\,(\mathbf{G} + s\,\mathbf{C})^{-1}\,\mathbf{B}, \quad s \in \mathbb{C}, \tag{3}$$

respectively

$$\mathbf{H}_n(s) := \mathbf{L}_n^{\mathrm{T}}\,(\mathbf{G}_n + s\,\mathbf{C}_n)^{-1}\,\mathbf{B}_n, \quad s \in \mathbb{C}.$$

Note that $\mathbf{H}, \mathbf{H}_n : \mathbb{C} \mapsto (\mathbb{C} \cup \{\infty\})^{p \times m}$ are matrix-valued rational functions.

All reduced-order models discussed in this paper satisfy a "moment-matching" property of the form

$$\mathbf{H}_n(s) = \mathbf{H}(s) + \mathcal{O}\,(s - s_0)^{q(n)}\,. \tag{4}$$

Here, $s_0 \in \mathbb{R}$ is a suitably chosen expansion point. If $q(n)$ in (4) is as large as possible, then $\mathbf{H}_n$ is an *n-th matrix-Padé approximant* of $\mathbf{H}$; see, e.g., [9]. We will also discuss certain *matrix-Padé-type approximants* for which $q(n)$ is not maximal.

2.3 Passive systems

An important special case is passive systems. Roughly speaking, a (linear or nonlinear) dynamical system is *passive* if it does not generate energy. For example, electrical networks consisting of only resistors, inductors, and capacitors are passive.

The following well-known theorem (see, e.g., [10, 11]) relates the passivity of linear dynamical systems (1) with $m = p$ to the positive realness of their transfer functions.

Theorem 1 *A linear dynamical system* (1) *with $m = p$ inputs and outputs is passive if, and only if, the associated transfer function* (3)*, $\mathbf{H}$ is positive real.*

The definition of a positive real matrix-valued function is as follows; see, e.g., [10].

Definition 2 *A function $\mathbf{H} : \mathbb{C} \mapsto (\mathbb{C} \cup \{\infty\})^{m \times m}$ is called positive real if*

(i) *$\mathbf{H}$ has no poles in $\mathbb{C}_+$;*

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

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

Recall that $\mathbf{H}$ is *stable* if $\mathbf{H}$ has no poles in $\mathbb{C}_+$ and if all possible purely imaginary poles of $\mathbf{H}$ are simple. Hence any positive real function is necessary stable. In the following theorem, we state a sufficient condition for positive realness; this result is proved in [12].

Theorem 3 *Let $\mathbf{G}$, $\mathbf{C} \in \mathbb{R}^{N \times N}$, and $\mathbf{B} \in \mathbb{R}^{N \times m}$. Assume that $\mathbf{G} + \mathbf{G}^{\mathrm{T}} \geq 0$, $\mathbf{C} = \mathbf{C}^{\mathrm{T}} \geq 0$, and that $\mathbf{G} + s\,\mathbf{C}$ is a regular matrix pencil. Then, the transfer function*

$$\mathbf{H}(s) := \mathbf{B}^{\mathrm{T}}\,(\mathbf{G} + s\,\mathbf{C})^{-1}\,\mathbf{B}, \quad s \in \mathbb{C},$$

is positive real.

2.4 Linear RLC subcircuits

In circuit simulation, an important special case is linear subcircuits that consist of only resistors, inductors, and capacitors. Such linear RLC subcircuits arise in the modeling of a circuit's interconnect and package; see, e.g., [13, 14, 3, 4].

The equations describing linear RLC subcircuits are of the form (1). Furthermore, as discussed [15, 14], the equations can be formulated such that the matrices in (1) have the following block structure:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{12}^{\mathrm{T}} & \mathbf{0} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{0} & -\mathbf{C}_{22} \end{bmatrix}, \\ \mathbf{B} = \mathbf{L} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{0}_{N_2 \times m} \end{bmatrix} \quad \text{with} \quad \mathbf{B}_1 \in \mathbb{R}^{N_1 \times m}. \tag{5}$$

Here, $\mathbf{G}_{11}$, $\mathbf{C}_{11} \in \mathbb{R}^{N_1 \times N_1}$ and $\mathbf{C}_{22} \in \mathbb{R}^{N_2 \times N_2}$ are symmetric positive semi-definite matrices, and $N = N_1 + N_2$. Note that the matrices $\mathbf{G}$ and $\mathbf{C}$ are symmetric, but in general indefinite. In view of (5), the transfer function (3) reduces to

$$\mathbf{H}(s) = \mathbf{B}^{\mathrm{T}} \, (\mathbf{G} + s\,\mathbf{C})^{-1} \, \mathbf{B}, \quad \mathbf{G} = \mathbf{G}^{\mathrm{T}}, \ \mathbf{C} = \mathbf{C}^{\mathrm{T}}. \tag{6}$$

We call a transfer function $\mathbf{H}$ *symmetric* if it is of the form (6) with real matrices $\mathbf{G}$, $\mathbf{C}$, and $\mathbf{B}$.

We will also use the following nonsymmetric formulation of (6). Let $\mathbf{J}$ be the block matrix

$$\mathbf{J} = \begin{bmatrix} \mathbf{I}_{N_1} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{N_2} \end{bmatrix}. \tag{7}$$

Note that, by (5) and (7), we have $\mathbf{B} = \mathbf{J}\,\mathbf{B}$. Using this relation, as well as (5), we can rewrite (6) as follows:

$$\mathbf{H}(s) = \mathbf{B}^{\mathrm{T}} \, (\mathbf{J}\,\mathbf{G} + s\,\mathbf{J}\,\mathbf{C})^{-1} \, \mathbf{B}, \\ \mathbf{J}\,\mathbf{G} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ -\mathbf{G}_{12}^{\mathrm{T}} & \mathbf{0} \end{bmatrix}, \quad \mathbf{J}\,\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} \end{bmatrix}. \tag{8}$$

In this formulation, the matrix $\mathbf{J}\,\mathbf{G}$ is no longer symmetric, but now

$$\mathbf{J}\,\mathbf{G} + (\mathbf{J}\,\mathbf{G})^{\mathrm{T}} \geq \mathbf{0} \quad \text{and} \quad \mathbf{J}\,\mathbf{C} \geq \mathbf{0}. \tag{9}$$

3 Bases for block Krylov subspaces

In this section, we introduce our notion of block Krylov subspaces for multiple starting vectors and discuss the construction of suitable basis vectors via Arnoldi and Lanczos algorithms.

3.1 Reduction to one matrix

Let $s_0 \in \mathbb{R}$ be the expansion point that is to be used in the characterization (4) of the reduced-order transfer function $\mathbf{H}_n$. The only assumption on s_0 is that the matrix $\mathbf{G} + s_0\,\mathbf{C}$ be nonsingular; this guarantees that s_0 is not a pole of the original transfer function (3), $\mathbf{H}$.

An approximate transfer function $\mathbf{H}_n$ satisfying (4) could be obtained by first explicitly computing the leading $q(n)$ Taylor coefficients of the expansion of $\mathbf{H}$ about s_0 and then generating $\mathbf{H}_n$ from these coefficients; see, e.g., [16]. However, any approach based on explicitly computing the Taylor coefficients of $\mathbf{H}$ is inherently numerically unstable; see [17]. A much better alternative is to use block Krylov-subspace methods that obtain the same information as contained in the leading $q(n)$ Taylor coefficients of $\mathbf{H}$, but in a more stable manner.

Before block Krylov-subspace methods can be employed, the two matrices $\mathbf{G}$ and $\mathbf{C}$ in the definition (3) of $\mathbf{H}$ have to be reduced to a single matrix, $\mathbf{A}$. This can be done by rewriting (3) as follows:

$$\mathbf{H}(s) = \mathbf{L}^{\mathrm{T}} \, (\mathbf{I} + (s - s_0)\,\mathbf{A})^{-1} \, \mathbf{R}, \\ \mathbf{A} := (\mathbf{G} + s_0\,\mathbf{C})^{-1} \, \mathbf{C}, \quad \mathbf{R} := (\mathbf{G} + s_0\,\mathbf{C})^{-1} \, \mathbf{B}. \tag{10}$$

In typical applications, $\mathbf{G}$ and $\mathbf{C}$ are large, but sparse matrices. However, the matrix $\mathbf{A}$ in (10) is dense in general. Block Krylov-subspace methods involve $\mathbf{A}$ only in the form of matrix-vector products $\mathbf{A}\,\mathbf{v}$ and possibly $\mathbf{A}^{\mathrm{T}}\,\mathbf{w}$. To efficiently compute these products, one never needs to form $\mathbf{A}$ explicitly. Instead, one precomputes a the sparse LU factorization of $\mathbf{G} + s_0\,\mathbf{C}$. Each matrix-vector product $\mathbf{A}\,\mathbf{v}$ then requires one multiplication with the sparse matrix $\mathbf{C}$ and two backsolves with the sparse triangular factors of $\mathbf{G} + s_0\,\mathbf{C}$. Similarly, $\mathbf{A}^{\mathrm{T}}\,\mathbf{w}$ requires one multiplication with $\mathbf{C}^{\mathrm{T}}$ and two sparse backsolves.

3.2 Block Krylov subspaces

Let $\mathbf{A} \in \mathbb{R}^{N \times N}$ be a given $N \times N$ matrix and

$$\mathbf{R} = [\,\mathbf{r}_1 \quad \mathbf{r}_2 \quad \cdots \quad \mathbf{r}_m\,] \in \mathbb{R}^{N \times m} \tag{11}$$

be a given matrix of m *right starting vectors*. Before we introduce block Krylov subspaces induced by $\mathbf{A}$ and $\mathbf{R}$, we briefly review the standard case $m = 1$ of a single starting vector $\mathbf{r} = \mathbf{r}_1$. In this case, the usual *n-th Krylov subspace* (induced by $\mathbf{A}$ and $\mathbf{r}$) is given by

$$\mathcal{K}_n(\mathbf{A}, \mathbf{r}) := \mathrm{span}\,\{\,\mathbf{r}, \mathbf{A}\,\mathbf{r}, \mathbf{A}^2\,\mathbf{r}, \ldots, \mathbf{A}^{n-1}\,\mathbf{r}\,\}. \tag{12}$$

Let n_0 be defined as the largest possible integer n such that in (12) the vectors $\mathbf{A}^{j-1}\,\mathbf{r}$, $1 \leq j \leq n-1$, are linearly independent. Note that $n_0 \leq N$. By the definition of n_0, the n-th Krylov subspace (12) has dimension n if $1 \leq n \leq n_0$ and dimension n_0 if $n > n_0$. Moreover, $\mathcal{K}_n(\mathbf{A}, \mathbf{r}) = \mathcal{K}_{n_0}(\mathbf{A}, \mathbf{r})$ for all $n > n_0$. Thus, $\mathcal{K}_{n_0}(\mathbf{A}, \mathbf{r})$ is the largest possible Krylov subspace (induced by $\mathbf{A}$ and $\mathbf{r}$), and we call the Krylov sequence $\mathbf{r}, \mathbf{A}\,\mathbf{r}, \mathbf{A}^2\,\mathbf{r}, \ldots, \mathbf{A}^{n-1}\,\mathbf{r}$ *exhausted* if $n > n_0$.

In the general case of $m \geq 1$ starting vectors (11), the situation is more complicated; see [18]. The main

difficulty is that in contrast to the case $m = 1$, linear independence of the columns in the block Krylov sequence,

$$\mathbf{R}, \mathbf{A}\,\mathbf{R}, \mathbf{A}^2\,\mathbf{R}, \ldots, \mathbf{A}^{j-1}\,\mathbf{R}, \ldots, \tag{13}$$

is lost only gradually in general. More precisely, if the j-th block, $\mathbf{A}^{j-1}\,\mathbf{R}$, contains a column that is linearly dependent on columns to its left in (13), then, in general, not all the columns of $\mathbf{A}^{j-1}\,\mathbf{R}$ are linear dependent on columns to their left. Hence, the occurrence of a linear dependent column does not necessarily imply that the block Krylov sequence $\mathbf{R}, \mathbf{A}\,\mathbf{R}, \mathbf{A}^2\,\mathbf{R}, \ldots, \mathbf{A}^{j-1}\,\mathbf{R}$ is exhausted. As a result, in general, the construction of suitable basis vectors for the subspaces spanned by the columns of (13) needs to be continued even after a linear dependent column has been found in (13). A proper handling of such linear dependent columns requires that the column itself and all its successive $\mathbf{A}$-multiples need to be deleted. Formally, this can be done as follows. By scanning the columns of the matrices in (13) from left to right and deleting each column that is linearly dependent on earlier columns, we obtain the *deflated* block Krylov sequence

$$\mathbf{R}_1, \mathbf{A}\,\mathbf{R}_2, \mathbf{A}^2\,\mathbf{R}_3, \ldots, \mathbf{A}^{j_{\max}-1}\,\mathbf{R}_{j_{\max}}. \tag{14}$$

This process of deleting linearly dependent vectors is called *exact deflation*. In (14), $\mathbf{R}_j$ is a submatrix of $\mathbf{R}_{j-1}$, with $\mathbf{R}_j \neq \mathbf{R}_{j-1}$ if, and only if, deflation occurred within the j-th Krylov block $\mathbf{A}^{j-1}\,\mathbf{R}$ in (13). Here, for $j = 1$, we set $\mathbf{R}_0 = \mathbf{R}$. Denoting by m_j the number of columns of $\mathbf{R}_j$, we thus have

$$m \geq m_1 \geq m_2 \geq \cdots \geq m_{j_{\max}} \geq 1.$$

By construction, the columns of the matrices (14) are linearly independent, and for each n, the subspace spanned by the first n of these columns is called the *n-th block Krylov subspace* (induced by $\mathbf{A}$ and $\mathbf{R}$). In the following, we denote the n-th block Krylov subspace by $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$. For later use, we remark that for

$$n = m_1 + m_2 + \cdots + m_j, \quad \text{where} \quad 1 \leq j \leq j_{\max},$$

the n-th block Krylov subspace is given by

$$\mathcal{K}_n(\mathbf{A}, \mathbf{R}) = \mathrm{Colspan}\left\{ \mathbf{R}_1, \mathbf{A}\,\mathbf{R}_2, \ldots, \mathbf{A}^{j-1}\,\mathbf{R}_j \right\}.$$

For Lanczos-based reduced-order modeling, we will also need the block Krylov subspaces induced by $\mathbf{A}^{\mathrm{T}}$ and a given matrix of *p left starting vectors*,

$$\mathbf{L} = [\,\mathbf{l}_1 \quad \mathbf{l}_2 \quad \cdots \quad \mathbf{l}_p\,] \in \mathbb{R}^{N \times p}. \tag{15}$$

Applying the above construction to $\mathbf{A}^{\mathrm{T}}$ and $\mathbf{L}$, the *n-th block Krylov subspace* (induced by $\mathbf{A}^{\mathrm{T}}$ and $\mathbf{L}$), $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$, is defined as the subspace spanned by the first n columns of the *deflated* block Krylov sequence

$$\mathbf{L}_1, \mathbf{A}^{\mathrm{T}}\,\mathbf{L}_2, \left(\mathbf{A}^{\mathrm{T}}\right)^2\,\mathbf{L}_3, \ldots, \left(\mathbf{A}^{\mathrm{T}}\right)^{k_{\max}-1}\,\mathbf{L}_{k_{\max}}. \tag{16}$$

We stress that in our construction of block Krylov subspaces, we only have used exact deflation of columns that are linearly dependent. In an actual algorithm for constructing basis vectors for $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$ and $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$ in finite-precision arithmetic, one also needs to delete vectors that are in some sense "almost" linearly dependent on earlier vectors. We will refer to the deletion of such almost linearly dependent vectors as *inexact deflation*. While inexact deflation is crucial in practice, concise statements of theoretical results are much simpler if only exact deflation is performed. Throughout this paper, theoretical results are thus formulated for exact deflation only.

For later use, we note the following invariance property of the block Krylov subspaces induced by (10).

Lemma 4 *Let $\mathbf{G}$, $\mathbf{C}$, $\mathbf{B}$ be the matrix triplet used in the definition of the matrices $\mathbf{A}$ and $\mathbf{R}$ in* (10)*, and let $\mathbf{J}$ be any nonsingular matrix of the same size as $\mathbf{A}$. Then the matrix triplets $\mathbf{G}$, $\mathbf{C}$, $\mathbf{B}$ and $\mathbf{J}\,\mathbf{G}$, $\mathbf{J}\,\mathbf{C}$, $\mathbf{J}\,\mathbf{B}$ lead to the same n-th block Krylov subspace $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$.*

Proof. By (10), we have

$$\mathbf{A} = (\mathbf{G} + s_0\,\mathbf{C})^{-1}\,\mathbf{C} = (\mathbf{J}\,\mathbf{G} + s_0\,\mathbf{J}\,\mathbf{C})^{-1}\,\mathbf{J}\,\mathbf{C},$$
$$\mathbf{R} = (\mathbf{G} + s_0\,\mathbf{C})^{-1}\,\mathbf{B} = (\mathbf{J}\,\mathbf{G} + s_0\,\mathbf{J}\,\mathbf{C})^{-1}\,\mathbf{J}\,\mathbf{B}.$$

Thus both matrix triplets result in the same matrices $\mathbf{A}$ and $\mathbf{R}$ and the associated block Krylov subspaces are identical. □

3.3 Basis vectors

The columns of the deflated block Krylov sequences (14) and (16), which are used in the above definitions of $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$ and $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$, respectively, tend to be almost linearly dependent even for moderate values of n. Thus, they should not be used in numerical computations. Instead, we construct other suitable basis vectors.

In the following,

$$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \in \mathbb{R}^N \tag{17}$$

denotes a set of basis vectors for $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$, i.e.,

$$\mathrm{span}\left\{ \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \right\} = \mathcal{K}_n(\mathbf{A}, \mathbf{R}).$$

The $N \times n$ matrix

$$\mathbf{V}_n := [\,\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n\,] \tag{18}$$

with columns (17) is called a *basis matrix* for $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$. Similarly,

$$\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n \in \mathbb{R}^N \tag{19}$$

denotes a set of basis vectors for $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$, i.e.,

$$\mathrm{span}\left\{ \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n \right\} = \mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L}).$$

The $N \times n$ matrix

$$\mathbf{W}_n := [\, \mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_n \,] \tag{20}$$

is called a *basis matrix* for $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$.

We stress that even though (17) and (19) are basis vectors of block Krylov subspaces, the algorithms discussed in this paper generate (17) and (19) in a vector-wise fashion, as opposed to the block-wise construction employed in more traditional block Krylov-subspace methods; see, e.g., the references given in [18]. There are two main reasons why the vector-wise construction is preferable to a block-wise construction. First, it greatly simplifies both the detection of necessary deflation and the actual deflation itself. Second, for Lanczos-type methods, which simultaneously construct basis vectors (17) and (19) for $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$ and $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$, respectively, only the vector-wise construction allows the treatment of the general case where the block sizes of the deflated block Krylov sequences (14) and (16) are not necessarily the same; for a detailed discussion, we refer the reader to [18].

3.4 Arnoldi basis

The classical Arnoldi process [19] generates orthonormal basis vectors for the sequence of Krylov subspaces $\mathcal{K}_n(\mathbf{A}, \mathbf{r})$, $n \geq 1$, induced by $\mathbf{A}$ and a single starting vector $\mathbf{r}$. In this subsection, we state an Arnoldi-type algorithm that extends the classical algorithm to block-Krylov subspaces $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$, $n \geq 1$.

Like the classical Arnoldi process, the algorithm constructs the basis vectors (17) to be orthonormal. In terms of the basis matrix (18), this orthonormality can be expressed as follows:

$$\mathbf{V}_n^{\mathrm{T}} \, \mathbf{V}_n = \mathbf{I}.$$

In addition to (17), the algorithm produces so-called *candidate vectors*,

$$\hat{\mathbf{v}}_{n+1}, \hat{\mathbf{v}}_{n+2}, \ldots, \hat{\mathbf{v}}_{n+m_{\mathrm{c}}}, \tag{21}$$

for the next m_{c} basis vectors $\mathbf{v}_{n+1}, \mathbf{v}_{n+2}, \ldots, \mathbf{v}_{n+m_{\mathrm{c}}}$. Here, $m_{\mathrm{c}} = m_{\mathrm{c}}(n)$ is the number m of columns in the starting block (11), $\mathbf{R}$, reduced by the number of exact and inexact deflations that have occurred so far. The candidate vectors (21) satisfy the orthogonality relation

$$\mathbf{V}_n^{\mathrm{T}} \, [\, \hat{\mathbf{v}}_{n+1} \quad \hat{\mathbf{v}}_{n+2} \quad \cdots \quad \hat{\mathbf{v}}_{n+m_{\mathrm{c}}} \,] = \mathbf{0}.$$

Due to the vector-wise construction of (17) and (21), detection of necessary deflation and the actual deflation becomes trivial. In fact, it is easy to show that exact deflation at step n of the Arnoldi-type process occurs if, and only if, $\hat{\mathbf{v}}_n = \mathbf{0}$. Similarly, inexact deflation occurs if, and only if, $\hat{\mathbf{v}}_n \approx \mathbf{0}$, but $\hat{\mathbf{v}}_n \neq \mathbf{0}$. Therefore, in the algorithm, one checks if some suitable norm of $\hat{\mathbf{v}}_n$ is close to zero. If it is, then $\hat{\mathbf{v}}_n$ is deflated by deleting $\hat{\mathbf{v}}_n$, shifting the indices of all remaining candidate vectors by -1, and setting $m_{\mathrm{c}} = m_{\mathrm{c}} - 1$. If this results in $m_{\mathrm{c}} = 0$, then the block-Krylov subspace is exhausted and the algorithm is stopped. Otherwise, the deflation procedure is repeated until a vector $\hat{\mathbf{v}}_n$ is found that passes the deflation check.

A complete statement of the resulting Arnoldi-type algorithm is as follows.

Algorithm 1 (Construction of Arnoldi basis.)

0) *Set* $\hat{\mathbf{v}}_i = \mathbf{r}_i$ *for* $i = 1, 2, \ldots, m$.
Set $m_{\mathrm{c}} = m$.

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

1) *Compute* $\|\hat{\mathbf{v}}_n\|_2$ *and check if* $\hat{\mathbf{v}}_n$ *should be deflated. If yes,* $\hat{\mathbf{v}}_n$ *is deflated by doing the following*:

   Set $m_{\mathrm{c}} = m_{\mathrm{c}} - 1$. *If* $m_{\mathrm{c}} = 0$, *set* $n = n - 1$ *and stop.*

   Set $\hat{\mathbf{v}}_i = \hat{\mathbf{v}}_{i+1}$ *for* $i = n, n+1, \ldots, n+m_{\mathrm{c}}-1$.

   Return to Step 1).

2) *Set* $t_{n,n-m_{\mathrm{c}}} = \|\hat{\mathbf{v}}_n\|_2$ *and* $\mathbf{v}_n = \hat{\mathbf{v}}_n / t_{n,n-m_{\mathrm{c}}}$.

3) *Compute* $\hat{\mathbf{v}}_{n+m_{\mathrm{c}}} = \mathbf{A}\, \mathbf{v}_n$.

4) *For* $i = 1, 2, \ldots, n$, *set*

$$t_{i,n} = \mathbf{v}_i^{\mathrm{T}} \, \hat{\mathbf{v}}_{n+m_{\mathrm{c}}} \quad \textit{and} \quad \hat{\mathbf{v}}_{n+m_{\mathrm{c}}} = \hat{\mathbf{v}}_{n+m_{\mathrm{c}}} - \mathbf{v}_i \, t_{i,n}.$$

5) *For* $i = n - m_{\mathrm{c}} + 1, n - m_{\mathrm{c}} + 2, \ldots, n-1$, *set*

$$t_{n,i} = \mathbf{v}_n^{\mathrm{T}} \, \hat{\mathbf{v}}_{i+m_{\mathrm{c}}} \quad \textit{and} \quad \hat{\mathbf{v}}_{i+m_{\mathrm{c}}} = \hat{\mathbf{v}}_{i+m_{\mathrm{c}}} - \mathbf{v}_n \, t_{n,i}.$$

3.5 Lanczos bases

The classical Lanczos process [20] generates two sequences of basis vectors (17) and (19) that span the Krylov subspaces $\mathcal{K}_n(\mathbf{A}, \mathbf{r})$ and $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{l})$, respectively, where $\mathbf{r}$ and $\mathbf{l}$ are single starting vectors. In [18], a Lanczos-type method was presented that extends the classical algorithm to block-Krylov subspaces $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$ and $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$ with blocks $\mathbf{R}$ and $\mathbf{L}$ of multiple right and left starting vectors (11) and (15).

Like the classical Lanczos process, the extended algorithm constructs the basis vectors (17) and (19) to be biorthogonal. In terms of the associated basis matrices (18) and (20), the biorthogonality can be expressed as follows:

$$\mathbf{W}_n^{\mathrm{T}} \, \mathbf{V}_n = \mathbf{\Delta}_n := \mathrm{diag}\,(\delta_1, \delta_2, \ldots \delta_n)\,. \tag{22}$$

Here, for simplicity, we have assumed that no look-ahead steps are necessary. This implies that $\mathbf{\Delta}_n$ is a diagonal matrix, as stated in (22), and that all diagonal entries of $\mathbf{\Delta}_n$ are nonzero. If look-ahead steps occur, then $\mathbf{\Delta}_n$ is a block-diagonal matrix; see [18] for further details.

In addition to (17) and (19), the algorithm constructs the candidate vectors

$$\hat{\mathbf{v}}_{n+1}, \ldots, \hat{\mathbf{v}}_{n+m_c} \quad \text{and} \quad \hat{\mathbf{w}}_{n+1}, \ldots, \hat{\mathbf{w}}_{n+p_c} \tag{23}$$

for the next m_c basis vectors $\mathbf{v}_{n+1}, \ldots, \mathbf{v}_{n+m_c}$ and the next p_c basis vectors $\mathbf{w}_{n+1}, \ldots, \mathbf{w}_{n+p_c}$, respectively. Here, as in Section 3.4, $m_c = m_c(n)$ is the number m of columns in the right starting block (11), $\mathbf{R}$, reduced by the number of exact and inexact deflations of candidate vectors $\hat{\mathbf{v}}_n$ that have occurred so far. Analogously, $p_c = p_c(n)$ is the number p of columns in the left starting block (15), $\mathbf{L}$, reduced by the number of exact and inexact deflations of candidate vectors $\hat{\mathbf{w}}_n$ that have occurred so far. Similar as in Section 3.4, a deflation of $\hat{\mathbf{v}}_n$, respectively $\hat{\mathbf{w}}_n$, is performed if a suitable norm of $\hat{\mathbf{v}}_n$, respectively $\hat{\mathbf{w}}_n$, is clsse to zero.

The candidate vectors (23) are constructed to satisfy the following biorthogonality relations:

$$\begin{aligned} \mathbf{W}_n^{\mathrm{T}} \, [\, \hat{\mathbf{v}}_{n+1} \quad \hat{\mathbf{v}}_{n+2} \quad \cdots \quad \hat{\mathbf{v}}_{n+m_c} \,] &= \mathbf{0}, \\ \mathbf{V}_n^{\mathrm{T}} \, [\, \hat{\mathbf{w}}_{n+1} \quad \hat{\mathbf{w}}_{n+2} \quad \cdots \quad \hat{\mathbf{w}}_{n+p_c} \,] &= \mathbf{0}. \end{aligned}$$

The coefficients in the actual recurrences used to generate the basis vectors (17) and (19) and the candidate vectors (23) can be summarized into a triplet of matrices $\boldsymbol{\rho}$, $\boldsymbol{\eta}$, and $\mathbf{T}_n$. For simplicity, we again assume that only exact deflation is performed. Then, these matrices satisfy the following relations:

$$\begin{aligned} \mathbf{V}_{m_1} \, \boldsymbol{\rho} = \mathbf{R}, \quad \mathbf{W}_{p_1} \, \boldsymbol{\eta} = \mathbf{L}, \\ \mathbf{W}_n^{\mathrm{T}} \, \mathbf{A} \, \mathbf{V}_n = \boldsymbol{\Delta}_n \, \mathbf{T}_n. \end{aligned} \tag{24}$$

A complete statement of the resulting Lanczos-type algorithm is as follows.

Algorithm 2 (Construction of Lanczos bases.)

0) *For* $i = 1, 2, \ldots, m$, *set* $\hat{\mathbf{v}}_i = \mathbf{r}_i$.
For $i = 1, 2, \ldots, p$, *set* $\hat{\mathbf{w}}_i = \mathbf{l}_i$.
Set $m_c = m$, $p_c = p$, *and* $\mathcal{I}_{\mathbf{v}} = \mathcal{I}_{\mathbf{w}} = \emptyset$.

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

1) *Compute* $\|\hat{\mathbf{v}}_n\|$ *and check if* $\hat{\mathbf{v}}_n$ *should be deflated. If yes,* $\hat{\mathbf{v}}_n$ *is deflated by doing the following*:

   If $n - m_c > 0$, *set* $\mathcal{I}_{\mathbf{w}} = \mathcal{I}_{\mathbf{w}} \cup \{n - m_c\}$.

   Set $m_c = m_c - 1$. *If* $m_c = 0$, *set* $n = n - 1$ *and stop.*

   For $i = n, n+1, \ldots, n+m_c-1$, *set* $\hat{\mathbf{v}}_i = \hat{\mathbf{v}}_{i+1}$.

   Return to Step 1).

2) *Compute* $\|\hat{\mathbf{w}}_n\|$ *and check if* $\hat{\mathbf{w}}_n$ *should be deflated. If yes,* $\hat{\mathbf{w}}_n$ *is deflated by doing the following*:

   If $n - p_c > 0$, *set* $\mathcal{I}_{\mathbf{v}} = \mathcal{I}_{\mathbf{v}} \cup \{n - p_c\}$.

   Set $p_c = p_c - 1$. *If* $p_c = 0$, *set* $n = n - 1$ *and stop.*

   For $i = n, n+1, \ldots, n+p_c-1$, *set* $\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i+1}$.

   Return to Step 2).

3) *Set* $t_{n,n-m_c} = \|\hat{\mathbf{v}}_n\|$ *and* $\tilde{t}_{n,n-p_c} = \|\hat{\mathbf{w}}_n\|$.
Normalize $\mathbf{v}_n = \dfrac{\hat{\mathbf{v}}_n}{t_{n,n-m_c}}$ *and* $\mathbf{w}_n = \dfrac{\hat{\mathbf{w}}_n}{\tilde{t}_{n,n-p_c}}$.

4) *Compute* $\delta_n = \mathbf{w}_n^{\mathrm{T}} \, \mathbf{v}_n$.
If $\delta_n = 0$, *stop.*

5) *Compute* $\hat{\mathbf{v}}_{n+m_c} = \mathbf{A} \, \mathbf{v}_n$.

6) *Set* $i_{\mathbf{v}} = \max\{1, n - p_c\}$.
Set $\mathcal{I}_{\mathbf{v}}^{(\mathrm{e})} = \mathcal{I}_{\mathbf{v}} \cup \{\, i_{\mathbf{v}}, i_{\mathbf{v}} + 1, \ldots, n - 1 \,\}$.
For $i \in \mathcal{I}_{\mathbf{v}}^{(\mathrm{e})}$ *(in ascending order), set*

$$t_{i,n} = \frac{\mathbf{w}_i^{\mathrm{T}} \, \hat{\mathbf{v}}_{n+m_c}}{\delta_i} \quad \text{and} \quad \hat{\mathbf{v}}_{n+m_c} = \hat{\mathbf{v}}_{n+m_c} - \mathbf{v}_i \, t_{i,n}.$$

7) *Compute* $\hat{\mathbf{w}}_{n+p_c} = \mathbf{A}^{\mathrm{T}} \, \mathbf{w}_n$.

8) *Set* $i_{\mathbf{w}} = \max\{1, n - m_c\}$.
Set $\mathcal{I}_{\mathbf{w}}^{(\mathrm{e})} = \mathcal{I}_{\mathbf{w}} \cup \{\, i_{\mathbf{w}}, i_{\mathbf{w}} + 1, \ldots, n - 1 \,\}$.
For $i \in \mathcal{I}_{\mathbf{w}}^{(\mathrm{e})}$ *(in ascending order), set*

$$\tilde{t}_{i,n} = \frac{\hat{\mathbf{w}}_{n+p_c}^{\mathrm{T}} \, \mathbf{v}_i}{\delta_i} \quad \text{and} \quad \hat{\mathbf{w}}_{n+p_c} = \hat{\mathbf{w}}_{n+p_c} - \mathbf{w}_i \, \tilde{t}_{i,n}.$$

9) *For* $i = n + 1, n + 2, \ldots, n + m_c$, *set*

$$t_{n,i-m_c} = \frac{\mathbf{w}_n^{\mathrm{T}} \, \hat{\mathbf{v}}_i}{\delta_n} \quad \text{and} \quad \hat{\mathbf{v}}_i = \hat{\mathbf{v}}_i - \mathbf{v}_n \, t_{n,i-m_c}.$$

10) *For* $i = n + 1, n + 2, \ldots, n + p_c$, *set*

$$\tilde{t}_{n,i-p_c} = \frac{\hat{\mathbf{w}}_i^{\mathrm{T}} \, \hat{\mathbf{v}}_n}{\delta_n} \quad \text{and} \quad \hat{\mathbf{w}}_i = \hat{\mathbf{w}}_i - \mathbf{w}_n \, \tilde{t}_{n,i-p_c}.$$

11) *For* $i \in \mathcal{I}_{\mathbf{w}}$, *set* $t_{n,i} = \dfrac{\tilde{t}_{i,n} \, \delta_i}{\delta_n}$.

12) *Set* $\mathbf{T}_n = [\, t_{ij} \,]_{i,j=1,2,\ldots,n}$.

We note that for symmetric transfer functions (6), Algorithm 2 can exploit the symmetry inherent in (6). Indeed, in this case, the Lanczos basis vectors (17) and (19) are connected as follows:

$$\mathbf{w}_n = \gamma_n \, (\mathbf{G} + s_0 \, \mathbf{C}) \, \mathbf{v}_n \quad \text{for all} \quad n. \tag{25}$$

Here, $\gamma_n \neq 0$ are suitable normalization factors. In view of (25), only the vectors (17) need to be generated. This results in a *symmetric* variant of Algorithm 2 that requires only half the computational work and storage of the general case; see [15, 13, 14] for further details. For later use, we note that for symmetric transfer functions (6), the matrices $\boldsymbol{\rho}$ and $\boldsymbol{\eta}$ in (24) are identical:

$$\boldsymbol{\rho} = \boldsymbol{\eta} \in \mathbb{R}^{m_1 \times m}. \tag{26}$$

4 Reduced-order models via projection

In this section, we introduce two reduced-order models based on a one-sided, respectively two-sided, projection.

4.1 Projection onto $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$

Let $\mathbf{V}_n \in \mathbb{R}^{N \times n}$ be a given matrix with full column rank n. By simply projecting the matrices $\mathbf{G}$, $\mathbf{C}$, $\mathbf{B}$, and $\mathbf{L}$ in the original linear dynamical system (1) onto the subspace spanned by the columns of $\mathbf{V}_n$, we obtain a reduced-order model (2) with matrices given by

$$\begin{aligned} \mathbf{G}_n &:= \mathbf{V}_n^{\mathrm{T}} \, \mathbf{G} \, \mathbf{V}_n, \quad \mathbf{C}_n := \mathbf{V}_n^{\mathrm{T}} \, \mathbf{C} \, \mathbf{V}_n, \\ \mathbf{B}_n &:= \mathbf{V}_n^{\mathrm{T}} \, \mathbf{B}, \quad \mathbf{L}_n := \mathbf{V}_n^{\mathrm{T}} \, \mathbf{L}. \end{aligned} \tag{27}$$

In particular, we now assume that $\mathbf{V}_n$ is a basis matrix (18) for $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$. In this case, the reduced-order model defined by (2) and (27) represents a (one-sided) projection of the original system (1) onto the n-th block-Krylov subspace $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$. In the sequel, we denote the associated reduced-order transfer function by

$$\mathbf{H}_n^{(1)}(s) := \mathbf{L}_n^{\mathrm{T}} \, (\mathbf{G}_n + s \, \mathbf{C}_n)^{-1} \, \mathbf{B}_n, \tag{28}$$

where the superscript $^{(1)}$ indicates the one-sided projection.

The following result shows that $\mathbf{H}_n^{(1)}$ is independent of the actual choice of the basis matrix $\mathbf{V}_n$ for $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$.

Proposition 5 *The function $\mathbf{H}_n^{(1)}$ given by* (27) *and* (28) *does not depend on the particular choice of the basis matrix* (18)*, $\mathbf{V}_n$, for the n-th block Krylov subspace $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$.*

Proof. The result immediately follows from the fact that any two basis matrices $\mathbf{V}_n$ and $\tilde{\mathbf{V}}_n$ for $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$ are connected via a relation of the form $\tilde{\mathbf{V}}_n = \mathbf{V}_n \, \mathbf{M}$ where $\mathbf{M}$ is a nonsingular $n \times n$ matrix. □

Although the function (28), $\mathbf{H}_n^{(1)}$, is defined via the simple projection (27), it satisfies an approximation property of the form (4), where, however, $q(n)$ is not maximal in general. This means that $\mathbf{H}_n^{(1)}$ is a matrix-Padé-type approximant of $\mathbf{H}$. For the special case of expansion point $s_0 = 0$ and a basis matrix $\mathbf{V}_n$ generated by a simple block Arnoldi procedure without deflation, this was first observed in [21, 7]. The following theorem extends this result to the most general case of arbitrary expansion points s_0 and arbitrary basis matrices $\mathbf{V}_n$ for the properly defined block Krylov subspaces $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$ that allow for necessary deflation of linearly dependent vectors. The only further assumption is that the matrix

$$\mathbf{G}_n + s_0 \, \mathbf{C}_n \quad \text{is nonsingular.} \tag{29}$$

This guarantees that s_0 is not a pole of $\mathbf{H}_n^{(1)}$. Since, by (27),

$$\mathbf{G}_n + s_0 \, \mathbf{C}_n = \mathbf{V}_n^{\mathrm{T}} \, (\mathbf{G} + s_0 \, \mathbf{C}) \, \mathbf{V}_n,$$

the condition (29) also ensures that the matrix $\mathbf{G} + s_0 \, \mathbf{C}$ is nonsingular.

A proof of the following result can be found in [12].

Theorem 6 *Let $n = m_1 + \cdots + m_j$, where $j \leq j_{\max}$, and let $\mathbf{H}_n^{(1)}$ be the reduced-order transfer function given by* (27) *and* (28)*. Let $s_0 \in \mathbb{C}$ be any expansion point such that* (29) *is satisfied. Then, $\mathbf{H}_n^{(1)}$ satisfies*

$$\mathbf{H}_n^{(1)}(s) = \mathbf{H}(s) + \mathcal{O}\left((s - s_0)^j\right),$$

i.e., $\mathbf{H}_n^{(1)}$ is a matrix-Padé-type approximant of the transfer function (3)*, $\mathbf{H}$.*

4.2 Projection onto $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$ and $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$

Let $\mathbf{V}_n$ and $\mathbf{W}_n$ be any two basis matrices of $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$ and $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$, and consider the representation (10) of the transfer function $\mathbf{H}$ of (1). By projecting the matrices in (10) from the right and left onto the columns of $\mathbf{V}_n$ and $\mathbf{W}_n$, respectively, we obtain the reduced-order transfer function

$$\begin{aligned} \mathbf{H}_n^{(2)}(s) &:= \mathbf{L}_n^{\mathrm{T}} \, (\mathbf{G}_n + (s - s_0) \, \mathbf{C}_n)^{-1} \, \mathbf{R}_n, \\ \mathbf{L}_n &:= \mathbf{V}_n^{\mathrm{T}} \, \mathbf{L}, \quad \mathbf{G}_n := \mathbf{W}_n^{\mathrm{T}} \, \mathbf{V}_n, \\ \mathbf{C}_n &:= \mathbf{W}_n^{\mathrm{T}} \, \mathbf{A} \, \mathbf{V}_n, \quad \mathbf{R}_n := \mathbf{W}_n^{\mathrm{T}} \, \mathbf{R}. \end{aligned} \tag{30}$$

In analogy to Proposition 5, we have the following result.

Proposition 7 *The function $\mathbf{H}_n^{(2)}$ given by* (30) *does not depend on the particular choice of the basis matrices $\mathbf{V}_n$ and $\mathbf{W}_n$ for the n-th block Krylov subspaces $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$ and $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$.*

It turns out that, in general, the reduced-order transfer function $\mathbf{H}_n^{(2)}$ defined in (30) is even a better approximation to $\mathbf{H}$ than $\mathbf{H}_n^{(1)}$. In fact, in [22, 12], we proved the following result.

Theorem 8 *Let $s_0 \in \mathbb{C}$ be any expansion point such that* (29) *is satisfied, and let $\mathbf{H}_n^{(2)}$ be the reduced-order transfer function given by the two-sided projection* (30)*. Then, $\mathbf{H}_n^{(2)}$ satisfies*

$$\mathbf{H}_n^{(2)}(s) = \mathbf{H}(s) + \mathcal{O}\left((s - s_0)^{q(n)}\right),$$

where $q(n)$ is as large as possible. Thus $\mathbf{H}_n^{(2)}$ is a matrix-Padé approximant of $\mathbf{H}$.

4.3 Computation via Krylov-subspace methods

In view of Proposition 7, the reduced-order transfer function (30), $\mathbf{H}_n^{(2)}$, can be generated from any two basis matrices $\mathbf{V}_n$ and $\mathbf{W}_n$. However, there is one distinguished choice of $\mathbf{V}_n$ and $\mathbf{W}_n$ that eliminates the need to explicitly compute the projections in (30). This choice is the Lanczos basese described in Section 3.5.

Assume that the Lanczos-type Algorithm 2 is run for n steps. Let $\mathbf{\Delta}_n$ be the diagonal matrix defined in (22), and let $\boldsymbol{\rho}$, $\boldsymbol{\eta}$, and $\mathbf{T}_n$ be the matrices in (24). Then, from (22) and (24), it readily follows that

$$\mathbf{W}_n^{\mathrm{T}}\,\mathbf{R} = \mathbf{\Delta}_n\,\boldsymbol{\rho}_n \quad \text{and} \quad \mathbf{V}_n^{\mathrm{T}}\,\mathbf{L} = \mathbf{\Delta}_n^{\mathrm{T}}\,\boldsymbol{\eta}_n, \tag{31}$$

where

$$\boldsymbol{\rho}_n := \begin{bmatrix} \boldsymbol{\rho} \\ \mathbf{0}_{n-m_1\times m} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\eta}_n := \begin{bmatrix} \boldsymbol{\eta} \\ \mathbf{0}_{n-p_1\times p} \end{bmatrix}. \tag{32}$$

Moreover, by (22) and the third relation in (24), we have

$$\mathbf{W}_n^{\mathrm{T}}\,\mathbf{A}\,\mathbf{V}_n = \mathbf{\Delta}_n\,\mathbf{T}_n \quad \text{and} \quad \mathbf{W}_n^{\mathrm{T}}\,\mathbf{V}_n = \mathbf{\Delta}_n. \tag{33}$$

Inserting (31) and (33) into (30), it readily follows that

$$\mathbf{H}_n^{(2)}(s) = \boldsymbol{\eta}_n^{\mathrm{T}}\left(\mathbf{\Delta}_n^{-1} + (s-s_0)\,\mathbf{T}_n\,\mathbf{\Delta}_n^{-1}\right)^{-1}\boldsymbol{\rho}_n. \tag{34}$$

The MPVL (matrix-Padé via Lanczos) algorithm, which was first proposed in [23], is a numerical procedure for computing $\mathbf{H}_n^{(2)}$ via the formula (34).

For symmetric transfer functions (6), by (26) and (32), the reduced-order transfer function (34) is also symmetric:

$$\mathbf{H}_n^{(2)}(s) = \boldsymbol{\rho}_n^{\mathrm{T}}\left(\mathbf{\Delta}_n^{-1} + (s-s_0)\,\mathbf{T}_n\,\mathbf{\Delta}_n^{-1}\right)^{-1}\boldsymbol{\rho}_n \tag{35}$$

where $\mathbf{\Delta}_n^{-1}$ and $\mathbf{T}_n\,\mathbf{\Delta}_n^{-1}$ are real symmetric matrices. The SyMPVL algorithm [13, 14] is a special symmetric variant of the general MPVL algorithm that computes symmetric reduced-order transfer functions (35).

5 Passivity and stability

Recall from Section 2.4 that RLC subcircuits are described by special symmetric transfer functions (6) with matrices $\mathbf{G}$, $\mathbf{C}$, and $\mathbf{B}$ of the form (5). In this case, the reduced-order transfer function (34) in general does not preserve the passivity of the RLC subcircuit. However, one can easily augment the SyMPVL algorithm to generate a second projected reduced-order model that, by Corollary 9 below, is always passive. To this end, let $\mathbf{J}$ be the matrix defined in (7), and consider the nonsymmetric formulation (8) of the symmetric transfer function (6). Note that by Lemma 4, both formulations (6) and (8) result in the same n-th block Krylov subspace $\mathcal{K}_n(\mathbf{A},\mathbf{R})$. In particular, the Lanczos basis matrix $\mathbf{V}_n$ generated by SyMPVL is also a basis matrix for the n-th block Krylov subspace associated with the nonsymmetric formulation (8). Hence we can also use $\mathbf{V}_n$ to apply the one-sided projection of Section 4.1 to (8). By (8), (27), and (28), the resulting projected reduced-order transfer function is given by

$$\mathbf{H}_n^{(1)}(s) := \left(\mathbf{V}_n^{\mathrm{T}}\,\mathbf{B}\right)^{\mathrm{T}}\left(\mathbf{G}_n + s\,\mathbf{C}_n\right)^{-1}\left(\mathbf{V}_n^{\mathrm{T}}\,\mathbf{B}\right) \tag{36}$$

where $\mathbf{G}_n + s\,\mathbf{C}_n$ is the matrix pencil

$$\mathbf{G}_n + s\,\mathbf{C}_n := \mathbf{V}_n^{\mathrm{T}}\,(\mathbf{J}\,\mathbf{G})\,\mathbf{V}_n + s\,\mathbf{V}_n^{\mathrm{T}}\,(\mathbf{J}\,\mathbf{C})\,\mathbf{V}_n. \tag{37}$$

As we discussed in Section 2.4, in circuit simulation, reduced-order modeling is mostly applied to large passive linear subcircuits, such as RLC networks. When reduced-order models of such subcircuits are used within a simulation of the whole circuit, stability of the overall simulation can only be guaranteed if the reduced-order models preserve the passivity of the original subcircuits; see, e.g., [24, 8]. Unfortunately, except for special cases such as RC subcircuits consisting of only resistors and capacitors, the Padé model given by $\mathbf{H}_n^{(2)}$ are not passive in general; see, e.g., [25, 26, 27, 15, 28]. The following corollary to Theorem 3 shows that the projected model given by (36) is passive.

Corollary 9 *Let* $\mathbf{H}$ *be the function given by* (8) *with matrices that satisfy* (9). *Let* $\mathbf{V}_n \in \mathbb{R}^{N\times n}$ *have rank* n *and assume that the matrix pencil* (37) *is regular. Then, the reduced-order transfer function* $\mathbf{H}_n^{(1)}$ *given by* (36) *and* (37) *is positive real, and thus the reduced-order model given by* (36) *is passive.*

Proof. By (9) and (37), it follows that $\mathbf{G}_n + \mathbf{G}_n^{\mathrm{T}} \geq 0$ and $\mathbf{C}_n = \mathbf{C}_n^{\mathrm{T}} \geq 0$. The transfer function (36), $\mathbf{H}_n^{(1)}$, is thus positive real by Theorem 3. □

6 Numerical examples

We consider an example that arises in the analysis of a 64-pin package model used for an RF integrated circuit. Only eight of the package pins carry signals, the rest being either unused or carrying supply voltages. The package is characterized as a passive linear dynamical system with $m = p = 16$ inputs and outputs, representing 8 exterior and 8 interior terminals. The package model is described by approximately 4000 circuit elements, resistors, capacitors, inductors, and inductive couplings, resulting in a linear dynamical system with a state-space dimension of about 2000.

In [13], SyMPVL was used to compute a Padé-based reduced-order model (35) of the package, and it was found that a model $\mathbf{H}_n^{(2)}$ of order $n = 80$ is sufficient to match the transfer-function components of interest. However, the model $\mathbf{H}_n^{(2)}$ has a few poles in the right half of the complex plane, and therefore, it is not passive.

In order to obtain a passive reduced-order model, we ran SyMPVL again on the package example, and this time, also generated the projected reduced-order model $\mathbf{H}_n^{(1)}$ given by (36). The expansion point $s_0 = 5\pi \times 10^9$ was used. Recall that $\mathbf{H}_n^{(1)}$ is only a Padé-type approximant and thus less accurate than the Padé approximant $\mathbf{H}_n^{(2)}$. Therefore, one now has to go to order $n = 112$ to obtain a projected reduced-order model $\mathbf{H}_n^{(1)}$ that

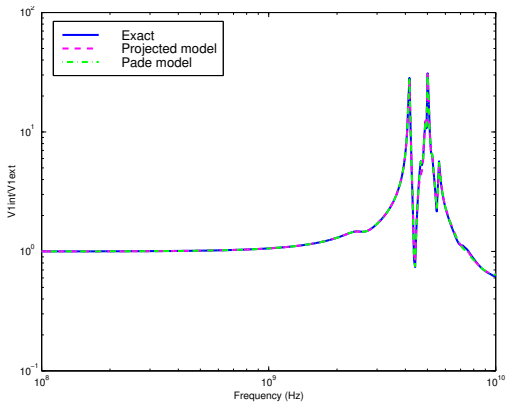

Figure 1: Pin no.1 external to Pin no.1 internal, exact, projected model, and Padé model

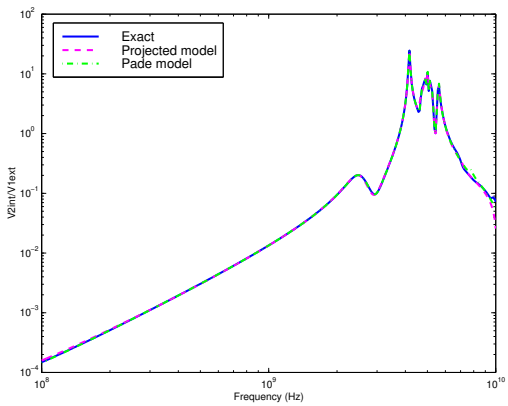

Figure 2: Pin no.1 external to Pin no.2 internal, exact, projected model, and Padé model

matches the transfer-function components of interest. Figures 1 and 2 show the voltage-to-voltage transfer function between the external terminal of pin no. 1 and the internal terminals of the same pin and the neighboring pin no. 2, respectively. The plots show results with the projected model $\mathbf{H}_n^{(1)}$ and the Padé model $\mathbf{H}_n^{(2)}$, both of order $n = 112$, compared with an exact analysis.

In Figure 3, we compare the relative error of the projected model $\mathbf{H}_{112}^{(1)}$ and the Padé model $\mathbf{H}_{112}^{(2)}$ of the same size. Clearly, the Padé model is more accurate. However, out of the 112 poles of $\mathbf{H}_{112}^{(2)}$, 22 have positive real part, violating the passivity of the Padé model. On the other hand, the projected model is passive.

7 Concluding remarks

In the last few years, reduced-order modeling techniques based on Krylov subspaces have become indispensable tools for tackling the large linear subcircuits that arise in the simulation of electronic circuits. Much of this development was and continues to be driven by the emerging need to accurately simulate the interconnect of electronic circuits. Today, circuit interconnect is typically modeled as large linear passive subcircuits that are generated by automatic parasitics-extraction programs. Using reduced-order modeling techniques has become crucial in order to reduce these subcircuits to a size that is manageable for circuit simulators.

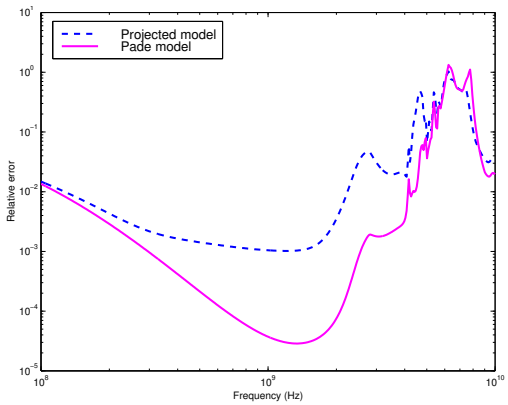

Figure 3: Relative error of projected and Padé model

To guarantee stability of the overall simulation, it is crucial that passive subcircuits are approximated by passive reduced-order models. While reduced-order models based on projection are passive, they are—in terms of number of matched Taylor coefficients—only half as accurate as the corresponding, in general non-passive, Padé models of the same size. It remains an open problem to describe and construct reduced-order models of a given size that are both passive and of maximal possible accuracy.

References

[1] Sangiovanni-Vincentelli A.L. *Circuit simulation.* In Antognetti P., Pederson D.O., and de Man D., eds., *Computer Design Aids for VLSI Circuits*, pp. 19–112. Sijthoff & Noordhoff International Publishers B.V., Alphen aan den Rijn, The Netherlands, 1981.

[2] Vlach J. and Singhal K. *Computer Methods for Circuit Analysis and Design.* Van Nostrand Reinhold, New York, second edition, 1993.

[3] Kim S.-Y., Gopal N., and Pillage L.T. *Time-domain macromodels for VLSI interconnect analysis. IEEE Trans. Computer-Aided Design*, 13, pp. 1257–1270, 1994.

[4] Pileggi L.T. *Coping with RC(L) interconnect design headaches.* Tech. Dig. *1995 IEEE/ACM International Conference on Computer-Aided Design*, pp. 246–253.

[5] Raghavan V., Bracken J.E., and Rohrer R.A. *AWESpice: A general tool for the accurate and efficient simulation of interconnect problems.* Proc.

29th ACM/IEEE Design Automation Conference, pp. 87–92, 1992.

[6] Freund R.W. *Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation.* In Datta B.N., ed., *Applied and Computational Control, Signals, and Circuits*, Vol. 1, pp. 435–498. Birkhäuser, Boston, 1999.

[7] Odabasioglu A., Celik M., and Pileggi L.T. *PRIMA: passive reduced-order interconnect macromodeling algorithm.* Tech. Dig. *1997 IEEE/ACM International Conference on Computer-Aided Design*, pp. 58–65.

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

[9] Baker Jr. G.A. and Graves-Morris P. *Padé Approximants.* Cambridge University Press, New York, second edition, 1996.

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

[11] Wohlers M.R. *Lumped and Distributed Passive Networks.* Academic Press, New York, 1969.

[12] Freund R.W. *Krylov-subspace methods for reduced-order modeling in circuit simulation. J. Comput. Appl. Math.*, 2000, to appear.

[13] Freund R.W. and Feldmann P. *The SyMPVL algorithm and its applications to interconnect simulation.* Proc. *1997 International Conference on Simulation of Semiconductor Processes and Devices*, pp. 113–116.

[14] Freund R.W. and Feldmann P. *Reduced-order modeling of large linear passive multi-terminal circuits using matrix-Padé approximation.* Proc. *Design, Automation and Test in Europe Conference 1998*, pp. 530–537.

[15] Freund R.W. and Feldmann P. *Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm.* Tech. Dig. *1996 IEEE/ACM International Conference on Computer-Aided Design*, pp. 280–287.

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

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

[18] Aliaga J.I., Boley D.L., Freund R.W., and Hernández V. *A Lanczos-type method for multiple starting vectors. Math. Comp.*, 2000, to appear.

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

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

[21] Odabasioglu A. *Provably passive RLC circuit reduction.* M.S. thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1996.

[22] Freund R.W. *Computation of matrix Padé approximations of transfer functions via a Lanczos-type process.* In Chui C.K. and Schumaker L.L., eds., *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, pp. 215–222. World Scientific Publishing Co., Inc., Singapore, 1995.

[23] Feldmann P. and Freund R.W. *Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm.* Proc. *32nd ACM/IEEE Design Automation Conference*, pp. 474–479, 1995.

[24] Chirlian P.M. *Integrated and Active Network Analysis and Synthesis.* Prentice-Hall, Englewood Cliffs, NJ, 1967.

[25] Bai Z., Feldmann P., and Freund R.W. *How to make theoretically passive reduced-order models passive in practice.* Proc. *IEEE 1998 Custom Integrated Circuits Conference*, pp. 207–210.

[26] Elfadel I.M. and Ling D.D. *Zeros and passivity of Arnoldi-reduced-order models for interconnect networks.* Proc. *34nd ACM/IEEE Design Automation Conference*, pp. 28–33, 1997.

[27] Freund R.W. *Passive reduced-order models for interconnect simulation and their computation via Krylov-subspace algorithms.* Proc. *36th ACM/IEEE Design Automation Conference*, pp. 195–200, 1999.

[28] Kerns K.J. and Yang A.T. *Preservation of passivity during RLC network reduction via split congruence transformations.* Proc. *34nd ACM/IEEE Design Automation Conference*, pp. 34–39, 1997.