ELSEVIER

Journal of Computational and Applied Mathematics 123 (2000) 395–421

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Krylov-subspace methods for reduced-order modeling in circuit simulation

Roland W. Freund

Bell Laboratories, Lucent Technologies, Room 2C-525, 700 Mountain Avenue, Murray Hill, New Jersey 07974-0636, USA

Received 11 September 1999; received in revised form 9 December 1999

Abstract

The simulation of electronic circuits involves the numerical solution of very large-scale, sparse, in general nonlinear, systems of differential-algebraic equations. Often, the size of these systems can be reduced considerably by replacing the equations corresponding to linear subcircuits by approximate models of much smaller state-space dimension. In this paper, we describe the use of Krylov-subspace methods for generating such reduced-order models of linear subcircuits. Particular emphasis is on reduced-order modeling techniques that preserve the passivity of linear RLC subcircuits. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Lanczos algorithm; Arnoldi process; Linear dynamical system; VLSI interconnect; Transfer function; Padé approximation; Stability; Passivity; Positive real function

1. Introduction

Today's integrated electronic circuits are extremely complex, with up to tens 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. [10,29,30] 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 contribute only linear equations to the system of DAEs describing the whole circuit. 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 equations corresponding to linear subcircuits by approximate models of much smaller state-space dimension, the size of the system of DAEs

0377-0427/00/$ - see front matter
PII: S0377-0427(00)00396-4

describing the whole circuit can be reduced significantly, so that time integration of the resulting system becomes feasible; see, e.g. [11,20,24,26] and the references given there. In recent years, there has been a lot of interest in generating suitable reduced-order models of linear subcircuits by means of Krylov-subspace methods, such as the Lanczos algorithm and the Arnoldi process. For a survey of these recent developments, we refer the reader to [13].

In this paper, we describe the use of Krylov-subspace methods for generating reduced-order models of systems of linear DAEs, such as the ones arising in circuit simulation. Particular emphasis is on projection techniques that, when applied to a passive circuit, preserve the passivity of the circuit. We stress that the methods discussed in this paper are not restricted to systems of DAEs arising in circuit simulation and that they can be applied to general time-invariant linear dynamical systems. However, the development of these methods was mostly motivated by the need for reduced-order modeling in circuit simulation.

The remainder of the paper is organized as follows. In Section 2, we briefly review the systems of DAEs that arise in circuit simulation, and we describe how reduced-order models of linear subcircuits can be employed to reduce the dimension of these systems. In Section 3, we introduce our notion of block Krylov subspaces and review the construction of basis vectors via Lanczos and Arnoldi algorithms. In Section 4, we define reduced-order models based on projection and describe their computation via Krylov-subspace methods. In Section 5, we discuss connections with Padé and Padé-type approximants. In Section 6, we establish results on the stability and passivity of reduced-order models obtained via projection. In Section 7, numerical results for two circuit examples are reported. Finally, in Section 8, we make some concluding remarks and mention a few open problems.

Throughout this article, we use boldface letters to denote vectors and matrices. Unless stated otherwise, vectors and matrices are allowed to have complex entries. As usual, $\overline{\boldsymbol{M}} = [\overline{m_{jk}}]$, $\boldsymbol{M}^{\mathrm{T}} = [m_{kj}]$, and $\boldsymbol{M}^{\mathrm{H}} = \overline{\boldsymbol{M}}^{\mathrm{T}} = [\overline{m_{kj}}]$ denote the complex conjugate, transpose, and the conjugate transpose, respectively, of the matrix $\boldsymbol{M} = [m_{jk}]$, and $\boldsymbol{M} \geqslant \boldsymbol{0}$ means that $\boldsymbol{M}$ is Hermitian positive semi-definite. The vector norm $\|\boldsymbol{x}\| := \sqrt{\boldsymbol{x}^{\mathrm{H}}\boldsymbol{x}}$ is always the Euclidean norm, and $\|\boldsymbol{M}\| := \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{M}\boldsymbol{x}\|$ is the corresponding induced matrix norm. We use $\boldsymbol{I}_n$ to denote the $n \times n$ identity matrix and $\boldsymbol{0}_{n\times m}$ to denote the $n \times m$ zero matrix; we will omit these indices whenever the actual dimensions of $\boldsymbol{I}$ and $\boldsymbol{0}$ 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. Circuit equations

In this section, we briefly describe the systems of DAEs that arise in circuit simulation and review how reduced-order modeling of linear subcircuits is employed in the numerical solution of such systems. For introductions to circuit simulation and overviews of typical simulation tasks, we refer the reader to [10,29,30].

2.1. General circuit equations

Electronic circuits are usually modeled as networks whose branches correspond to the circuit elements and whose nodes correspond to the interconnections of the circuit elements; see, e.g.

[10,29,30]. Such networks are characterized by three types of equations: *Kirchhoff's current law* (KCL), *Kirchhoff's voltage law* (KVL), and *branch constitutive relations* (BCRs). The unknowns in these equations are the currents through the branches of the network, the voltage drops along the branches, and the voltages at the nodes of the network. The KCLs and KVLs are linear algebraic equations that only depend on the topology of the circuit. The KCLs state that, at each node $\mathscr{N}$ of the network, the currents flowing in and out of $\mathscr{N}$ sum up to zero. The KVLs state that, for each closed loop $\mathscr{L}$ of the network, the voltage drops along $\mathscr{L}$ sum up to zero. The BCRs are equations that characterize the actual circuit elements. For example, the BCR of a linear resistor is Ohm's law. The BCRs are linear equations for simple devices, such as linear resistors, capacitors, and inductors, and they are nonlinear equations for more complex devices, such as diodes and transistors. Furthermore, in general, the BCRs involve first time derivatives of the unknowns, and thus they are first-order DAEs.

All the KCLs, KVLs, and BCRs characterizing a given circuit can be summarized as a system of first-order, in general nonlinear, DAEs of the form

$$\boldsymbol{f}(\hat{\boldsymbol{x}},t)+\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{q}(\hat{\boldsymbol{x}},t)=\boldsymbol{0}, \tag{1}$$

together with suitable initial conditions. Here, $\hat{\boldsymbol{x}}=\hat{\boldsymbol{x}}(t)$ is the unknown vector of circuit variables at time t, the vector-valued function $\boldsymbol{f}(\hat{\boldsymbol{x}},t)$ represents the contributions of nonreactive elements such as resistors, sources, etc., and the vector-valued function $(\mathrm{d}/\mathrm{d}t)\boldsymbol{q}(\hat{\boldsymbol{x}},t)$ represents the contributions of reactive elements such as capacitors and inductors. There are a number of established methods, such as sparse tableau, nodal formulation, and modified nodal analysis, for writing down the system (1); see, e.g. [30]. The vector functions $\hat{\boldsymbol{x}}$, $\boldsymbol{f}$, $\boldsymbol{q}$ in (1), as well as their dimension, $\hat{N}$, depend on the chosen formulation method. The most general method is sparse tableau, which consists of just listing all the KCLs, KVLs, and BCRs. The other formulation methods can be interpreted as starting from sparse tableau and eliminating some of the unknowns by using some of the KCL or KVL equations. For all the standard formulation methods, the dimension $\hat{N}$ is of the order of the number of devices in the circuit.

2.2. Linear subcircuits

Traditional circuit simulators are based on the numerical solution of the system of DAEs (1); see, e.g. [30]. However, the dimension of (1) can be so large that time integration of (1) is inefficient or even prohibitive. On the other hand, circuits often contain large linear subcircuits that can be well approximated by reduced-order models of much smaller dimension. By replacing the equations in (1) corresponding to such linear subcircuits by their respective reduced-order models, one obtains an approximate system of DAEs of much smaller dimension that can then be solved numerically by time integration. We now describe this process in more detail.

Let $\mathscr{C}_{\mathrm{l}}$ be a large linear subcircuit of a given circuit, and denote by $\mathscr{C}_{\mathrm{r}}$ the, in general nonlinear, remainder of the circuit. After a suitable reordering, the vector $\hat{\boldsymbol{x}}$ of circuit variables in (1) can be partitioned as follows:

$$\hat{\boldsymbol{x}}=\begin{bmatrix}\hat{\boldsymbol{x}}_{\mathrm{r}}\\ \boldsymbol{y}\\ \hat{\boldsymbol{x}}_{\mathrm{l}}\end{bmatrix}. \tag{2}$$

Here, $\hat{\boldsymbol{x}}_{\mathrm{r}}$ and $\hat{\boldsymbol{x}}_{\mathrm{l}}$ denote the circuit variables exclusive to $\mathscr{C}_{\mathrm{r}}$ and $\mathscr{C}_{\mathrm{l}}$, respectively, and $\boldsymbol{y}$ represents the variables shared by $\mathscr{C}_{\mathrm{r}}$ and $\mathscr{C}_{\mathrm{l}}$. Using the partitioning (2) and setting

$$\boldsymbol{x}_0 := \begin{bmatrix} \hat{\boldsymbol{x}}_{\mathrm{r}} \\ \boldsymbol{y} \end{bmatrix} \quad \text{and} \quad \boldsymbol{x} := \begin{bmatrix} \boldsymbol{y} \\ \hat{\boldsymbol{x}}_{\mathrm{l}} \end{bmatrix}, \tag{3}$$

the functions $\boldsymbol{f}$ and $\boldsymbol{q}$ in (1), after a suitable reordering of the equations in (1), can be expressed as follows:

$$\boldsymbol{f}(\hat{\boldsymbol{x}}, t) = \begin{bmatrix} \boldsymbol{f}_0(\boldsymbol{x}_0, t) \\ \boldsymbol{0}_{k\times 1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{\hat{N}-N\times 1} \\ \boldsymbol{G}\boldsymbol{x} \end{bmatrix}, \quad \boldsymbol{q}(\hat{\boldsymbol{x}}, t) = \begin{bmatrix} \boldsymbol{q}_0(\boldsymbol{x}_0, t) \\ \boldsymbol{0}_{k\times 1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{\hat{N}-N\times 1} \\ \boldsymbol{C}\boldsymbol{x} \end{bmatrix}. \tag{4}$$

Here, $\boldsymbol{f}_0$ and $\boldsymbol{q}_0$ represent the contributions of resistive and reactive elements from the subcircuit $\mathscr{C}_{\mathrm{r}}$, and the matrices $\boldsymbol{G}$ and $\boldsymbol{C}$ represent the contributions of resistive and reactive elements in the linear subcircuit $\mathscr{C}_{\mathrm{l}}$. In (4), without loss of generality, we have assumed that the vector-valued functions $\boldsymbol{f}_0$ and $\boldsymbol{q}_0$ have the same number of components, that the zero vectors below $\boldsymbol{f}_0$ and $\boldsymbol{q}_0$ have the same length, k, and that the matrices $\boldsymbol{G}$ and $\boldsymbol{C}$ are square and of the same size, $N \times N$; this can always be achieved by padding $\boldsymbol{f}_0$, $\boldsymbol{q}_0$, $\boldsymbol{G}$, and $\boldsymbol{C}$ with additional zeros, if necessary. Unless the subcircuit $\mathscr{C}_{\mathrm{l}}$ is completely decoupled form the remainder circuit $\mathscr{C}_{\mathrm{r}}$, we have $m := N - k > 0$. This means that, in (4), the last m components of the, in general nonlinear, functions $\boldsymbol{f}_0$ and $\boldsymbol{q}_0$ are connected with the first m components of the linear functions $\boldsymbol{G}\boldsymbol{x}$ and $\boldsymbol{C}\boldsymbol{x}$. By introducing an additional m-dimensional vector, $\boldsymbol{u} = \boldsymbol{u}(t)$, of circuit variables, these m connecting equations can be decoupled. Indeed, using (4), one readily verifies that the original system (1) is equivalent to the following system:

$$\boldsymbol{f}_0(\boldsymbol{x}_0, t) + \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{q}_0(\boldsymbol{x}_0, t) + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I}_m \end{bmatrix} \boldsymbol{u} = 0, \tag{5}$$

$$\boldsymbol{C}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} + \boldsymbol{G}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{I}_m \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{u}. \tag{6}$$

We remark that the additional variables $\boldsymbol{u}$ in (5) and (6) can be interpreted as interface signals between the subcircuits $\mathscr{C}_{\mathrm{r}}$ and $\mathscr{C}_{\mathrm{l}}$.

Let p denote the length of the vector $\boldsymbol{y}$ in the partitioning (2) of $\hat{\boldsymbol{x}}$, and set

$$\boldsymbol{B} := \begin{bmatrix} \boldsymbol{I}_m \\ \boldsymbol{0}_{N-m\times m} \end{bmatrix} \quad \text{and} \quad \boldsymbol{L} := \begin{bmatrix} \boldsymbol{I}_p \\ \boldsymbol{0}_{N-p\times p} \end{bmatrix}.$$

Note that, by (3), the matrix $\boldsymbol{L}^{\mathrm{H}} = \boldsymbol{L}^{\mathrm{T}}$ selects the subvector $\boldsymbol{y}$ from $\boldsymbol{x}$, i.e.

$$\boldsymbol{y} = \boldsymbol{L}^{\mathrm{H}}\boldsymbol{x}. \tag{7}$$

Eqs. (6) and (7) constitute a *linear dynamical system* of the form

$$\begin{aligned} \boldsymbol{C}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} &= -\boldsymbol{G}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}(t), \\ \boldsymbol{y}(t) &= \boldsymbol{L}^{\mathrm{H}}\boldsymbol{x}(t). \end{aligned} \tag{8}$$

In (8), in general, $\boldsymbol{C}$, $\boldsymbol{G} \in \mathbb{C}^{N\times N}$, $\boldsymbol{B} \in \mathbb{C}^{N\times m}$, and $\boldsymbol{L} \in \mathbb{C}^{N\times p}$ are given matrices, m and p denote the number of inputs and outputs, respectively, the components of the given vector-valued function $\boldsymbol{u} : [0,\infty) \mapsto \mathbb{C}^m$ are the inputs, and $\boldsymbol{y} : [0,\infty) \mapsto \mathbb{C}^p$ is the unknown function of outputs. The components of the unknown vector-valued function $\boldsymbol{x} : [0,\infty) \mapsto \mathbb{C}^N$ are the state variables, and N

is the state-space dimension. In general, the matrices $\boldsymbol{C}$ and $\boldsymbol{G}$ in (8) are allowed to be singular. However, we assume that $\boldsymbol{G}+s\boldsymbol{C}$ is a *regular matrix pencil*, i.e., $\boldsymbol{G}+s\boldsymbol{C}$ is singular only for finitely many values of $s \in \mathbb{C}$. This condition is always satisfied for linear dynamical systems (8) arising in circuit simulation.

A *reduced-order model* of (8) is a linear dynamical system of the same form as (8), but of smaller state-space dimension $n < N$. More precisely, a reduced-order model of state-space dimension n is of the form

$$\begin{aligned} \boldsymbol{C}_n \frac{\mathrm{d}\boldsymbol{z}}{\mathrm{d}t} &= -\boldsymbol{G}_n \boldsymbol{z} + \boldsymbol{B}_n \boldsymbol{u}(t), \\ \boldsymbol{y}(t) &= \boldsymbol{L}_n^{\mathrm{H}} \boldsymbol{z}(t). \end{aligned} \tag{9}$$

Here, $\boldsymbol{C}_n, \boldsymbol{G}_n \in \mathbb{C}^{n\times n}$, $\boldsymbol{B}_n \in \mathbb{C}^{n\times m}$, and $\boldsymbol{L}_n \in \mathbb{C}^{n\times p}$ are matrices that should be chosen such that the input–output mapping $\boldsymbol{u}(t) \mapsto \boldsymbol{y}(t)$ of (9) somehow approximates the input–output mapping of the original system (8); see Section 2.3 below.

After a suitable reduced-order model (9) for systems (6) and (7) has been determined, the linear part (6) of the circuit equations is replaced by the first set of equations in (9). The result is a reduced-order system of DAEs that represents an approximation to the original system (1); see, e.g. [9,26]. Provided that the size of $\mathscr{C}_l$ dominates that of $\mathscr{C}_r$, the approximate system has a much smaller state-space dimension than (1), and thus time integration by means of standard circuit simulators becomes feasible.

2.3. Transfer functions

Next, we introduce the so-called transfer function, which describes the input–output behavior of a linear dynamical system (9) in frequency domain.

For vector-valued functions $\boldsymbol{g}(t)$, $t \in [0,\infty)$, with $\boldsymbol{g}(0) = \boldsymbol{0}$, we denote by

$$\hat{\boldsymbol{g}}(s) = \int_0^\infty \boldsymbol{g}(t)\mathrm{e}^{-st}\,\mathrm{d}t, \quad s \in \mathbb{C}, \tag{10}$$

the (frequency-domain) *Laplace transform* of $\boldsymbol{g}$. We remark that in (10), the purely imaginary values $s = i\omega$, $\omega \geqslant 0$, correspond to the frequency ω; these are the physically meaningful values of the complex variable s.

We now assume, for simplicity, zero initial conditions $\boldsymbol{x}(0) = \boldsymbol{0}$ and $\boldsymbol{u}(0) = \boldsymbol{0}$ in (8). By applying (10) to the linear dynamical system (8), we obtain its frequency-domain formulation

$$\begin{aligned} s\boldsymbol{C}\hat{\boldsymbol{x}} &= -\boldsymbol{G}\hat{\boldsymbol{x}} + \boldsymbol{B}\hat{\boldsymbol{u}}(s), \\ \hat{\boldsymbol{y}}(s) &= \boldsymbol{L}^{\mathrm{H}}\hat{\boldsymbol{x}}(s). \end{aligned} \tag{11}$$

Eliminating $\hat{\boldsymbol{x}}$ in (11) results in the frequency-domain input–output relation $\hat{\boldsymbol{y}}(s) = \boldsymbol{H}(s)\hat{\boldsymbol{u}}(s)$, where $\boldsymbol{H}$, the *transfer function* of (8), is given by

$$\boldsymbol{H}(s) := \boldsymbol{L}^{\mathrm{H}}(\boldsymbol{G} + s\boldsymbol{C})^{-1}\boldsymbol{B}, \quad s \in \mathbb{C}. \tag{12}$$

Note that $\boldsymbol{H} : \mathbb{C} \mapsto (\mathbb{C} \cup \{\infty\})^{p\times m}$ is a matrix-valued rational function.

Similarly, the transfer function $\boldsymbol{H}_n : \mathbb{C} \mapsto (\mathbb{C} \cup \{\infty\})^{p\times m}$ of the reduced-order model (9) is given by

$$\boldsymbol{H}_n(s) := \boldsymbol{L}_n^{\mathrm{H}}(\boldsymbol{G}_n + s\boldsymbol{C}_n)^{-1}\boldsymbol{B}_n, \quad s \in \mathbb{C}. \tag{13}$$

In terms of transfer functions, the problem of constructing a reduced-order model (9) of size n that approximates the input–output behavior of (8) can be stated as follows: Determine the matrices $\boldsymbol{C}_n$, $\boldsymbol{G}_n$, $\boldsymbol{B}_n$, and $\boldsymbol{L}_n$ in (9) such that the reduced-order transfer function (13), $\boldsymbol{H}_n$, in some sense approximates the transfer function (12), $\boldsymbol{H}$, of the original linear dynamical system (8).

For systems (8) of small-to-moderate state-space dimension N, there is a variety of techniques to construct reduced-order models such that, in some appropriate norm, $\boldsymbol{H}_n$ approximates $\boldsymbol{H}$ over a whole range of values of s; see the references given in [13]. However, these techniques are usually not applicable to large-scale systems (8), such as the ones arising in circuit simulation. In the latter case, the matrices $\boldsymbol{C}$ and $\boldsymbol{G}$ in (8) are large and sparse. Note that, in view of (12), the evaluation of $\boldsymbol{H}(s_0)$ at even a single point $s_0 \in \mathbb{C}$ requires the solution of systems of linear equations with the large sparse coefficient matrix $\boldsymbol{G} + s_0\boldsymbol{C}$. Fortunately, the circuit matrices $\boldsymbol{C}$ and $\boldsymbol{G}$ are usually such that sparse Gaussian elimination can be employed to compute an LU factorization

$$\boldsymbol{G} + s_0\boldsymbol{C} = \boldsymbol{P}_1\boldsymbol{L}_0\boldsymbol{U}_0\boldsymbol{P}_2 \tag{14}$$

of the matrix $\boldsymbol{G} + s_0\boldsymbol{C}$. In (14), $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are permutation matrices that record pivoting for sparsity and numerical stability, $\boldsymbol{L}_0$ is a lower triangular matrix, and $\boldsymbol{U}_0$ is an upper triangular matrix. Pivoting for sparsity means that the original ordering of the rows and columns of $\boldsymbol{G} + s_0\boldsymbol{C}$ is changed so that potential fill-in in the factors $\boldsymbol{L}_0$ and $\boldsymbol{U}_0$ is reduced. For circuit matrices, typically very little fill-in occurs in $\boldsymbol{L}_0$ and $\boldsymbol{U}_0$, although this cannot be guaranteed in general. Once the factorization (14) is computed, the solution of the linear systems needed to evaluate $\boldsymbol{H}(s_0)$ is obtained by sparse backsolves.

Note that evaluating $\boldsymbol{H}(s_0)$ at several points s_0 would require the computation of a new factorization (14) for each new point s_0. Despite the limited fill-in for circuit matrices, the cost for factoring $\boldsymbol{G} + s_0\boldsymbol{C}$ is high enough that one tries to get away with computing a single factorization (14). This is the case for reduced-order models that are characterized by a matching of the leading terms in Taylor expansions of $\boldsymbol{H}$ and $\boldsymbol{H}_n$ about a given expansion point s_0. More precisely, such a reduced-order model of given size n is defined by

$$\boldsymbol{H}_n(s) = \boldsymbol{H}(s) + \mathcal{O}(s - s_0)^{q(n)}. \tag{15}$$

If $q(n)$ in (15) is as large as possible, then $\boldsymbol{H}_n$ is an *nth matrix-Padé approximant* of $\boldsymbol{H}$; see, e.g. [5]. In Section 5, we will also discuss certain *matrix-Padé-type approximants* for which $q(n)$ is not maximal.

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. [16,17,20,24].

The equations describing linear RLC subcircuits are of the form (8). Furthermore, the equations can be formulated such that the matrices in (8) exhibit certain symmetries; see [15,17]. More precisely, the $N \times N$ matrices $\boldsymbol{G}$ and $\boldsymbol{C}$ are real and symmetric, and have the following block structure:

$$\boldsymbol{G} = \boldsymbol{G}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{G}_{11} & \boldsymbol{G}_{12} \\ \boldsymbol{G}_{12}^{\mathrm{T}} & \boldsymbol{0} \end{bmatrix} \quad \text{and} \quad \boldsymbol{C} = \boldsymbol{C}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{C}_{22} \end{bmatrix}. \tag{16}$$

Here, the submatrices $\boldsymbol{G}_{11}, \boldsymbol{C}_{11} \in \mathbb{R}^{N_1 \times N_1}$ and $\boldsymbol{C}_{22} \in \mathbb{R}^{N_2 \times N_2}$ are symmetric positive semi-definite, and $N = N_1 + N_2$. Note that, except for the special case $N_2 = 0$, the matrices $\boldsymbol{G}$ and $\boldsymbol{C}$ are indefinite. The special case $N_2 = 0$ arises for RC subcircuits that contain only resistors and capacitors, but no inductors.

If the RLC subcircuit is viewed as an m-terminal component with $m = p$ inputs and outputs, then the matrices $\boldsymbol{B}$ and $\boldsymbol{L}$ in (8) are identical and of the form

$$\boldsymbol{B} = \boldsymbol{L} = \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{0}_{N_2 \times m} \end{bmatrix} \quad \text{with } \boldsymbol{B}_1 \in \mathbb{R}^{N_1 \times m}. \tag{17}$$

For such an m-terminal RLC subcircuit, in view of (16) and (17), the transfer function (12) reduces to

$$\boldsymbol{H}(s) = \boldsymbol{B}^{\mathrm{T}}(\boldsymbol{G} + s\boldsymbol{C})^{-1}\boldsymbol{B} \quad \text{where } \boldsymbol{G} = \boldsymbol{G}^{\mathrm{T}},\ \boldsymbol{C} = \boldsymbol{C}^{\mathrm{T}}. \tag{18}$$

We call a transfer function $\boldsymbol{H}$ *symmetric* if it is of the form (18) with real matrices $\boldsymbol{G}$, $\boldsymbol{C}$, and $\boldsymbol{B}$. For symmetric transfer functions, we will always assume that the expansion point s_0 in (15) is chosen to be real:

$$s_0 \in \mathbb{R} \quad \text{if } \boldsymbol{H} \text{ is symmetric.} \tag{19}$$

The condition (19) is necessary in order to generate passive reduced-order models of symmetric transfer functions.

We will also use the following nonsymmetric formulation of (18). Let $\boldsymbol{J}$ be the block matrix

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{I}_{N_1} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I}_{N_2} \end{bmatrix}. \tag{20}$$

Note that, by (17) and (20), we have $\boldsymbol{B} = \boldsymbol{J}\boldsymbol{B}$. Using this relation, as well as (16), we can rewrite (18) as follows:

$$\boldsymbol{H}(s) = \boldsymbol{B}^{\mathrm{T}}(\boldsymbol{J}\boldsymbol{G} + s\boldsymbol{J}\boldsymbol{C})^{-1}\boldsymbol{B}, \tag{21}$$

where

$$\boldsymbol{J}\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_{11} & \boldsymbol{G}_{12} \\ -\boldsymbol{G}_{12}^{\mathrm{T}} & \boldsymbol{0} \end{bmatrix} \quad \text{and} \quad \boldsymbol{J}\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_{22} \end{bmatrix}.$$

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

$$\boldsymbol{J}\boldsymbol{G} + (\boldsymbol{J}\boldsymbol{G})^{\mathrm{T}} \geqslant \boldsymbol{0} \quad \text{and} \quad \boldsymbol{J}\boldsymbol{C} \geqslant \boldsymbol{0}. \tag{22}$$

3. Basis vectors for block Krylov subspaces

In this section, we introduce our notion of block Krylov subspaces for multiple starting vectors. We also review variants of the Arnoldi and Lanczos algorithms for generating basis vectors for block Krylov subspaces.

3.1. Reduction to one matrix

Let $s_0 \in \mathbb{C}$ be the expansion point that is to be used in the characterization (15) of the reduced-order transfer function $\boldsymbol{H}_n$. The only assumption on s_0 is that the matrix $\boldsymbol{G} + s_0\boldsymbol{C}$ be nonsingular; this guarantees that s_0 is not a pole of the original transfer function (12), $\boldsymbol{H}$.

An approximate transfer function $\boldsymbol{H}_n$ satisfying (15) could be obtained by first explicitly computing the leading $q(n)$ Taylor coefficients of the expansion of $\boldsymbol{H}$ about s_0 and then generating $\boldsymbol{H}_n$ from these coefficients; see, e.g. [25]. However, any approach based on explicitly computing the Taylor coefficients of $\boldsymbol{H}$ is inherently numerically unstable; see [8]. 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 $\boldsymbol{H}$, but in a more stable manner.

Before block Krylov-subspace methods can be employed, the two matrices $\boldsymbol{G}$ and $\boldsymbol{C}$ in the definition (12) of $\boldsymbol{H}$ have to be reduced to a single matrix, denoted by $\boldsymbol{A}$ in the sequel. This can be done by rewriting (12) as follows:

$$\boldsymbol{H}(s) = \boldsymbol{L}^{\mathrm{H}}(\boldsymbol{I} + (s - s_0)\boldsymbol{A})^{-1}\boldsymbol{R}, \tag{23}$$

where

$$\boldsymbol{A} := (\boldsymbol{G} + s_0\boldsymbol{C})^{-1}\boldsymbol{C} \quad \text{and} \quad \boldsymbol{R} := (\boldsymbol{G} + s_0\boldsymbol{C})^{-1}\boldsymbol{B}.$$

Although $\boldsymbol{G}$ and $\boldsymbol{C}$ are sparse matrices, in general, the matrix $\boldsymbol{A}$ in (23) is a dense matrix. However, block Krylov-subspace methods involve $\boldsymbol{A}$ only in the form of matrix-vector products $\boldsymbol{Av}$ and possibly $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{w}$. To efficiently compute these products, one never needs to form $\boldsymbol{A}$ explicitly. Instead, one uses the sparse factorization (14) of $\boldsymbol{G} + s_0\boldsymbol{C}$. Each matrix–vector product $\boldsymbol{Av}$ then requires one multiplication with the sparse matrix $\boldsymbol{C}$ and two backsolves with the sparse triangular matrices $\boldsymbol{L}_0$ and $\boldsymbol{U}_0$ from (14). Similarly, $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{w}$ requires one multiplication with $\boldsymbol{C}^{\mathrm{H}}$ and two backsolves with the sparse triangular matrices $\boldsymbol{L}_0^{\mathrm{H}}$ and $\boldsymbol{U}_0^{\mathrm{H}}$.

3.2. Block Krylov subspaces

Next, we introduce block Krylov subspaces. The proper definition of these subspaces is necessarily quite involved, and the reader may ask if block Krylov subspaces could not be avoided altogether by using standard Krylov subspaces induced by single vectors instead. For example, one can generate scalar approximations for all the $p \cdot m$ coefficient functions of the $p \times m$-matrix-valued transfer function $\boldsymbol{H}$ via suitable basis vectors for $m + p$ standard Krylov subspaces. However, the resulting approximation is not a matrix-Padé approximant of $\boldsymbol{H}$, and in fact, one can show that, in order to obtain an approximation of the same quality as the matrix-Padé approximant, at least $\lfloor (m + p)/2 \rfloor$ times more computational work is required compared to computing a matrix-Padé approximant. Therefore, the use of block Krylov subspaces results in much more efficient reduced-order modeling techniques than those based on standard Krylov subspaces.

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

$$\boldsymbol{R} = [\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \cdots \quad \boldsymbol{r}_m] \in \mathbb{C}^{N\times m} \tag{24}$$

be a given matrix of m *right starting vectors*, $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_m$. Before we introduce block Krylov subspaces induced by $\boldsymbol{A}$ and $\boldsymbol{R}$, we briefly review the standard case $m = 1$ of a single starting vector $\boldsymbol{r} = \boldsymbol{r}_1$. In this case, the usual *nth Krylov subspace* (induced by $\boldsymbol{A}$ and $\boldsymbol{r}$) is given by

$$\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{r}) := \mathrm{span}\{\boldsymbol{r}, \boldsymbol{Ar}, \boldsymbol{A}^2\boldsymbol{r}, \ldots, \boldsymbol{A}^{n-1}\boldsymbol{r}\}. \tag{25}$$

Let n_0 be defined as the largest possible integer n such that in (25), all the Krylov vectors, $\boldsymbol{A}^{j-1}\boldsymbol{r}$, $1 \leqslant j \leqslant n - 1$, are linearly independent. Note that $n_0 \leqslant N$. By the definition of n_0, the nth Krylov

subspace (25) has dimension n if $1 \leqslant n \leqslant n_0$ and dimension n_0 if $n > n_0$. Moreover, $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{r}) = \mathscr{K}_{n_0}(\boldsymbol{A}, \boldsymbol{r})$ for all $n > n_0$. Thus, $\mathscr{K}_{n_0}(\boldsymbol{A}, \boldsymbol{r})$ is the largest possible Krylov subspace (induced by $\boldsymbol{A}$ and $\boldsymbol{r}$), and we call the Krylov sequence $\boldsymbol{r}, \boldsymbol{A}\boldsymbol{r}, \boldsymbol{A}^2\boldsymbol{r}, \ldots, \boldsymbol{A}^{n-1}\boldsymbol{r}$ *exhausted* if $n > n_0$.

In the general case of $m \geqslant 1$ starting vectors (24), the situation is more complicated; we refer the reader to the discussion in [1]. The main difficulty is that in contrast to the case $m = 1$, linear independence of the columns in the block Krylov sequence,

$$\boldsymbol{R}, \boldsymbol{A}\boldsymbol{R}, \boldsymbol{A}^2\boldsymbol{R}, \ldots, \boldsymbol{A}^{j-1}\boldsymbol{R}, \ldots, \tag{26}$$

is lost only gradually in general. More precisely, if the jth block, $\boldsymbol{A}^{j-1}\boldsymbol{R}$, contains a column that is linearly dependent on columns to its left in (26), then, in general, not all the columns of $\boldsymbol{A}^{j-1}\boldsymbol{R}$ 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 $\boldsymbol{R}, \boldsymbol{A}\boldsymbol{R}, \boldsymbol{A}^2\boldsymbol{R}, \ldots, \boldsymbol{A}^{j-1}\boldsymbol{R}$ is exhausted. As a result, in general, the construction of suitable basis vectors for the subspaces spanned by the columns of (26) needs to be continued even after a linear-dependent column has been found in (26). However, a proper handling of such linear-dependent columns requires that the column itself and all its successive $\boldsymbol{A}$-multiples need to be deleted. Formally, this can be done as follows. By scanning the columns of the matrices in (26) from left to right and deleting each column that is linearly dependent on earlier columns, we obtain the *deflated* block Krylov sequence

$$\boldsymbol{R}_1, \boldsymbol{A}\boldsymbol{R}_2, \boldsymbol{A}^2\boldsymbol{R}_3, \ldots, \boldsymbol{A}^{j_{\max}-1}\boldsymbol{R}_{j_{\max}}. \tag{27}$$

This process of deleting linearly dependent vectors is referred to as *exact deflation* in the following. In (27), for each $j = 1, 2, \ldots, j_{\max}$, $\boldsymbol{R}_j$ is a submatrix of $\boldsymbol{R}_{j-1}$, with $\boldsymbol{R}_j \neq \boldsymbol{R}_{j-1}$ if, and only if, deflation occurred within the jth Krylov block $\boldsymbol{A}^{j-1}\boldsymbol{R}$ in (26). Here, for $j = 1$, we set $\boldsymbol{R}_0 = \boldsymbol{R}$. Denoting by m_j the number of columns of $\boldsymbol{R}_j$, we thus have

$$m \geqslant m_1 \geqslant m_2 \geqslant \cdots \geqslant m_{j_{\max}} \geqslant 1. \tag{28}$$

By construction, the columns of the matrices (27) are linearly independent, and for each n, the subspace spanned by the first n of these columns is called the *nth block Krylov subspace* (induced by $\boldsymbol{A}$ and $\boldsymbol{R}$). In the following, we denote the nth block Krylov subspace by $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$. For later use, we remark that for

$$n = m_1 + m_2 + \cdots + m_j \quad \text{where } 1 \leqslant j \leqslant j_{\max}, \tag{29}$$

the nth block Krylov subspace is given by

$$\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R}) = \operatorname{Colspan}\{\boldsymbol{R}_1, \boldsymbol{A}\boldsymbol{R}_2, \boldsymbol{A}^2\boldsymbol{R}_3, \ldots, \boldsymbol{A}^{j-1}\boldsymbol{R}_j\}. \tag{30}$$

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

$$\boldsymbol{L} = [\boldsymbol{l}_1 \quad \boldsymbol{l}_2 \quad \cdots \quad \boldsymbol{l}_p] \in \mathbb{C}^{N \times p}. \tag{31}$$

Applying the above construction to $\boldsymbol{A}^{\mathrm{H}}$ and $\boldsymbol{L}$, the *nth block Krylov subspace* (induced by $\boldsymbol{A}^{\mathrm{H}}$ and $\boldsymbol{L}$), $\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}}, \boldsymbol{L})$, is defined as the subspace spanned by the first n columns of the *deflated* block Krylov sequence

$$\boldsymbol{L}_1, \boldsymbol{A}^{\mathrm{H}}\boldsymbol{L}_2, (\boldsymbol{A}^{\mathrm{H}})^2\boldsymbol{L}_3, \ldots, (\boldsymbol{A}^{\mathrm{H}})^{k_{\max}-1}\boldsymbol{L}_{k_{\max}}. \tag{32}$$

Denoting by p_k the number of columns of $\boldsymbol{L}_k$, we have

$$p \geqslant p_1 \geqslant p_2 \geqslant \cdots \geqslant p_{k_{\max}} \geqslant 1. \tag{33}$$

Note that for

$$n = p_1 + p_2 + \cdots + p_k \quad \text{where } 1 \leqslant k \leqslant k_{\max},$$

we have

$$\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}}, \boldsymbol{L}) = \mathrm{Colspan}\{\boldsymbol{L}_1, \boldsymbol{A}^{\mathrm{H}}\boldsymbol{L}_2, (\boldsymbol{A}^{\mathrm{H}})^2\boldsymbol{L}_3, \ldots, (\boldsymbol{A}^{\mathrm{H}})^{k-1}\boldsymbol{L}_k\}.$$

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 $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$ and $\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}}, \boldsymbol{L})$ 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*. In Sections 3.4 and 3.5 below, we describe how exact and inexact deflation can be built easily into Arnoldi- and Lanczos-type algorithms for multiple starting vectors. 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 $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$ induced by the matrices $\boldsymbol{A}$ and $\boldsymbol{R}$ defined in (23).

Lemma 1. *Let $\boldsymbol{G}$, $\boldsymbol{C}$, $\boldsymbol{B}$ be the matrix triplet used in the definition of the matrices $\boldsymbol{A}$ and $\boldsymbol{R}$ in* (23), *and let $\boldsymbol{J}$ be any nonsingular matrix of the same size as $\boldsymbol{A}$. Then the matrix triplets $\boldsymbol{G}$, $\boldsymbol{C}$, $\boldsymbol{B}$ and $\boldsymbol{JG}$, $\boldsymbol{JC}$, $\boldsymbol{JB}$ lead to the same nth block Krylov subspace $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$.*

Proof. By (23), we have

$$\boldsymbol{A} = (\boldsymbol{G} + s_0\boldsymbol{C})^{-1}\boldsymbol{C} = (\boldsymbol{JG} + s_0\boldsymbol{JC})^{-1}\boldsymbol{JC},$$
$$\boldsymbol{R} = (\boldsymbol{G} + s_0\boldsymbol{C})^{-1}\boldsymbol{B} = (\boldsymbol{JG} + s_0\boldsymbol{JC})^{-1}\boldsymbol{JB}.$$

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

3.3. Basis vectors

The columns of the deflated block Krylov sequences (27) and (32), which are used in the above definitions of $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$ and $\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}}, \boldsymbol{L})$, respectively, tend to be almost linearly dependent even for moderate values of n. Therefore, they should not be used in numerical computations. Instead, we construct other suitable basis vectors.

In the following,

$$\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n \in \mathbb{C}^N \tag{34}$$

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

$$\mathrm{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\} = \mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R}).$$

The $N \times n$ matrix

$$\boldsymbol{V}_n := [\boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \cdots \quad \boldsymbol{v}_n] \tag{35}$$

whose columns are the basis vectors (34) is called a *basis matrix* for $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$.

Similarly,

$$\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n \in \mathbb{C}^N \tag{36}$$

denotes a set of basis vectors for $\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}}, \boldsymbol{L})$, i.e.,

$$\operatorname{span}\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n\} = \mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}}, \boldsymbol{L}).$$

The $N \times n$ matrix

$$\boldsymbol{W}_n := [\boldsymbol{w}_1 \quad \boldsymbol{w}_2 \quad \cdots \quad \boldsymbol{w}_n] \tag{37}$$

is called a *basis matrix* for $\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}}, \boldsymbol{L})$.

We stress that even though (34) and (36) are basis vectors of block Krylov subspaces, the algorithms discussed in this paper generate (34) and (36) 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 [1]. 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. In fact, all that is required is checking if a suitable candidate vector for the next basis vector is the zero vector (for exact deflation) or close to the zero vector (for inexact deflation). Second, for Lanczos-type methods, which simultaneously construct basis vectors (34) and (36) for $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$ and $\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}}, \boldsymbol{L})$, respectively, only the vector-wise construction allows the treatment of the general case where the block sizes (28) and (33) of the deflated block Krylov sequences (27) and (32) are not necessarily the same; for a detailed discussion, we refer the reader to [1].

3.4. Arnoldi basis

The classical Arnoldi process [3] generates orthonormal basis vectors for the sequence of Krylov subspaces $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{r})$, $n \geqslant 1$, induced by $\boldsymbol{A}$ and a single starting vector $\boldsymbol{r}$. In this subsection, we state an Arnoldi-type algorithm that extends the classical algorithm to block-Krylov subspaces $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$, $n \geqslant 1$.

Like the classical Arnoldi process, the algorithm constructs the basis vectors (34) to be orthonormal. In terms of the basis matrix (35), this orthonormality can be expressed as follows:

$$\boldsymbol{V}_n^{\mathrm{H}} \boldsymbol{V}_n = \boldsymbol{I}.$$

In addition to (34), the algorithm produces the so-called *candidate vectors*,

$$\hat{\boldsymbol{v}}_{n+1}, \hat{\boldsymbol{v}}_{n+2}, \ldots, \hat{\boldsymbol{v}}_{n+m_c}, \tag{38}$$

for the next m_c basis vectors $\boldsymbol{v}_{n+1}, \boldsymbol{v}_{n+2}, \ldots, \boldsymbol{v}_{n+m_c}$. Here, $m_c = m_c(n)$ is the number m of columns in the starting block (24), $\boldsymbol{R}$, reduced by the number of exact and inexact deflations that have occurred so far. The candidate vectors (38) satisfy the orthogonality relation

$$\boldsymbol{V}_n^{\mathrm{H}} [\hat{\boldsymbol{v}}_{n+1} \quad \hat{\boldsymbol{v}}_{n+2} \quad \cdots \quad \hat{\boldsymbol{v}}_{n+m_c}] = \boldsymbol{0}.$$

Due to the vector-wise construction of (34) and (38), detection of necessary deflation and the actual deflation becomes trivial. In fact, essentially the same proof as given in [1] for the case of a Lanczos-type algorithm can be used to show that exact deflation at step n of the Arnoldi-type process occurs if, and only if, $\hat{\boldsymbol{v}}_n = \boldsymbol{0}$. Similarly, inexact deflation occurs if, and only if, $\|\hat{\boldsymbol{v}}_n\| \approx 0$, but $\hat{\boldsymbol{v}}_n \neq \boldsymbol{0}$. Therefore, in the algorithm, one checks if

$$\|\hat{\boldsymbol{v}}_n\| \leqslant \texttt{dtol}, \tag{39}$$

where $\texttt{dtol} \geqslant 0$ is a suitably chosen deflation tolerance. If (39) is satisfied, then $\hat{\boldsymbol{v}}_n$ is deflated by deleting $\hat{\boldsymbol{v}}_n$, shifting the indices of all remaining candidate vectors by -1, and setting $m_c = m_c - 1$. If this results in $m_c = 0$, then the block-Krylov subspace is exhausted and the algorithm is stopped. Otherwise, the deflation procedure is repeated until a vector $\hat{\boldsymbol{v}}_n$ with $\|\hat{\boldsymbol{v}}_n\| > \texttt{dtol}$ is found. This vector is then turned into $\boldsymbol{v}_n$ by normalizing it to have Euclidean norm 1.

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

Algorithm 1 (Construction of Arnoldi basis for $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$.)**.**

(0) *Set* $\hat{\boldsymbol{v}}_i = \boldsymbol{r}_i$ *for* $i = 1, 2, \ldots, m$.
Set $m_c = m$.
For $n = 1, 2, \ldots$, *do*:
(1) *Compute* $\|\hat{\boldsymbol{v}}_n\|$ *and check if the deflation criterion* (39) *is fulfilled.*
If yes, $\hat{\boldsymbol{v}}_n$ *is deflated by doing the following*:
Set $m_c = m_c - 1$. *If* $m_c = 0$, *set* $n = n - 1$ *and stop.*
Set $\hat{\boldsymbol{v}}_i = \hat{\boldsymbol{v}}_{i+1}$ *for* $i = n, n+1, \ldots, n + m_c - 1$.
Return to Step (1).
(2) *Set* $t_{n,n-m_c} = \|\hat{\boldsymbol{v}}_n\|$ *and* $\boldsymbol{v}_n = \hat{\boldsymbol{v}}_n / t_{n,n-m_c}$.
(3) *Compute* $\hat{\boldsymbol{v}}_{n+m_c} = \boldsymbol{A}\boldsymbol{v}_n$.
(4) *For* $i = 1, 2, \ldots, n$ *do*:
Set $t_{i,n} = \boldsymbol{v}_i^{\mathrm{H}} \hat{\boldsymbol{v}}_{n+m_c}$ *and* $\hat{\boldsymbol{v}}_{n+m_c} = \hat{\boldsymbol{v}}_{n+m_c} - \boldsymbol{v}_i t_{i,n}$.
(5) *For* $i = n - m_c + 1, n - m_c + 2, \ldots, n - 1$ *do*:
Set $t_{n,i} = \boldsymbol{v}_n^{\mathrm{H}} \hat{\boldsymbol{v}}_{i+m_c}$ *and* $\hat{\boldsymbol{v}}_{i+m_c} = \hat{\boldsymbol{v}}_{i+m_c} - \boldsymbol{v}_n t_{n,i}$.

Remark 2. If $\texttt{dtol} = 0$ in (39), then Algorithm 1 performs only exact deflation.

Remark 3. Other block-Arnoldi algorithms (all without deflation though) can be found in [28, Section 6.12].

After n passes through the main loop, Algorithm 1 has constructed the first n basis vectors (34) and the candidate vectors (38) for the next m_c basis vectors. In terms of the basis matrix (35), $\boldsymbol{V}_n$, the recurrences used to generate all these vectors can be written compactly in matrix format. To this end, we collect the recurrence coefficients computed during the first $n = m_1$ and $n \geqslant 1$ passes through the main loop of Algorithm 1 in the matrices

$$\boldsymbol{\rho} := [t_{i,l-m}]_{\substack{i=1,2,\ldots,m_1 \\ l=1,2,\ldots,m}} \quad \text{and} \quad \boldsymbol{T}_n := [t_{i,l}]_{\substack{i=1,2,\ldots,n \\ l=1,2,\ldots,n}}, \tag{40}$$

respectively. Here, m_1 is the integer given by (27) and (28). Moreover, in (40), all elements $t_{i,l-m}$ and $t_{i,l}$ that are not explicitly defined in Algorithm 1 are set to be zero. The compact statement of the recurrences used in Algorithm 1 is now as follows. For $n \geqslant 1$, we have

$$\boldsymbol{A}\boldsymbol{V}_n = \boldsymbol{V}_n\boldsymbol{T}_n + [\boldsymbol{0} \quad \cdots \quad \boldsymbol{0} \quad \hat{\boldsymbol{v}}_{n+1} \quad \hat{\boldsymbol{v}}_{n+2} \quad \cdots \quad \hat{\boldsymbol{v}}_{n+m_c}]. \tag{41}$$

Furthermore, for $n = m_1$, we have the relation

$$\boldsymbol{R} = \boldsymbol{V}_{m_1}\boldsymbol{\rho}, \tag{42}$$

which reflects the fact that the first m_1 basis vectors are obtained by orthonormalizing the columns of the matrix (24), $\boldsymbol{R}$. In (41) and (42), we assumed that only exact deflations are performed. If both exact and inexact deflations are performed, then an additional matrix term, say $\hat{\boldsymbol{V}}_n^{\text{defl}}$, appears on the right-hand side of (41), respectively (42). The only non-zero columns of $\hat{\boldsymbol{V}}_n^{\text{defl}}$ are those non-zero vectors that satisfied the deflation check (39). Since at any stage of Algorithm 1, at most $m - m_c = m - m_c(n)$ vectors have been deflated, the additional matrix term is small in the sense that

$$\|\hat{\boldsymbol{V}}_n^{\text{defl}}\| \leqslant \texttt{dtol}\sqrt{m - m_c(n)}.$$

3.5. Lanczos basis

The classical Lanczos process [21] generates two sequences of basis vectors (34) and (36) that span the Krylov subspaces $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{r})$ and $\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}}, \boldsymbol{l})$, respectively, where $\boldsymbol{r}$ and $\boldsymbol{l}$ are single starting vectors. In [1], a Lanczos-type method was presented that extends the classical algorithm to block-Krylov subspaces $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$ and $\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}}, \boldsymbol{L})$ with blocks $\boldsymbol{R}$ and $\boldsymbol{L}$ of multiple right and left starting vectors (24) and (31). Such a Lanczos-type method is necessarily quite involved, and in order to keep this paper reasonably short, here we only state the governing equations that underlie the algorithm. For a complete listing of the actual algorithm, we refer the reader to [13, Algorithm 9.2].

Like the classical Lanczos process, the extended algorithm constructs the basis vectors (34) and (36) to be biorthogonal. In terms of the associated basis matrices (35) and (37), the biorthogonality can be expressed as follows:

$$\boldsymbol{W}_n^{\mathrm{H}}\boldsymbol{V}_n = \boldsymbol{\Delta}_n := \operatorname{diag}(\delta_1, \delta_2, \ldots \delta_n). \tag{43}$$

Here, for simplicity, we have assumed that no look-ahead steps are necessary. This implies that $\boldsymbol{\Delta}_n$ is a diagonal matrix, as stated in (43), and that all diagonal entries of $\boldsymbol{\Delta}_n$ are nonzero. If look-ahead steps occur, then $\boldsymbol{\Delta}_n$ is a block-diagonal matrix; see [1] for further details. In addition to (34) and (36), the algorithm constructs the candidate vectors

$$\hat{\boldsymbol{v}}_{n+1}, \hat{\boldsymbol{v}}_{n+2}, \ldots, \hat{\boldsymbol{v}}_{n+m_c} \quad \text{and} \quad \hat{\boldsymbol{w}}_{n+1}, \hat{\boldsymbol{w}}_{n+2}, \ldots, \hat{\boldsymbol{w}}_{n+p_c} \tag{44}$$

for the next m_c basis vectors $\boldsymbol{v}_{n+1}, \boldsymbol{v}_{n+2}, \ldots, \boldsymbol{v}_{n+m_c}$ and the next p_c basis vectors $\boldsymbol{w}_{n+1}, \boldsymbol{w}_{n+2}, \ldots, \boldsymbol{w}_{n+p_c}$, respectively. Here, as in Section 3.4, $m_c = m_c(n)$ is the number m of columns in the right starting block (24), $\boldsymbol{R}$, reduced by the number of exact and inexact deflations of candidate vectors $\hat{\boldsymbol{v}}_n$ that have occurred so far. Analogously, $p_c = p_c(n)$ is the number p of columns in the left starting block (31), $\boldsymbol{L}$, reduced by the number of exact and inexact deflations of candidate vectors $\hat{\boldsymbol{w}}_n$ that have

occurred so far. Similar to (39) a deflation of the candidate vector $\hat{\boldsymbol{v}}_n$, respectively $\hat{\boldsymbol{w}}_n$, is performed if, and only if,

$$\|\hat{\boldsymbol{v}}_n\| \leqslant \texttt{dtol}, \quad \text{respectively } \|\hat{\boldsymbol{w}}_n\| \leqslant \texttt{dtol}, \tag{45}$$

where $\texttt{dtol} \geqslant 0$ is a suitably chosen deflation tolerance. If $\texttt{dtol} = 0$, then (45) only checks for exact deflation.

The candidate vectors (44) are constructed to satisfy the following biorthogonality relations:

$$\begin{aligned} \boldsymbol{W}_n^{\mathrm{H}}[\hat{\boldsymbol{v}}_{n+1} \quad \hat{\boldsymbol{v}}_{n+2} \quad \cdots \quad \hat{\boldsymbol{v}}_{n+m_c}] &= \boldsymbol{0}, \\ \boldsymbol{V}_n^{\mathrm{H}}[\hat{\boldsymbol{w}}_{n+1} \quad \hat{\boldsymbol{w}}_{n+2} \quad \cdots \quad \hat{\boldsymbol{w}}_{n+p_c}] &= \boldsymbol{0}. \end{aligned} \tag{46}$$

The actual recurrences used to generate the basis vectors (34) and (36) and the candidate vectors (44) can be summarized compactly in matrix form. For simplicity, we again assume that only exact deflation is performed. Then, for $n \geqslant 1$, we have

$$\begin{aligned} \boldsymbol{A}\boldsymbol{V}_n &= \boldsymbol{V}_n\boldsymbol{T}_n + [\boldsymbol{0} \quad \cdots \quad \boldsymbol{0} \quad \hat{\boldsymbol{v}}_{n+1} \quad \hat{\boldsymbol{v}}_{n+2} \quad \cdots \quad \hat{\boldsymbol{v}}_{n+m_c}], \\ \boldsymbol{A}^{\mathrm{H}}\boldsymbol{W}_n &= \boldsymbol{W}_n\tilde{\boldsymbol{T}}_n + [\boldsymbol{0} \quad \cdots \quad \boldsymbol{0} \quad \hat{\boldsymbol{w}}_{n+1} \quad \hat{\boldsymbol{w}}_{n+2} \quad \cdots \quad \hat{\boldsymbol{w}}_{n+p_c}]. \end{aligned} \tag{47}$$

Moreover, for $n = m_1$, respectively $n = p_1$, we have the relations

$$\boldsymbol{V}_{m_1}\boldsymbol{\rho} = \boldsymbol{R}, \quad \text{respectively } \boldsymbol{W}_{p_1}\boldsymbol{\eta} = \boldsymbol{L}, \tag{48}$$

which summarize the recurrences for processing the starting blocks $\boldsymbol{R}$ and $\boldsymbol{L}$. We note that the matrices $\boldsymbol{T}_n$ and $\tilde{\boldsymbol{T}}_n$ in (47) essentially encode the same information. In fact, by pre-multiplying the first and second relation in (47) by $\boldsymbol{W}_n$ and $\boldsymbol{V}_n$ and by using (43) and (46), it follows that

$$\boldsymbol{W}_n^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_n = \boldsymbol{\Delta}_n\boldsymbol{T}_n = \tilde{\boldsymbol{T}}_n^{\mathrm{H}}\boldsymbol{\Delta}_n. \tag{49}$$

In particular, (49) implies that $\tilde{\boldsymbol{T}}_n^{\mathrm{H}} = \boldsymbol{\Delta}_n\boldsymbol{T}_n\boldsymbol{\Delta}_n^{-1}$.

Finally, we note that for symmetric transfer functions (18), such as the ones describing RLC subcircuits, the Lanczos-type method sketched in this section can exploit the symmetry inherent in (18). Indeed, in this case, the Lanczos basis vectors (34) and (36) are connected as follows:

$$\boldsymbol{w}_n = \gamma_n(\boldsymbol{G} + s_0\boldsymbol{C})\boldsymbol{v}_n \quad \text{for all } n. \tag{50}$$

Here, $\gamma_n \neq 0$ are suitable normalization factors. In view of (50), only the vectors (34) need to be generated. This results in a *symmetric* variant of the Lanczos-type method that requires only half the computational work and storage of the general case; see [15–17] for further details. For later use, we note that for symmetric transfer functions (18), the coefficient matrices in (48) can be chosen to be identical:

$$\boldsymbol{\rho} = \boldsymbol{\eta} \in \mathbb{R}^{m_1 \times m}. \tag{51}$$

In fact, by (23), (18), and (19), $(\boldsymbol{G} + s_0\boldsymbol{C})\boldsymbol{R} = \boldsymbol{B} = \boldsymbol{L}$ and all these matrices are real. In view of (48) and (50), this implies (51).

4. Reduced-order models based on projection

In this section, we introduce two reduced-order models based on a one-sided projection onto $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$, respectively a two-sided projection onto $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$ and $\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}}, \boldsymbol{L})$.

4.1. Projection onto $\mathscr{K}_n(\boldsymbol{A},\boldsymbol{R})$

Let $\boldsymbol{V}_n \in \mathbb{C}^{N\times n}$ be a given matrix with full column rank n. By simply projecting the matrices $\boldsymbol{G}$, $\boldsymbol{C}$, $\boldsymbol{B}$, and $\boldsymbol{L}$ in the original linear dynamical system (8) onto the subspace spanned by the columns of $\boldsymbol{V}_n$, we obtain a reduced-order model (9) with matrices $\boldsymbol{G}_n$, $\boldsymbol{C}_n$, $\boldsymbol{B}_n$, and $\boldsymbol{L}_n$ given by

$$\boldsymbol{G}_n := \boldsymbol{V}_n^{\mathrm{H}}\boldsymbol{G}\boldsymbol{V}_n, \quad \boldsymbol{C}_n := \boldsymbol{V}_n^{\mathrm{H}}\boldsymbol{C}\boldsymbol{V}_n, \quad \boldsymbol{B}_n := \boldsymbol{V}_n^{\mathrm{H}}\boldsymbol{B}, \quad \boldsymbol{L}_n := \boldsymbol{V}_n^{\mathrm{H}}\boldsymbol{L}. \tag{52}$$

In particular, we now assume that $\boldsymbol{V}_n$ is a basis matrix (35) for $\mathscr{K}_n(\boldsymbol{A},\boldsymbol{R})$. In this case, the reduced-order model defined by (9) and (35) represents a (one-sided) projection of the original system (8) onto the nth block-Krylov subspace $\mathscr{K}_n(\boldsymbol{A},\boldsymbol{R})$. In the sequel, we denote the associated reduced-order transfer function by

$$\boldsymbol{H}_n^{(1)}(s) := \boldsymbol{L}_n^{\mathrm{H}}(\boldsymbol{G}_n + s\boldsymbol{C}_n)^{-1}\boldsymbol{B}_n, \tag{53}$$

where the superscript $^{(1)}$ indicates the one-sided projection. In Section 5.1 below, we show that $\boldsymbol{H}_n^{(1)}$ is a certain Padé-type approximant of the original transfer function $\boldsymbol{H}$.

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

Proposition 4. *The reduced-order transfer function $\boldsymbol{H}_n^{(1)}$ given by* (52) *and* (53) *does not depend on the particular choice of the basis matrix* (35), *$\boldsymbol{V}_n$, for the nth block Krylov subspace $\mathscr{K}_n(\boldsymbol{A},\boldsymbol{R})$.*

Proof. Let $\boldsymbol{V}_n$ be the basis matrix for $\mathscr{K}_n(\boldsymbol{A},\boldsymbol{R})$ that is used in (52). Let $\tilde{\boldsymbol{V}}_n$ be any other basis matrix for $\mathscr{K}_n(\boldsymbol{A},\boldsymbol{R})$. In analogy to (52) and (53), $\tilde{\boldsymbol{V}}_n$ induces the reduced-order transfer function

$$\tilde{\boldsymbol{H}}_n(s) = \tilde{\boldsymbol{L}}_n^{\mathrm{H}}(\tilde{\boldsymbol{G}}_n + s\tilde{\boldsymbol{C}}_n)^{-1}\tilde{\boldsymbol{B}}_n, \tag{54}$$

where

$$\tilde{\boldsymbol{G}}_n = \tilde{\boldsymbol{V}}_n^{\mathrm{H}}\boldsymbol{G}\tilde{\boldsymbol{V}}_n, \quad \tilde{\boldsymbol{C}}_n = \tilde{\boldsymbol{V}}_n^{\mathrm{H}}\boldsymbol{C}\tilde{\boldsymbol{V}}_n, \quad \tilde{\boldsymbol{B}}_n = \tilde{\boldsymbol{V}}_n^{\mathrm{H}}\boldsymbol{B}, \quad \tilde{\boldsymbol{L}}_n = \tilde{\boldsymbol{V}}_n^{\mathrm{H}}\boldsymbol{L}.$$

Since $\boldsymbol{V}_n$ and $\tilde{\boldsymbol{V}}_n$ are basis matrices for the same subspace, there exists a nonsingular $n\times n$ matrix $\boldsymbol{M}$ such that $\tilde{\boldsymbol{V}}_n = \boldsymbol{V}_n\boldsymbol{M}$. Using this relation, we obtain from (54) and (52) that

$$\tilde{\boldsymbol{G}}_n = \boldsymbol{M}^{\mathrm{H}}\boldsymbol{G}_n\boldsymbol{M}, \quad \tilde{\boldsymbol{C}}_n = \boldsymbol{M}^{\mathrm{H}}\boldsymbol{C}_n\boldsymbol{M}, \quad \tilde{\boldsymbol{B}}_n = \boldsymbol{M}^{\mathrm{H}}\boldsymbol{B}_n, \quad \tilde{\boldsymbol{L}}_n = \boldsymbol{M}^{\mathrm{H}}\boldsymbol{L}_n. \tag{55}$$

Inserting (55) into (54), we get

$$\begin{aligned}\tilde{\boldsymbol{H}}_n(s) &= \boldsymbol{L}_n^{\mathrm{H}}\boldsymbol{M}(\boldsymbol{M}^{\mathrm{H}}(\boldsymbol{G}_n + s\boldsymbol{C}_n)\boldsymbol{M})^{-1}\boldsymbol{M}^{\mathrm{H}}\boldsymbol{B}_n\\ &= \boldsymbol{L}_n^{\mathrm{H}}(\boldsymbol{G}_n + s\boldsymbol{C}_n)^{-1}\boldsymbol{B}_n = \boldsymbol{H}_n^{(1)}(s).\end{aligned}$$

Thus, the reduced-order transfer functions $\boldsymbol{H}_n^{(1)}$ and $\tilde{\boldsymbol{H}}_n$ are identical. □

4.2. Two-sided projection onto $\mathscr{K}_n(\boldsymbol{A},\boldsymbol{R})$ and $\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}},\boldsymbol{L})$

Let $\boldsymbol{V}_n$ and $\boldsymbol{W}_n$ be any two basis matrices of $\mathscr{K}_n(\boldsymbol{A},\boldsymbol{R})$ and $\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}},\boldsymbol{L})$, and consider the representation (23) of the transfer function $\boldsymbol{H}$ of (8). By projecting the matrices in (23) from the right and left onto the columns of $\boldsymbol{V}_n$ and $\boldsymbol{W}_n$, respectively, we obtain the reduced-order transfer function

$$\boldsymbol{H}_n^{(2)}(s) := (\boldsymbol{V}_n^{\mathrm{H}}\boldsymbol{L})^{\mathrm{H}}(\boldsymbol{W}_n^{\mathrm{H}}\boldsymbol{V}_n + (s-s_0)\boldsymbol{W}_n^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_n)^{-1}(\boldsymbol{W}_n^{\mathrm{H}}\boldsymbol{R}). \tag{56}$$

In analogy to Proposition 4, we have the following result.

Proposition 5. *The reduced-order transfer function* $\boldsymbol{H}_n^{(2)}$ *given by* (56) *does not depend on the particular choice of the basis matrices* $\boldsymbol{V}_n$ *and* $\boldsymbol{W}_n$ *for the nth block Krylov subspaces* $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$ *and* $\mathscr{K}_n(\boldsymbol{A}^{\mathrm{H}}, \boldsymbol{L})$.

Proof. Analogous to the proof of Proposition 4. □

4.3. Computation via Krylov-subspace algorithms

In view of Proposition 5, the reduced-order transfer function (56), $\boldsymbol{H}_n^{(2)}$, can be generated from any two basis matrices $\boldsymbol{V}_n$ and $\boldsymbol{W}_n$. However, there is one distinguished choice of $\boldsymbol{V}_n$ and $\boldsymbol{W}_n$ that eliminates the need to explicitly compute the projections in (56). This choice is the Lanczos basis described in Section 3.5.

Indeed, let $n \geqslant \max\{m_1, p_1\}$, and assume that the Lanczos-type algorithm is run for n steps. Let $\boldsymbol{\Delta}_n$ be the diagonal matrix defined in (43), and let $\boldsymbol{T}_n$, $\boldsymbol{\rho}$, and $\boldsymbol{\eta}$ be the matrices of recurrence coefficients given by (47) and (48). Then, from (43) and (48), it readily follows that

$$\boldsymbol{W}_n^{\mathrm{H}} \boldsymbol{R} = \boldsymbol{\Delta}_n \boldsymbol{\rho}_n \quad \text{and} \quad \boldsymbol{V}_n^{\mathrm{H}} \boldsymbol{L} = \boldsymbol{\Delta}_n^{\mathrm{H}} \boldsymbol{\eta}_n, \tag{57}$$

where

$$\boldsymbol{\rho}_n := \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{0}_{n-m_1 \times m} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\eta}_n := \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{0}_{n-p_1 \times p} \end{bmatrix}. \tag{58}$$

Furthermore, by multiplying the first relation in (47) from the left by $\boldsymbol{W}_n^{\mathrm{H}}$ and using the first relation in (46), as well as (43), we get

$$\boldsymbol{W}_n^{\mathrm{H}} \boldsymbol{A} \boldsymbol{V}_n = \boldsymbol{\Delta}_n \boldsymbol{T}_n \quad \text{and} \quad \boldsymbol{W}_n^{\mathrm{H}} \boldsymbol{V}_n = \boldsymbol{\Delta}_n. \tag{59}$$

Inserting (57) and (59) into (56), it readily follows that

$$\boldsymbol{H}_n^{(2)}(s) = \boldsymbol{\eta}_n^{\mathrm{H}} (\boldsymbol{\Delta}_n^{-1} + (s - s_0) \boldsymbol{T}_n \boldsymbol{\Delta}_n^{-1})^{-1} \boldsymbol{\rho}_n. \tag{60}$$

The MPVL (matrix-Padé via Lanczos) algorithm, which was first proposed in [9], is a numerical procedure for computing $\boldsymbol{H}_n^{(2)}$ via the formula (60).

For symmetric transfer functions (18), by (51) and (58), the reduced-order transfer function (60) is also symmetric:

$$\boldsymbol{H}_n^{(2)}(s) = \boldsymbol{\rho}_n^{\mathrm{T}} (\boldsymbol{\Delta}_n^{-1} + (s - s_0) \boldsymbol{T}_n \boldsymbol{\Delta}_n^{-1})^{-1} \boldsymbol{\rho}_n, \tag{61}$$

where $\boldsymbol{\Delta}_n^{-1}$ and $\boldsymbol{T}_n \boldsymbol{\Delta}_n^{-1}$ are real symmetric matrices. The SyMPVL algorithm [16,17] is a special symmetric variant of the general MPVL algorithm that computes symmetric reduced-order transfer functions (61).

Furthermore, recall from Section 2.4 that RLC subcircuits are described by special symmetric transfer functions (18) with matrices $\boldsymbol{G}$, $\boldsymbol{C}$, and $\boldsymbol{B}$ of the form (16) and (17). In this case, as we will discuss in Section 6, the reduced-order transfer function (60) 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 14 below, is always passive. To this end, let $\boldsymbol{J}$ be the matrix defined in (20), and consider the nonsymmetric formulation (21) of the symmetric transfer function (18). Note that by Lemma 1, both formulations (18) and (21) result in the same nth block Krylov subspace $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$. In particular, the Lanczos basis matrix $\boldsymbol{V}_n$

generated by SyMPVL is also a basis matrix for the nth block Krylov subspace associated with the nonsymmetric formulation (21). Hence we can also use $\boldsymbol{V}_n$ to apply the one-sided projection of Section 4.1 to (21). By (21), (52), and (53), the resulting projected reduced-order transfer function is given by

$$\boldsymbol{H}_n^{(1)}(s) := (\boldsymbol{V}_n^{\mathrm{T}}\boldsymbol{B})^{\mathrm{T}}(\boldsymbol{V}_n^{\mathrm{T}}(\boldsymbol{JG})\boldsymbol{V}_n + s\boldsymbol{V}_n^{\mathrm{T}}(\boldsymbol{JC})\boldsymbol{V}_n)^{-1}(\boldsymbol{V}_n^{\mathrm{T}}\boldsymbol{B}). \tag{62}$$

5. Connections with Padé-type and Padé approximants

In this section, we show that the one-sided projection $\boldsymbol{H}_n^{(1)}$ is actually a matrix-Padé-type approximant of $\boldsymbol{H}$, and we review the matrix-Padé property of $\boldsymbol{H}_n^{(2)}$.

5.1. $\boldsymbol{H}_n^{(1)}$ is a matrix-Padé-type approximant

Although the reduced-order transfer function (53), $\boldsymbol{H}_n^{(1)}$, is defined via the simple one-sided projection (52), it satisfies an approximation property of the form (15), where, however, $q(n)$ is not maximal in general. This means that $\boldsymbol{H}_n^{(1)}$ is a matrix-Padé-type approximant of $\boldsymbol{H}$. For the special case of expansion point $s_0 = 0$ and a basis matrix $\boldsymbol{V}_n$ generated by a simple block Arnoldi procedure without deflation, it was first observed in [22,23] that $\boldsymbol{H}_n^{(1)}$ satisfies an approximation property (15). Here, we extend this result to the most general case of arbitrary expansion points s_0 and arbitrary basis matrices $\boldsymbol{V}_n$ for the properly defined block Krylov subspaces $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$ that allow for necessary deflation of linearly dependent vectors. The only further assumption we need is that the matrix

$$\boldsymbol{G}_n + s_0\boldsymbol{C}_n \quad \text{is nonsingular.} \tag{63}$$

This guarantees that s_0 is not a pole of $\boldsymbol{H}_n^{(1)}$. Since, by (52),

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

the condition (63) also ensures that the matrix $\boldsymbol{G} + s_0\boldsymbol{C}$ is nonsingular.

Expanding the transfer function $\boldsymbol{H}$ in (23) about s_0, we get

$$\boldsymbol{H}(s) = \sum_{i=0}^{\infty} (-1)^i \boldsymbol{M}_i (s - s_0)^i, \quad \text{where } \boldsymbol{M}_i := \boldsymbol{L}^{\mathrm{H}}\boldsymbol{A}^i\boldsymbol{R}. \tag{64}$$

On the other hand, expanding the reduced-order transfer function $\boldsymbol{H}_n^{(1)}$ in (52) about s_0 gives

$$\boldsymbol{H}_n^{(1)}(s) = \sum_{i=0}^{\infty} (-1)^i \boldsymbol{M}_i^{(1)} (s - s_0)^i, \tag{65}$$

where

$$\boldsymbol{M}_i^{(1)} := \boldsymbol{L}_n^{\mathrm{H}}((\boldsymbol{G}_n + s_0\boldsymbol{C}_n)^{-1}\boldsymbol{C}_n)^i(\boldsymbol{G}_n + s_0\boldsymbol{C}_n)^{-1}\boldsymbol{B}_n.$$

We now show that for any n of the form (29), the first j terms in the expansions (64) and (65) are identical. To this end, we first establish the following proposition.

Proposition 6. *Let* $n = m_1 + m_2 + \cdots + m_j$, *where* $1 \leqslant j \leqslant j_{\max}$. *Then, the matrix*

$$\boldsymbol{F}_n := \boldsymbol{V}_n(\boldsymbol{G}_n + s_0\boldsymbol{C}_n)^{-1}\boldsymbol{V}_n^{\mathrm{H}}\boldsymbol{C} \tag{66}$$

satisfies the following relations:

$$\boldsymbol{M}_i^{(1)} = \boldsymbol{L}^{\mathrm{H}}\boldsymbol{F}_n^i\boldsymbol{R} \quad \textit{for all } i = 0, 1, \ldots, \tag{67}$$

$$\boldsymbol{F}_n^i\boldsymbol{R} = \boldsymbol{A}^i\boldsymbol{R} \quad \textit{for all } i = 0, 1, \ldots, j-1. \tag{68}$$

Proof. By (29) and (30), for each $i = 1, 2, \ldots, j$, the columns of the matrix $\boldsymbol{A}^i\boldsymbol{R}$ are all contained in $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$. Since $\boldsymbol{V}_n$ is a basis matrix for $\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{R})$, for each $i = 1, 2, \ldots, j$, there exists an $n \times m$ matrix $\boldsymbol{E}_i$ such that

$$\boldsymbol{A}^{i-1}\boldsymbol{R} = \boldsymbol{V}_n\boldsymbol{E}_i. \tag{69}$$

We now prove (67). From (23) and (69) (for $i = 1$), we get

$$\boldsymbol{B} = (\boldsymbol{G} + s_0\boldsymbol{C})\boldsymbol{R} = (\boldsymbol{G}\boldsymbol{V}_n + s_0\boldsymbol{C}\boldsymbol{V}_n)\boldsymbol{E}_1. \tag{70}$$

Multiplying (70) from the left by $\boldsymbol{V}_n^{\mathrm{H}}$ and using (52), it follows that

$$(\boldsymbol{G}_n + s_0\boldsymbol{C}_n)^{-1}\boldsymbol{B}_n = \boldsymbol{E}_1. \tag{71}$$

Moreover, from (66), we obtain the relation $\boldsymbol{F}_n\boldsymbol{V}_n = \boldsymbol{V}_n(\boldsymbol{G}_n + s_0\boldsymbol{C}_n)^{-1}\boldsymbol{C}_n$, which, by induction on i, implies that

$$\boldsymbol{F}_n^i\boldsymbol{V}_n = \boldsymbol{V}_n((\boldsymbol{G}_n + s_0\boldsymbol{C}_n)^{-1}\boldsymbol{C}_n)^i \quad \text{for all } i = 0, 1, \ldots . \tag{72}$$

Note that, by (52), $\boldsymbol{L}_n^{\mathrm{H}} = \boldsymbol{L}^{\mathrm{H}}\boldsymbol{V}_n$. Using this relation, as well as (65), (71), (72), and (69) (for $i = 1$), it follows that, for all $i = 0, 1, \ldots$,

$$\begin{aligned}\boldsymbol{M}_i^{(1)} &= \boldsymbol{L}^{\mathrm{H}}(\boldsymbol{V}_n((\boldsymbol{G}_n + s_0\boldsymbol{C}_n)^{-1}\boldsymbol{C}_n)^i)((\boldsymbol{G}_n + s_0\boldsymbol{C}_n)^{-1}\boldsymbol{B}_n)\\ &= \boldsymbol{L}^{\mathrm{H}}(\boldsymbol{F}_n^i\boldsymbol{V}_n)\boldsymbol{E}_1 = \boldsymbol{L}^{\mathrm{H}}\boldsymbol{F}_n^i\boldsymbol{R}.\end{aligned}$$

This is just the desired relation (67).

Next, we prove (68) using induction on i. For $i = 0$, (68) is trivially satisfied. Now assume that (68) is true for some $0 \leqslant i < j - 1$. We show that (68) then also holds true for $i + 1$, i.e.,

$$\boldsymbol{F}_n^{i+1}\boldsymbol{R} = \boldsymbol{A}^{i+1}\boldsymbol{R}. \tag{73}$$

Using (23), (68), and (69) (with i replaced by $i + 1$), we get

$$((\boldsymbol{G} + s_0\boldsymbol{C})^{-1}\boldsymbol{C})(\boldsymbol{F}_n^i\boldsymbol{R}) = \boldsymbol{A}(\boldsymbol{A}^i\boldsymbol{R}) = \boldsymbol{A}^{i+1}\boldsymbol{R} = \boldsymbol{V}_n\boldsymbol{E}_{i+1}. \tag{74}$$

Multiplying (74) from the left by $\boldsymbol{V}_n^{\mathrm{H}}(\boldsymbol{G} + s_0\boldsymbol{C})$, it follows that

$$(\boldsymbol{V}_n^{\mathrm{H}}\boldsymbol{C})(\boldsymbol{F}_n^i\boldsymbol{R}) = (\boldsymbol{V}_n^{\mathrm{H}}(\boldsymbol{G} + s_0\boldsymbol{C})\boldsymbol{V}_n)\boldsymbol{E}_{i+1} = (\boldsymbol{G}_n + s_0\boldsymbol{C}_n)\boldsymbol{E}_{i+1}. \tag{75}$$

Using (66) and (69) (with i replaced by $i + 1$), we obtain from (75) that

$$\begin{aligned}\boldsymbol{F}_n^{i+1}\boldsymbol{R} &= \boldsymbol{F}_n(\boldsymbol{F}_n^i\boldsymbol{R}) = \boldsymbol{V}_n((\boldsymbol{G}_n + s_0\boldsymbol{C}_n)^{-1}\boldsymbol{V}_n^{\mathrm{H}}\boldsymbol{C})(\boldsymbol{F}_n^i\boldsymbol{R})\\ &= \boldsymbol{V}_n\boldsymbol{E}_{i+1} = \boldsymbol{A}^{i+1}\boldsymbol{R},\end{aligned}$$

which is just the desired relation (73). □

Theorem 7. *Let $n=m_1+m_2+\cdots+m_j$, where $1\leqslant j\leqslant j_{\max}$, and let $\boldsymbol{H}_n^{(1)}$ be the reduced-order transfer function given by* (52) *and* (53). *Let $s_0\in\mathbb{C}$ be an expansion point such that* (63) *is satisfied. Then, $\boldsymbol{H}_n^{(1)}$ satisfies*

$$\boldsymbol{H}_n^{(1)}(s)=\boldsymbol{H}(s)+\mathcal{O}(s-s_0)^j, \tag{76}$$

i.e., $\boldsymbol{H}_n^{(1)}$ is a matrix-Padé-type approximant of the transfer function (12), $\boldsymbol{H}$.

Proof. By (64) and (65), the assertion (76) is equivalent to

$$\boldsymbol{M}_i^{(1)}=\boldsymbol{M}_i \quad \text{for all } i=0,1,\ldots,j-1. \tag{77}$$

By Proposition 6, the matrix $\boldsymbol{F}_n$ defined in (66) satisfies relations (67) and (68). Inserting (68) into (67) gives

$$\boldsymbol{M}_i^{(1)}=\boldsymbol{L}^{\mathrm{H}}\boldsymbol{F}_n^i\boldsymbol{R}=\boldsymbol{L}^{\mathrm{H}}\boldsymbol{A}^i\boldsymbol{R}=\boldsymbol{M}_i \quad \text{for all } i=0,1,\ldots,j-1,$$

which is just the desired property (77). □

Remark 8. By (28) and (29), we have $j\geqslant\lfloor n/m\rfloor$. Therefore, by Theorem 7, the Taylor expansions of $\boldsymbol{H}_n^{(1)}$ and $\boldsymbol{H}$ about s_0 match in at least the first $\lfloor n/m\rfloor$ coefficient matrices.

Remark 9. If $p<m$, then a matrix-Padé-type approximant that matches at least the first $\lfloor n/p\rfloor$ Taylor coefficient matrices of $\boldsymbol{H}$ about s_0 can be obtained by performing the one-sided projection described in Section 4.1 onto $\mathcal{K}_n(\boldsymbol{A}^{\mathrm{H}},\boldsymbol{L})$, instead of $\mathcal{K}_n(\boldsymbol{A},\boldsymbol{R})$.

5.2. $\boldsymbol{H}_n^{(2)}$ is a matrix-Padé approximant

It turns out that, in general, the reduced-order transfer function $\boldsymbol{H}_n^{(2)}$ defined in (56) is even a better approximation to $\boldsymbol{H}$ than $\boldsymbol{H}_n^{(1)}$. To properly state this result, we first define the integers

$$n_{\min}:=\max\{m_1,p_1\} \quad \text{and} \quad n_{\max}:=\min\left\{\sum_{j=1}^{j_{\max}}m_j,\sum_{k=1}^{k_{\max}}p_k\right\},$$

where the m_j's and p_k's are the integers given by (27), (28) and (32), (33), respectively. The main result of this section is then as follows.

Theorem 10. *Let $n_{\min}\leqslant n\leqslant n_{\max}$, and let $j=j(n)$ and $k=k(n)$ be the maximal integers such that*

$$m_1+m_2+\cdots+m_j\leqslant n \quad \text{and} \quad p_1+p_2+\cdots+p_k\leqslant n, \tag{78}$$

respectively. Let $s_0\in\mathbb{C}$ be an expansion point such that (63) *is satisfied, and let $\boldsymbol{H}_n^{(2)}$ be the reduced-order transfer function given by the two-sided projection* (56). *Then, $\boldsymbol{H}_n^{(2)}$ satisfies*

$$\boldsymbol{H}_n^{(2)}(s)=\boldsymbol{H}(s)+\mathcal{O}(s-s_0)^{q(n)}, \quad \text{where } q(n)=j(n)+k(n). \tag{79}$$

Moreover, in (79), *the exponent $q(n)$ is as large as possible, and hence $\boldsymbol{H}_n^{(2)}$ is a matrix-Padé approximant of the transfer function* (12), $\boldsymbol{H}$.

Proof. In [12], we studied the reduced-order transfer function $\boldsymbol{H}_n^{(2)}$ given by (60), where $\boldsymbol{\rho}_n$, $\boldsymbol{\eta}_n$, $\boldsymbol{\Delta}_n$, and $\boldsymbol{T}_n$ are the matrices generated by n steps of the Lanczos-type method sketched in Section 3.5. In particular, in [12, Theorem 1], we showed that $\boldsymbol{H}_n^{(2)}$ satisfies the properties listed in Theorem 10 above. Recall from Section 4.2 that the reduced-order transfer functions defined in (56) via a two-sided projection and the one given by (60) in terms of the Lanczos-type method are identical. Therefore, the assertions in Theorem 10 follow from [12, Theorem 1]. □

Remark 11. In view of (28), (33), and (78), we have $j(n) \geqslant \lfloor n/m \rfloor$ and $k(n) \geqslant \lfloor n/p \rfloor$. Therefore, by Theorem 10, the Taylor expansions of $\boldsymbol{H}_n^{(2)}$ and $\boldsymbol{H}$ about s_0 match in at least the first $\lfloor n/m \rfloor + \lfloor n/p \rfloor$ coefficient matrices.

6. Passivity and stability

As we discussed in Section 2, in circuit simulation, reduced-order modeling is mostly applied to large passive linear subcircuits, such as RLC networks consisting of only resistors, inductors, and capacitors. 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. [6,27]. Unfortunately, except for special cases such as RC subcircuits consisting of only resistors and capacitors, the Padé model given by $\boldsymbol{H}_n^{(2)}$ is not passive in general; see, e.g. [4,7,14,15,19]. In this section, we derive a sufficient criterion for passivity of general transfer functions, and then apply the criterion to establish passivity of the particular projected model given by (62).

Roughly speaking, a (linear or nonlinear) dynamical system is *passive* if it does not generate energy. The concept was first used in circuit theory; see, e.g. [2,18]. For example, networks consisting of only resistors, inductors, and capacitors are passive.

The following well-known theorem (see, e.g. [2,31]) relates the passivity of the linear dynamical system (8) to the positive realness of its transfer function. Here and in the sequel, we assume that $m = p$ in (8).

Theorem A. *The linear dynamical system* (8) *is passive if, and only if, the associated transfer function* (12), $\boldsymbol{H}$, *is positive real.*

The definition of a positive real matrix-valued function is as follows; see, e.g. [2].

Definition 12. A function $\boldsymbol{H} : \mathbb{C} \mapsto (\mathbb{C} \cup \{\infty\})^{m \times m}$ is called *positive real* if

(i) $\boldsymbol{H}$ has no poles in $\mathbb{C}_+$;
(ii) $\boldsymbol{H}(\bar{s}) = \overline{\boldsymbol{H}(s)}$ for all $s \in \mathbb{C}$;
(iii) $\mathrm{Re}(\boldsymbol{x}^{\mathrm{H}} \boldsymbol{H}(s)\boldsymbol{x}) \geqslant 0$ for all $s \in \mathbb{C}_+$ and $\boldsymbol{x} \in \mathbb{C}^m$.

Recall that a function $\boldsymbol{H} : \mathbb{C} \mapsto (\mathbb{C} \cup \{\infty\})^{m \times m}$ is *stable* if $\boldsymbol{H}$ has no poles in $\mathbb{C}_+$ and if all possible purely imaginary poles of $\boldsymbol{H}$ are simple. It is well known that any positive real function is necessary stable.

Next, we prove the following sufficient condition for positive realness.

Theorem 13. *Let* $\boldsymbol{G}, \boldsymbol{C} \in \mathbb{R}^{N\times N}$, *and* $\boldsymbol{B} \in \mathbb{R}^{N\times m}$. *Assume that* $\boldsymbol{G} + \boldsymbol{G}^{\mathrm{T}} \geqslant 0$, $\boldsymbol{C} = \boldsymbol{C}^{\mathrm{T}} \geqslant 0$, *and that* $\boldsymbol{G} + s\boldsymbol{C}$ *is a regular matrix pencil. Then, the transfer function*

$$\boldsymbol{H}(s) := \boldsymbol{B}^{\mathrm{T}}(\boldsymbol{G} + s\boldsymbol{C})^{-1}\boldsymbol{B}, \quad s \in \mathbb{C}, \tag{80}$$

is positive real.

Proof. We need to show that $\boldsymbol{H}$ satisfies the conditions (i)–(iii) given in Definition 12.

Condition (ii) follows directly from the fact that the matrices $\boldsymbol{G}$, $\boldsymbol{C}$, and $\boldsymbol{B}$ in (80) are real.

Next, we verify condition (iii). Let $s = s_1 + is_2 \in \mathbb{C}_+$ be arbitrary, but fixed. Here $i := \sqrt{-1}$ denotes the purely imaginary unit. Note that

$$(\boldsymbol{G} + s\boldsymbol{C})^{\mathrm{H}} = \boldsymbol{S} - \boldsymbol{K}, \tag{81}$$

where

$$\boldsymbol{S} := \tfrac{1}{2}(\boldsymbol{G} + \boldsymbol{G}^{\mathrm{T}}) + s_1\boldsymbol{C}, \quad \boldsymbol{K} := \tfrac{1}{2}(\boldsymbol{G} - \boldsymbol{G}^{\mathrm{T}}) + is_2\boldsymbol{C}.$$

Recall that $\boldsymbol{G} + \boldsymbol{G}^{\mathrm{T}} \geqslant 0$, $\boldsymbol{C} = \boldsymbol{C}^{\mathrm{T}} \geqslant 0$, and $s_1 = \mathrm{Re}\, s > 0$. This guarantees that $\boldsymbol{S} \geqslant 0$ and $\boldsymbol{K} = -\boldsymbol{K}^{\mathrm{H}}$. These properties imply that $\boldsymbol{y}^{\mathrm{H}}\boldsymbol{S}\boldsymbol{y} \geqslant 0$ and $\mathrm{Re}(\boldsymbol{y}^{\mathrm{H}}\boldsymbol{K}\boldsymbol{y}) = 0$ for all $\boldsymbol{y} \in \mathbb{C}^N$. Therefore, by (81), we have

$$\mathrm{Re}(\boldsymbol{y}^{\mathrm{H}}(\boldsymbol{G} + s\boldsymbol{C})^{\mathrm{H}}\boldsymbol{y}) = \boldsymbol{y}^{\mathrm{H}}\boldsymbol{S}\boldsymbol{y} \geqslant 0 \quad \text{for all } \boldsymbol{y} \in \mathbb{C}^N. \tag{82}$$

We now assume that $s \in \mathbb{C}_+$ is such that the matrix $\boldsymbol{G} + s\boldsymbol{C}$ is nonsingular. Furthermore, let $\boldsymbol{x} \in \mathbb{C}^m$ be arbitrary, and set

$$\boldsymbol{y} := (\boldsymbol{G} + s\boldsymbol{C})^{-1}\boldsymbol{B}\boldsymbol{x}. \tag{83}$$

Then, by (80) and (83), we have

$$\begin{aligned}\boldsymbol{x}^{\mathrm{H}}\boldsymbol{H}(s)\boldsymbol{x} &= \boldsymbol{x}^{\mathrm{H}}\boldsymbol{B}^{\mathrm{T}}(\boldsymbol{G} + s\boldsymbol{C})^{-1}\boldsymbol{B}\boldsymbol{x} \\ &= \boldsymbol{x}^{\mathrm{H}}\boldsymbol{B}^{\mathrm{T}}(\boldsymbol{G} + s\boldsymbol{C})^{-\mathrm{H}}(\boldsymbol{G} + s\boldsymbol{C})^{\mathrm{H}}(\boldsymbol{G} + s\boldsymbol{C})^{-1}\boldsymbol{B}\boldsymbol{x} \\ &= \boldsymbol{y}^{\mathrm{H}}(\boldsymbol{G} + s\boldsymbol{C})^{\mathrm{H}}\boldsymbol{y}.\end{aligned} \tag{84}$$

Combining (82) and (84), it follows that

$$\mathrm{Re}(\boldsymbol{x}^{\mathrm{H}}\boldsymbol{H}(s)\boldsymbol{x}) = \mathrm{Re}(\boldsymbol{y}^{\mathrm{H}}(\boldsymbol{G} + s\boldsymbol{C})^{\mathrm{H}}\boldsymbol{y}) \geqslant 0 \quad \text{for all } \boldsymbol{x} \in \mathbb{C}^m \tag{85}$$

and for all $s \in \mathbb{C}_+$ for which $\boldsymbol{G} + s\boldsymbol{C}$ is nonsingular. Now let $\hat{s} \in \mathbb{C}_+$ be such that $\boldsymbol{G} + \hat{s}\boldsymbol{C}$ is singular. Note that there are at most N such "singular" points $\hat{s}$, since $\boldsymbol{G} + s\boldsymbol{C}$ is assumed to be a regular matrix pencil. Therefore, each $\hat{s}$ is an isolated point in $\mathbb{C}_+$, i.e., there exists an $\varepsilon = \varepsilon(\hat{s}) > 0$ and a (punctured) neighborhood

$$D_\varepsilon := \{s \in \mathbb{C} \mid 0 < |s - \hat{s}| \leqslant \varepsilon\}$$

of $\hat{s}$ such that $D_\varepsilon \subset \mathbb{C}_+$ and the matrix $\boldsymbol{G} + s\boldsymbol{C}$ is nonsingular for all $s \in D_\varepsilon$. Thus (85) holds true for all $s \in D_\varepsilon$. If $\hat{s}$ is not a pole of the rational function $\boldsymbol{H}$, then $\boldsymbol{H}(\hat{s}) = \lim_{s\to\hat{s}} \boldsymbol{H}(s)$ is a finite $m \times m$ matrix. In this case, by taking limits $s \to \hat{s}$ in (85), it follows that (85) also holds true for $s = \hat{s}$. Now suppose that $\hat{s}$ is a pole of $\boldsymbol{H}$. Then at least one of the components $h_{jk}(s)$ of $\boldsymbol{H}(s) = [h_{jk}(s)]_{1\leqslant j,k\leqslant m}$ has a pole at $\hat{s}$. Such an $h_{jk}(s)$ maps D_ε onto a suitable neighborhood of ∞ in the complex plane and, in particular, attains large negative numbers in D_ε. By selecting a suitable component h_{jk} of $\boldsymbol{H}$ and an associated vector $\boldsymbol{x} \in \mathbb{C}^N$, it is thus possible to find a point $s \in D_\varepsilon$ such

that $\mathrm{Re}(\boldsymbol{x}^{\mathrm{H}}\boldsymbol{H}(s)\boldsymbol{x}) < 0$. However, this is a contradiction to (85), and therefore, $\hat{s}$ cannot be a pole of $\boldsymbol{H}$. This concludes the proof of (iii).

It remains to verify condition (i). By (80), if $\hat{s}$ is a pole of $\boldsymbol{H}$, then the matrix $\boldsymbol{G} + \hat{s}\boldsymbol{C}$ is necessarily singular, i.e., $\hat{s}$ is a singular point. However, we have just shown that there are no such singular points $\hat{s} \in \mathbb{C}_+$. Consequently, $\boldsymbol{H}$ cannot have poles in $\mathbb{C}_+$.

The matrix function $\boldsymbol{H}$ satisfies all three conditions (i)–(iii), and hence $\boldsymbol{H}$ is positive real. □

Finally, we apply Theorem 13 to the reduced-order transfer function (62).

Corollary 14. *Let $\boldsymbol{H}$ be the transfer function given by* (21) *with matrices that satisfy* (22). *Let $\boldsymbol{V}_n \in \mathbb{R}^{N\times n}$ have rank n and assume that the matrix pencil*

$$\boldsymbol{G}_n + s\boldsymbol{C}_n := \boldsymbol{V}_n^{\mathrm{T}}(\boldsymbol{J}\boldsymbol{G})\boldsymbol{V}_n + s\boldsymbol{V}_n^{\mathrm{T}}(\boldsymbol{J}\boldsymbol{C})\boldsymbol{V}_n \tag{86}$$

is regular. Then, the reduced-order transfer function

$$\boldsymbol{H}_n^{(1)}(s) := (\boldsymbol{V}_n^{\mathrm{T}}\boldsymbol{B})^{\mathrm{T}}(\boldsymbol{G}_n + s\boldsymbol{C}_n)^{-1}(\boldsymbol{V}_n^{\mathrm{T}}\boldsymbol{B}) \tag{87}$$

is positive real, and thus the reduced-order model given by (87) *is passive.*

Proof. By (22) and (86), it follows that $\boldsymbol{G}_n + \boldsymbol{G}_n^{\mathrm{T}} \geqslant 0$ and $\boldsymbol{C}_n = \boldsymbol{C}_n^{\mathrm{T}} \geqslant 0$. The transfer function (87), $\boldsymbol{H}_n^{(1)}$, is thus positive real by Theorem 13. □

7. Numerical examples

In this section, we present two circuit examples.

7.1. A package model

The first example 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 eight exterior and eight 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 [16], SyMPVL was used to compute a Padé-based reduced-order model (61) of the package, and it was found that a model $\boldsymbol{H}_n^{(2)}$ of order $n = 80$ is sufficient to match the transfer-function components of interest. However, the model $\boldsymbol{H}_n^{(2)}$ 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 $\boldsymbol{H}_n^{(1)}$ given by (62). The expansion point $s_0 = 5\pi \times 10^9$ was used. Recall that $\boldsymbol{H}_n^{(1)}$ is only a Padé-type approximant and thus less accurate than the Padé approximant $\boldsymbol{H}_n^{(2)}$. Therefore, one now has to go to order $n = 112$ to obtain a projected reduced-order model $\boldsymbol{H}_n^{(1)}$ that matches the transfer-function components of interest. Figs. 1 and 2 show the voltage-to-voltage transfer function between the external terminal of pin no. 1 and the

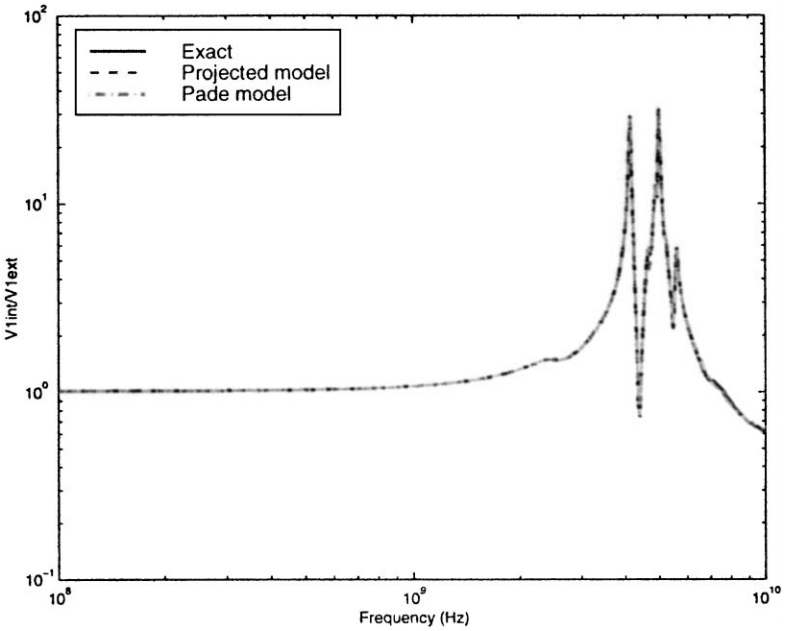

Fig. 1. Package: Pin no. 1 external to Pin no. 1 internal, exact, projected model, and Padé model.

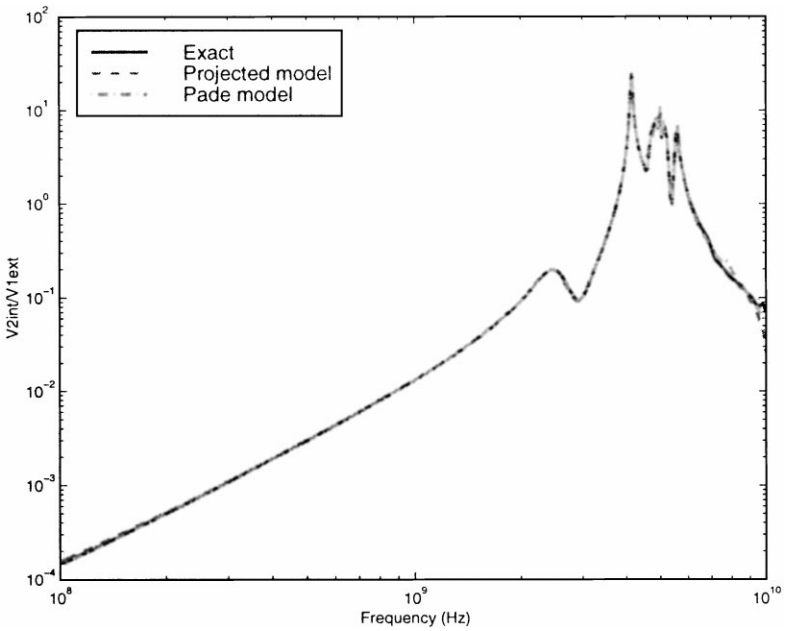

Fig. 2. Package: Pin no. 1 external to Pin no. 2 internal, exact, projected model, and Padé model.

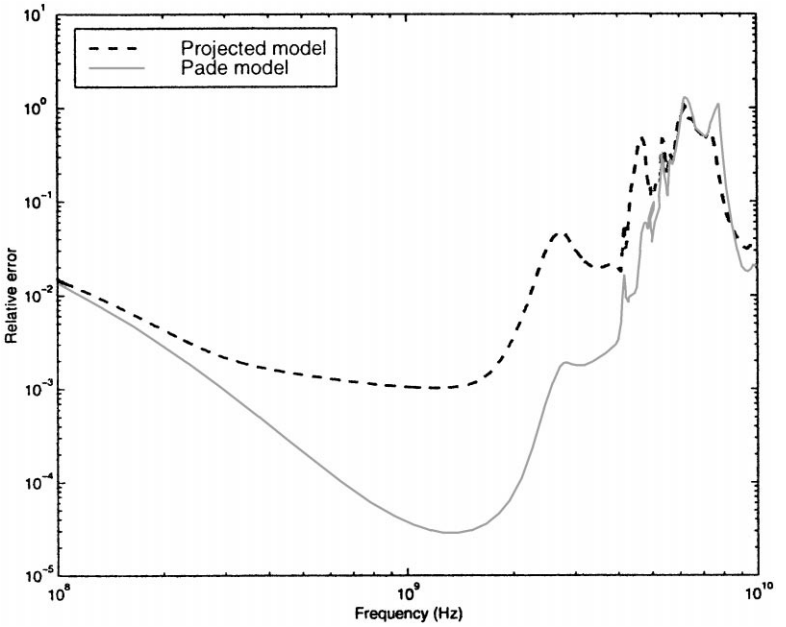

Fig. 3. Relative error of projected model and Padé model.

internal terminals of the same pin and the neighboring pin no. 2, respectively. The plots show results with the projected model $\boldsymbol{H}_n^{(1)}$ and the Padé model $\boldsymbol{H}_n^{(2)}$, both of order $n = 112$, compared with an exact analysis.

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

7.2. An extracted RC circuit

This is an extracted RC circuit with about 4000 elements and $m = 20$ ports. The expansion point $s_0 = 0$ was used. Since the projected model and the Padé model are identical for RC circuits, we only computed the Padé model via SyMPVL.

The point of this example is to illustrate the usefulness of the deflation procedure built into SyMPVL. It turned out that sweeps through the first two Krylov blocks, $\boldsymbol{R}$ and $\boldsymbol{AR}$, of the block Krylov sequence (26) were sufficient to obtain a reduced-order model that matches the transfer function in the frequency range of interest. During the sweep through the second block, 6 almost linearly dependent vectors were discovered and deflated. As a result, the reduced-order model obtained with deflation is only of size $n = 2m - 6 = 34$. When SyMPVL was rerun on this example, with deflation turned off, a reduced-order model of size $n = 40$ was needed to match the transfer function. In Fig. 4, we show the $\boldsymbol{H}_{1,11}$ component of the reduced-order model obtained with deflation and without deflation, compared to the exact transfer function. Clearly, deflation leads to a significantly smaller reduced-order model that is as accurate as the bigger one generated without deflation.

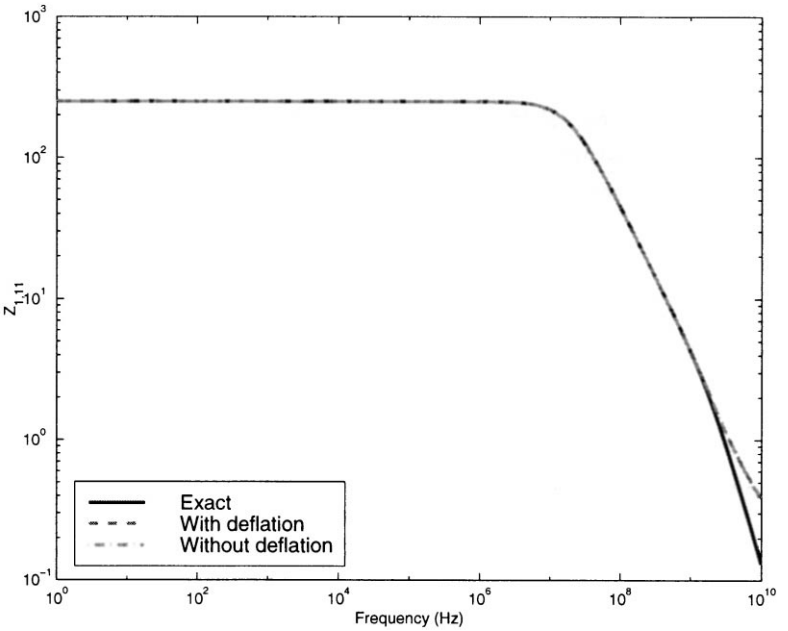

Fig. 4. Impedance $H_{1,11}$.

8. 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.

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.

Finally, today's circuit simulation is based on the paradigm of lumped circuit elements, which leads to systems of DAEs. As circuit feature sizes continue to decrease and circuit speeds continue to increase, feature sizes are becoming comparable in size with signal wavelengths. As a result, at least parts of a circuit must be modeled as distributed elements, such as transmission lines. Including distributed elements in the simulation paradigm requires a fusion of traditional lumped circuit simulation and electromagnetic simulation. Electromagnetic simulation, however, involves systems of partial differential equations (PDEs). Combining lumped circuit simulation with electromagnetic

simulation will thus require efficient techniques for the solution of very large systems of DAEs coupled with PDEs. One of the challenges then is to develop reduced-order modeling techniques that allow to replace parts of such coupled systems with much smaller models. Research into and development of such techniques have hardly begun.

References

[1] J.I. Aliaga, D.L. Boley, R.W. Freund, V. Hernández, A Lanczos-type method for multiple starting vectors, Math. Comp. 69 (2000) 1577–1601; Also available online from http://cm.bell-labs.com/cs/doc/98.
[2] B.D.O. Anderson, S. Vongpanitlerd, Network Analysis and Synthesis, Prentice-Hall, Englewood Cliffs, NJ, 1973.
[3] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9 (1951) 17–29.
[4] Z. Bai, P. Feldmann, R.W. Freund, How to make theoretically passive reduced-order models passive in practice, in: Proceedings of IEEE 1998 Custom Integrated Circuits Conference, IEEE, Piscataway, NJ, 1998, pp. 207–210.
[5] G.A. Baker Jr., P. Graves-Morris, Padé Approximants, 2nd Edition, Cambridge University Press, New York, 1996.
[6] P.M. Chirlian, Integrated and Active Network Analysis and Synthesis, Prentice-Hall, Englewood Cliffs, NJ, 1967.
[7] I.M. Elfadel, D.D. Ling, Zeros and passivity of Arnoldi-reduced-order models for interconnect networks, in: Proceedings of the 34th ACM/IEEE Design Automation Conference, ACM, New York, 1997, pp. 28–33.
[8] 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.
[9] P. Feldmann, R.W. Freund, Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm, in: Proceedings of the 32nd ACM/IEEE Design Automation Conference, ACM, New York, 1995, pp. 474–479.
[10] P. Feldmann, R.W. Freund, Numerical Simulation of Electronic Circuits: State-of-the-Art Techniques and Challenges, Course Notes, 1995. Available online from http://cm.bell-labs.com/who/freund.
[11] P. Feldmann, R.W. Freund, T. Young, Interconnect Extraction and Analysis in High-Frequency, Sub-Micron, Digital VLSI Design, Tutorial at the 35th Design Automation Conference, San Francisco, CA, 1998. Available online from http://cm.bell-labs.com/who/freund.
[12] R.W. Freund, Computation of matrix Padé approximations of transfer functions via a Lanczos-type process, in: C.K. Chui, L.L. Schumaker (Eds.), Approximation Theory VIII, Vol. 1: Approximation and Interpolation, World Scientific, Singapore, 1995, pp. 215–222.
[13] R.W. Freund, Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation, in: B.N. Datta (Ed.), Applied and Computational Control, Signals, and Circuits, Vol. 1, Birkhäuser, Boston, 1999, pp. 435–498.
[14] R.W. Freund, Passive reduced-order models for interconnect simulation and their computation via Krylov-subspace algorithms, in: Proceedings of the 36th ACM/IEEE Design Automation Conference, ACM, New York, 1999, pp. 195–200.
[15] R.W. Freund, P. Feldmann, Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm, in: Tech. Dig. 1996 IEEE/ACM International Conference on Computer-Aided Design, IEEE Computer Society Press, Los Alamitos, CA, 1996, pp. 280–287.
[16] R.W. Freund, P. Feldmann, The SyMPVL algorithm and its applications to interconnect simulation, in: Proceedings of the 1997 International Conference on Simulation of Semiconductor Processes and Devices, IEEE, Piscataway, NJ, 1997, pp. 113–116.
[17] R.W. Freund, P. Feldmann, Reduced-order modeling of large linear passive multi-terminal circuits using matrix-Padé approximation, in: Proc. Design, Automation and Test in Europe Conference 1998, IEEE Computer Society Press, Los Alamitos, CA, 1998, pp. 530–537.
[18] E.A. Guillemin, Synthesis of Passive Networks, Wiley, New York, 1957.
[19] K.J. Kerns, A.T. Yang, Preservation of passivity during RLC network reduction via split congruence transformations, in: Proceedings of the 34th ACM/IEEE Design Automation Conference, ACM, New York, 1997, pp. 34–39.
[20] S.-Y. Kim, N. Gopal, L.T. Pillage, Time-domain macromodels for VLSI interconnect analysis, IEEE Trans. Computer-Aided Design 13 (1994) 1257–1270.

[21] 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.
[22] A. Odabasioglu, Provably passive RLC circuit reduction, M.S. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1996.
[23] A. Odabasioglu, M. Celik, L.T. Pileggi, PRIMA: passive reduced-order interconnect macromodeling algorithm, in: Tech. Dig. 1997 IEEE/ACM International Conference on Computer-Aided Design, IEEE Computer Society Press, Los Alamitos, CA, 1997, pp. 58–65.
[24] L.T. Pileggi, Coping with RC(L) interconnect design headaches, in: Tech. Dig. 1995 IEEE/ACM International Conference on Computer-Aided Design, IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 246–253.
[25] L.T. Pillage, R.A. Rohrer, Asymptotic waveform evaluation for timing analysis, IEEE Trans. Computer-Aided Design 9 (1990) 352–366.
[26] V. Raghavan, J.E. Bracken, R.A. Rohrer, AWESpice: A general tool for the accurate and efficient simulation of interconnect problems, in: Proceedings of the 29th ACM/IEEE Design Automation Conference, ACM, New York, 1992, pp. 87–92.
[27] R.A. Rohrer, H. Nosrati, Passivity considerations in stability studies of numerical integration algorithms, IEEE Trans. Circuits Syst. 28 (1981) 857–866.
[28] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, Boston, 1996.
[29] A.L. Sangiovanni-Vincentelli, Circuit simulation, in: P. Antognetti, D.O. Pederson, D. de Man (Eds.), Computer Design Aids for VLSI Circuits, Sijthoff & Noordhoff, Alphen aan de Rijn, The Netherlands, 1981, pp. 19–112.
[30] J. Vlach, K. Singhal, Computer Methods for Circuit Analysis and Design, 2nd Edition, Van Nostrand Reinhold, New York, 1993.
[31] M.R. Wohlers, Lumped and Distributed Passive Networks, Academic Press, New York, 1969.