Acta Numerica (2003), pp. 267–319
DOI: 10.1017/S0962492902000120

Model reduction methods based on Krylov subspaces

Roland W. Freund
Bell Laboratories, Lucent Technologies,
Room 2C–525,
Murray Hill, NJ 07974–0636, USA
E-mail: freund@research.bell-labs.com

In recent years, reduced-order modelling techniques based on Krylov-subspace iterations, especially the Lanczos algorithm and the Arnoldi process, have become popular tools for tackling the large-scale time-invariant linear dynamical systems that arise in the simulation of electronic circuits. This paper reviews the main ideas of reduced-order modelling techniques based on Krylov subspaces and describes some applications of reduced-order modelling in circuit simulation.

CONTENTS

1. Introduction

Roughly speaking, the problem of model reduction is to replace a given mathematical model of a system or a process by a model that is much 'smaller' than the original model, but still describes, at least 'approximately', certain aspects of the system or process. Clearly, model reduction involves a number of interesting issues. First and foremost is the issue of selecting appropriate approximation schemes that allow the definition of suitable reduced-order models. In addition, it is often important that the reduced-order model preserves certain crucial properties of the original system, such as stability or passivity. Other issues include the characterization of the quality of the models, the extraction of the data from the original model that is needed to actually generate the reduced-order models, and the efficient and numerically stable computation of the models. We refer the reader to Fortuna, Nunnari and Gallo (1992) for a review of general model-reduction techniques, and to the more specialized survey papers by Bultheel and Van Barel (1986), Bultheel and De Moor (2000), Freund (1997, 1999*b*), and Bai (2002), which review methods based on Padé and more general rational approximation, and techniques tailored to applications in VLSI circuit simulation.

In this paper, we discuss Krylov subspace-based reduced-order modelling techniques for large-scale linear dynamical systems, especially those that arise in the simulation of electronic circuits and of microelectromechanical systems.

We begin with a brief description of reduced-order modelling problems in circuit simulation. Electronic circuits are usually modelled as networks whose branches correspond to the circuit elements and whose nodes correspond to the interconnections of the circuit elements. Such networks are characterized by three types of equation. *Kirchhoff's current law* (KCL) states that, for each node of the network, the currents flowing in and out of that node sum up to zero. *Kirchhoff's voltage law* (KVL) states that, for each closed loop of the network, the voltage drops along that loop sum up to zero. *Branch constitutive relations* (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 time-derivatives of the unknowns, and thus they are *ordinary differential equations* (ODEs). On the other hand, the KCLs and KVLs are linear algebraic equations that only depend on the topology of the circuit.

The KCLs, KVLs, and BCRs can be summarized as a system of first-order, in general nonlinear, *differential-algebraic equations* (DAEs) of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\, q(\hat{x},t) + f(\hat{x},t) = 0, \tag{1.1}$$

together with suitable initial conditions. Here, $\hat{x} = \hat{x}(t)$ is the unknown vector of circuit variables at time t, the vector-valued function $f(\hat{x},t)$ represents the contributions of nonreactive elements, such as resistors and sources, and the vector-valued function $\frac{\mathrm{d}}{\mathrm{d}t}\, q(\hat{x},t)$ represents the contributions of reactive elements, such as capacitors and inductors. There are several established

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

methods (Vlach and Singhal 1994), such as sparse tableau, nodal formulation, and modified nodal analysis, for generating a system of equations of the form (1.1) from a so-called *netlist* description of a given circuit. The vector functions $\hat{x}$, f, q, as well as their dimension, 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 of the system (1.1) is of the order of the number of elements in the circuit. Since today's VLSI circuits can have up to hundreds of millions of circuit elements, systems (1.1) describing such circuits can be of extremely large dimension. Reduced-order modelling allows us to first replace large systems of the form (1.1) by systems of smaller dimension and then tackle these smaller systems by suitable DAE solvers. Ideally, we would like to apply nonlinear reduced-order modelling directly to the nonlinear system (1.1). However, since nonlinear reduction techniques are a lot less developed and less well understood than linear ones, linear reduced-order modelling is almost always employed at present. To that end, we either linearize the system (1.1) or decouple (1.1) into nonlinear and linear subsystems; see, *e.g.*, Freund (1999*b*) and the references given there.

For example, the first case arises in *small-signal analysis*; see, *e.g.*, Freund and Feldmann (1996*b*). Given a *DC operating point*, say $\hat{x}_0$, of the circuit described by (1.1), we linearize the system (1.1) around $\hat{x}_0$. The resulting linearized version of (1.1) is of the following form:

$$E\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + Bu(t), \tag{1.2}$$

$$y(t) = C^T x(t). \tag{1.3}$$

Here, $A = -\mathrm{D}_x f$ is the negative of the Jacobian matrix of f at the DC operating point $\hat{x}_0$, and $E = \mathrm{D}_x q$ is the Jacobian matrix of q at $\hat{x}_0$. Furthermore, $x(t) = \hat{x}(t) - \hat{x}_0$ is the distance of the solution $\hat{x}$ of (1.1) to the DC operating point, $u(t)$ is the vector of excitations applied to the sources of the circuit, and $y(t)$ is the vector of circuit variables of interest. Equations (1.2) and (1.3) represent a *time-invariant linear dynamical system*. Its *state-space dimension*, N, is the length of the vector x of circuit variables. For a circuit with many elements, the system (1.2) and (1.3) is thus of very high dimension. The idea of reduced-order modelling is then to replace the system (1.2) and (1.3) by one of the same form,

$$\begin{aligned} E_n\frac{\mathrm{d}z}{\mathrm{d}t} &= A_n z + B_n u(t), \\ y(t) &= C_n^T z(t), \end{aligned}$$

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

but of much smaller state-space dimension $n \ll N$.

Time-invariant linear dynamical systems of the form (1.2) and (1.3) also arise when equations describing linear subcircuits of a given circuit are decoupled from the system (1.1) that characterizes the whole circuit; see, *e.g.*, Freund (1999*b*). For example, as discussed in Cheng, Lillis, Lin and Chang (2000), the interconnect and the pin package of VLSI circuits are often modelled as large linear RCL networks consisting only of resistors, capacitors, and inductors. Such linear subcircuits are described by systems of the form (1.2) and (1.3), where $x(t)$ is the vector of circuit variables associated with the subcircuit, and the vectors $u(t)$ and $y(t)$ contain the variables of the connections of the subcircuit to the generally nonlinear remainder of the whole circuit. By replacing, in the nonlinear system (1.1), the linear subsystem (1.2) and (1.3) by a reduced-order model of much smaller state-space dimension, the dimension of (1.1) can be reduced significantly, before a DAE solver is then applied to such a smaller version of (1.1).

The remainder of this paper is organized as follows. In Section 2, we review some basic facts about time-invariant linear dynamical systems. In Section 3, we introduce reduced-order models that are defined via Padé or Padé-type approximation. In Section 4, we discuss the concepts of stability and passivity of linear dynamical systems. In Section 5, we discuss reduced-order modelling approaches based on Lanczos and Lanczos-type methods. In Section 6, we describe the use of the Arnoldi process for reduced-order modelling. In Section 7, we discuss reduced-order modelling of noise-type transfer functions, which arise in circuit-noise computations. Section 8 is concerned with reduced-order modelling of second-order dynamical systems. Finally, in Section 9, we make some concluding remarks.

2. Time-invariant linear dynamical systems

In this section, we review some basic facts about time-invariant linear dynamical systems.

2.1. State-space description

We consider m-input p-output time-invariant linear dynamical systems given by a *state-space description* of the form

$$E\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + Bu(t), \tag{2.1}$$

$$y(t) = C^T x(t) + Du(t), \tag{2.2}$$

together with suitable initial conditions. Here, A, $E \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times m}$, $C \in \mathbb{R}^{N\times p}$, and $D \in \mathbb{R}^{p\times m}$ are given matrices, $x(t) \in \mathbb{R}^N$ is the vector of state variables, $u(t) \in \mathbb{R}^m$ is the vector of inputs, $y(t) \in \mathbb{R}^p$ the vector

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

of outputs, N is the state-space dimension, and m and p are the number of inputs and outputs, respectively. Note that systems of the form (1.2) and (1.3) are just a special case of (2.1) and (2.2) with $D = 0$.

The linear system (2.1) and (2.2) is called *regular* if the matrix E in (2.1) is nonsingular, and it is called *singular* or a *descriptor system* if E is singular. In the regular case, the linear system (2.1) and (2.2) can always be rewritten as

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t} &= \left(E^{-1}A\right)x + \left(E^{-1}B\right)u(t),\\ y(t) &= C^T x(t) + Du(t),\end{aligned}$$

which is also a system of the form (2.1) and (2.2) with $E = I$. Note that the first equation is just a system of ODEs.

The linear dynamical systems arising in circuit simulation are descriptor systems in general. Therefore, in the following, we allow $E \in \mathbb{R}^{N\times N}$ to be a general, possibly singular, matrix. The only assumption that we make on the matrices A and E in (2.1) is that the matrix pencil $A - sE$ be *regular*, that is, the matrix $A - sE$ is singular only for finitely many values of $s \in \mathbb{C}$.

In the case of singular E, equation (2.1) represents a system of DAEs, rather than ODEs. Solving DAEs is significantly more complex than solving ODEs. Moreover, there are constraints on the possible initial conditions that can be imposed on the solutions of (2.1). For a detailed discussion of DAEs and the structure of their solutions, we refer the reader to Campbell (1980, 1982), Dai (1989), and Verghese, Lévy and Kailath (1981). Here, we only present a brief glimpse of the issues arising in DAEs.

A general descriptor system (2.1) has three different types of *modes*, which are characterized by the eigenstructure of the pencil $A - sE$; see, *e.g.*, Bender and Laub (1987). The finite eigenvalues, $s \in \mathbb{C}$, of the pencil are the *finite dynamic* modes of (2.1). The eigenvectors associated with an infinite eigenvalue $s = \infty$ of the pencil span the space of *nondynamic* solutions of (2.1), and the corresponding eigenvalues $s = \infty$ are the *nondynamic* modes of (2.1). Note that the set of nondynamic solutions of (2.1) is just the null space of E. If $r := \operatorname{rank} E$ is bigger than the degree ρ of the characteristic polynomial $\det(A - sE)$, then the pencil also has $r - \rho$ *impulsive* modes. The impulsive modes correspond to generalized eigenvectors of eigenvalues $s = \infty$ with Jordan blocks of size bigger than 1. A descriptor system is called *impulsive-free* if it has no impulsive modes.

To explain the different types of modes further, we bring the matrices A and E in (2.1) to an appropriate normal form. For any regular pencil $A - sE$, there exist nonsingular matrices P and Q such that

$$P\,(A - sE)\,Q = \begin{bmatrix} A^{(1)} - sI & 0 \\ 0 & I - sJ \end{bmatrix}, \tag{2.3}$$

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

where the submatrix J is nilpotent. The matrix pencil on the right-hand side of (2.3) is called the *Weierstrass form* of $A - sE$. Assuming that the matrices A and E in (2.1) are already in Weierstrass form, the system (2.1) can be decoupled as follows:

$$\frac{\mathrm{d}x^{(1)}}{\mathrm{d}t} = A^{(1)}x^{(1)} + B^{(1)}u(t), \tag{2.4}$$

$$J\frac{\mathrm{d}x^{(2)}}{\mathrm{d}t} = x^{(2)} + B^{(2)}u(t). \tag{2.5}$$

The first subsystem, (2.4), is just a system of linear ODEs. Thus, for any given initial condition $x^{(1)}(0) = \hat{x}^{(1)}$, there exists a unique solution of (2.4). Moreover, the so-called *free-response* of (2.4), that is, the solutions $x(t)$ for $t \geq 0$ when $u \equiv 0$, consists of combinations of exponential modes at the eigenvalues of the matrix $A^{(1)}$. Note that, in view of (2.3), the eigenvalues of $A^{(1)}$ are just the finite eigenvalues of the pencil $A - sE$ and thus they are the finite dynamic modes. The solutions of the second subsystem, (2.5), however, are of a quite different nature. In particular, the free-response of (2.5) consists of $k_i - 1$ independent impulsive motions for each $k_i \times k_i$ Jordan block of the matrix J; see, *e.g.*, Verghese *et al.* (1981).

For example, consider the case when the nilpotent matrix J in (2.5) is a single $k \times k$ Jordan block, that is,

$$J = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{k \times k}.$$

The k components of the free-response $x^{(2)}(t)$ of (2.5) are then given by

$$\begin{aligned} x_1^{(2)}(t) &= -x_2^{(2)}(0-)\delta(t) - x_3^{(2)}(0-)\delta^{(1)}(t) - \cdots - x_k^{(2)}(0-)\delta^{(k-2)}(t), \\ x_2^{(2)}(t) &= -x_3^{(2)}(0-)\delta(t) - x_4^{(2)}(0-)\delta^{(1)}(t) - \cdots - x_k^{(2)}(0-)\delta^{(k-3)}(t), \\ \vdots &= \vdots \\ x_{k-1}^{(2)}(t) &= -x_k^{(2)}(0-)\delta(t), \\ x_k^{(2)}(t) &= 0. \end{aligned}$$

Here, $\delta(t)$ is the delta function and $\delta^{(i)}(t)$ is its ith derivative. Moreover, $x_i^{(2)}(0-)$, $i = 2, 3, \ldots, k$, are the components of the initial conditions that can be imposed on (2.5). Note that there are only $k - 1$ degrees of freedom for the initial condition and that it is not possible to prescribe $x_1^{(2)}(0-)$.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

In particular, the free-response of (2.5) corresponding to a 1×1 Jordan block of J is just the zero solution, and there is no degree of freedom for the selection of an initial value corresponding to that block.

Finally, we remark that, in view of (2.3), the eigenvalues of the matrix pencil $A - sE$ corresponding to the subsystem (2.5) are just the infinite eigenvalues of $A - sE$ and thus the nondynamic modes.

2.2. Reduced-order models and transfer functions

The basic idea of reduced-order modelling is to replace a given system by a system of the same type, but with smaller state-space dimension. Thus, a *reduced-order model* of state-space dimension n of a given time-invariant linear dynamical system (2.1) and (2.2) of dimension N is a system of the form

$$E_n \frac{\mathrm{d}z}{\mathrm{d}t} = A_n z + B_n u(t), \tag{2.6}$$

$$y(t) = C_n^T z(t) + D_n u(t), \tag{2.7}$$

where A_n, $E_n \in \mathbb{R}^{n \times n}$, $B_n \in \mathbb{R}^{n \times m}$, $C_n \in \mathbb{R}^{n \times p}$, $D_n \in \mathbb{R}^{p \times m}$, and $n < N$.

The challenge then is to choose the matrices A_n, E_n, B_n, C_n, and D_n in (2.6) and (2.7) such that the reduced-order model in some sense approximates the original system. One possible measure of the approximation quality of reduced-order models is based on the concept of transfer functions.

If we assume zero initial conditions, then, by applying the Laplace transform

$$X(s) = \int_0^\infty x(t)\, \mathrm{e}^{-st}\, \mathrm{d}t$$

to the original system (2.1) and (2.2), we obtain the following algebraic equations:

$$\begin{aligned} sEX(s) &= AX(s) + BU(s), \\ Y(s) &= C^T X(s) + DU(s). \end{aligned}$$

Here, the frequency-domain variables $X(s)$, $U(s)$, and $Y(s)$ are the Laplace transforms of the time-domain variables of $x(t)$, $u(t)$, and $y(t)$, respectively. Note that $s \in \mathbb{C}$. Then, formally eliminating $X(s)$ in the above equations, we arrive at the frequency-domain input-output relation $Y(s) = H(s)U(s)$. Here,

$$H(s) := D + C^T\, (sE - A)^{-1} B, \quad s \in \mathbb{C}, \tag{2.8}$$

is the so-called *transfer function* of the system (2.1) and (2.2). Note that

$$H\ :\ \mathbb{C} \mapsto (\mathbb{C} \cup \infty)^{p \times m},$$

is a $(p \times m)$-matrix-valued rational function.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

Similarly, the transfer function, H_n, of the reduced-order model (2.6) and (2.7) is given by

$$H_n(s) := D_n + C_n^T \, (sE_n - A_n)^{-1} B_n, \quad s \in \mathbb{C}. \tag{2.9}$$

Note that H_n is also a $(p \times m)$-matrix-valued rational function.

3. Padé and Padé-type models

The concept of transfer functions allows us to define reduced-order models by means of Padé or Padé-type approximation.

3.1. Padé approximants of transfer functions

Let $s_0 \in \mathbb{C}$ be any point such that s_0 is not a pole of the transfer function H given by (2.8). In practice, the point s_0 is chosen such that it is in some sense close to the frequency range of interest. We remark that the frequency range of interest is usually a subset of the imaginary axis in the complex s-plane. Since s_0 is not a pole of H, the function H admits the Taylor expansion

$$H(s) = \mu_0 + \mu_1 \, (s - s_0) + \mu_2 \, (s - s_0)^2 + \cdots + \mu_j \, (s - s_0)^j + \cdots \tag{3.1}$$

about s_0. The coefficients μ_j, $j = 0, 1, \ldots,$ in (3.1) are called the *moments* of H about the expansion point s_0. Note that each μ_j is a $p \times m$ matrix.

A reduced-order model (2.6) and (2.7) of state-space dimension n is called an nth *Padé model* (at the expansion point s_0) of the original system (2.1) and (2.2) if the Taylor expansions about s_0 of the transfer function (2.8), H, of the original system and the transfer function (2.9), H_n, of the reduced-order model agree in as many leading terms as possible, that is,

$$H(s) = H_n(s) + \mathcal{O}\big((s - s_0)^{q(n)}\big), \tag{3.2}$$

where $q(n)$ is as large as possible. For an introduction to Padé approximation, we refer the reader to Baker, Jr. and Graves-Morris (1996). In Feldmann and Freund (1995*b*) and Freund (1995), it was shown that

$$q(n) \geq \left\lfloor \frac{n}{m} \right\rfloor + \left\lfloor \frac{n}{p} \right\rfloor,$$

with equality in the ‘generic’ case.

Even though Padé models are defined via the local approximation property (3.2), in practice, they usually are excellent approximations over large frequency ranges. The following single-input single-output example illustrates this statement. The example is a circuit resulting from the so-called PEEC discretization (Ruehli 1974) of an electromagnetic problem. The circuit is an RCL network consisting of 2100 capacitors, 172 inductors, 6990 inductive couplings, and a single resistive source that drives the circuit.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

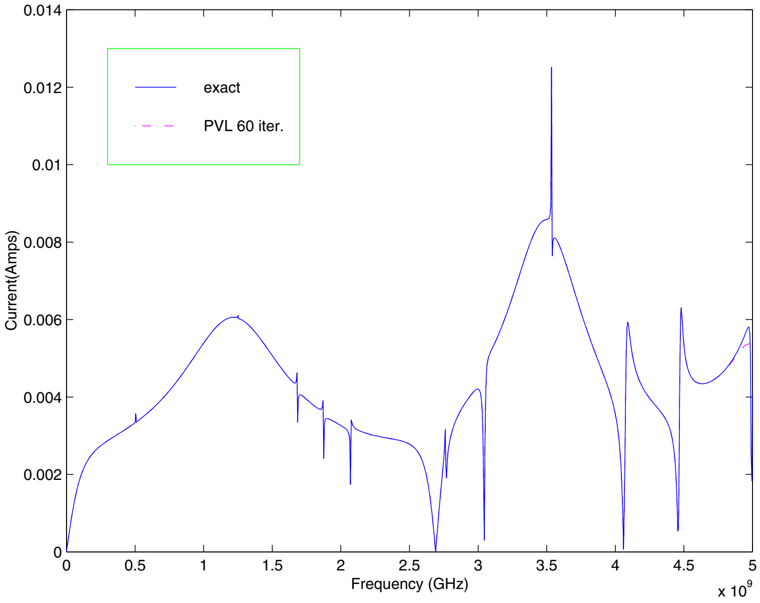

Figure 3.1. The PEEC transfer function, exact and Padé model of dimension $n = 60$.

Modified nodal analysis is used to set up the circuit equations, resulting in a linear dynamical system of dimension $N = 306$. In turns out that a Padé model of dimension $n = 60$ is sufficient to produce an almost exact transfer function in the relevant frequency range $s = 2\pi\,\mathrm{i}\,\omega$, $0 \leq \omega \leq 5 \times 10^9$. The corresponding curves for $|H(s)|$ and $|H_{60}(s)|$ are shown in Figure 3.1. The Padé model shown in Figure 3.1 was computed with the PVL method described in Section 5 below.

It is very tempting to compute Padé models by exploiting the definition (3.2) directly. More precisely, we would first explicitly generate the $q(n)$ moment matrices $\mu_0, \mu_1, \ldots, \mu_{q(n)-1}$, and then compute H_n and the system matrices in the reduced-order model (2.6) and (2.7) from these moments. In fact, for the special case $m = p = 1$ of single-input single-output systems, this approach is the *asymptotic waveform evaluation* (AWE) method that was introduced to the circuit simulation community by Pillage and Rohrer (1990). For surveys of AWE and its derivatives, we refer the reader to Chiprout and Nakhla (1994) and Raghavan, Rohrer, Pillage, Lee, Bracken and Alaybeyi (1993). However, computing Padé models directly from the moments is extremely ill-conditioned, and consequently, such an approach is not a viable numerical procedure in general. We discuss these shortcomings of the AWE approach in more detail in Section 3.4 below.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

3.2. Reduction to a single matrix

Instead of employing explicit moment matching, the preferred way to compute Padé models is to use Krylov-subspace techniques, such as a suitable Lanczos-type process, as we will describe in Section 5. This becomes possible after the transfer function (2.8) is rewritten in terms of a single matrix M, instead of the two matrices A and E. To this end, let

$$A - s_0 E = F_1 F_2, \quad \text{where} \quad F_1,\, F_2 \in \mathbb{C}^{N\times N}, \tag{3.3}$$

be any factorization of $A - s_0E$. For example, the matrices $A - s_0E$ arising in circuit simulation are large, but sparse, and are such that a sparse LU factorization is feasible. In this case, the matrices F_1 and F_2 in (3.3) are the lower and upper triangular factors, possibly with rows and columns permuted due to pivoting, of such a sparse LU factorization of $A - s_0E$. Using (3.3), the transfer function (2.8) can be rewritten as follows:

$$\begin{aligned} H(s) &= D + C^T\,(sE - A)^{-1}B \\ &= D - C^T\,(A - s_0E - (s - s_0)E)^{-1}B \\ &= D - L^T\,(I - (s - s_0)M)^{-1}R, \end{aligned} \tag{3.4}$$

where

$$M := F_1^{-1}EF_2^{-1}, \quad R := F_1^{-1}B, \quad \text{and} \quad L := F_2^{-T}C. \tag{3.5}$$

Note that (3.4) only involves one $N \times N$ matrix, namely M, instead of the two $N \times N$ matrices A and E in (2.8). This allows us to apply Krylov-subspace methods to the single matrix M, with the $N \times m$ matrix R and the $N \times p$ matrix L as blocks of right and left starting vectors.

3.3. Padé-type approximants

While Padé models often provide very good approximations in the frequency domain, they also have undesirable properties. In particular, Padé models in general do not preserve stability or passivity of the original system. However, by relaxing the Padé-approximation property (3.2), it is often possible to obtain stable or passive models. More precisely, we call a reduced-order model (2.6) and (2.7) of state-space dimension n an nth *Padé-type model* (at the expansion point s_0) of the original system (2.1) and (2.2) if the Taylor expansions about s_0 of the transfer functions H and H_n of the original system and the reduced-order system agree in a number of leading terms, that is,

$$H(s) = H_n(s) + \mathcal{O}\big((s - s_0)^{q'}\big),$$

where $1 \le q' < q(n)$. Recall that $q(n)$ denotes the optimal approximation order of a true Padé approximant.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

Unless $m = p = 1$, the transfer functions H and H_n are matrix-valued, and thus the Padé and Padé-type approximants underlying Padé and Padé-type models are so-called matrix-Padé and matrix-Padé-type approximants in general.

3.4. Explicit moment matching

In this subsection, we restrict ourselves to the single-input single-output case, $m = p = 1$. In this case, in (3.4), both R and L are vectors, and we set $r = R$ and $l = L$. Moreover, we assume that $D = 0$ in (3.4). Thus, (3.4) reduces to the representation

$$H(s) = -l^T \left(I - (s - s_0)M\right)^{-1} r. \tag{3.6}$$

Note that H is a scalar-valued rational function. Correspondingly, the nth Padé approximant H_n defined by (2.9) (with $D_n = 0$) and (3.2) is now also a scalar-valued rational function with numerator and denominator polynomial φ_{n-1} and ψ_n of degree at most $n - 1$ and n, respectively. Instead of (2.9), we represent H_n in terms of these polynomials:

$$H_n(s) = \frac{\varphi_{n-1}(s)}{\psi_n(s)}. \tag{3.7}$$

There are $2n$ free parameters in (3.7), namely the coefficients of the polynomials φ_{n-1} and ψ_n. Except for certain degenerate cases, these parameters can be chosen such that, in (3.2), the first $2n$ moments match:

$$H(s) = H_n(s) + \mathcal{O}\big((s - s_0)^{2n}\big) = \sum_{j=0}^{2n-1} \mu_j \, (s - s_0)^j + \mathcal{O}\big((s - s_0)^{2n}\big).$$

Here, the $\{\mu_j\}$ are the moments defined by the expansion (3.1). Using the representation (3.6) of H, the moments can be expressed as follows:

$$\mu_j = -l^T M^j r, \quad j = 0, 1, 2, \ldots. \tag{3.8}$$

The standard approach to computing H_n is based on the representation (3.7) and on explicit moment generation via (3.8). First, we use (3.8) to compute the leading $2n$ moments,

$$\mu_0, \mu_1, \ldots, \mu_{2n-1}, \tag{3.9}$$

of H, and from these, we then generate the coefficients of the polynomials φ_{n-1} and ψ_n in (3.7) by solving a system of linear equations with a Hankel matrix whose entries are the moments (3.9). This standard approach to computing H_n is employed in the AWE method (Pillage and Rohrer 1990). However, computing Padé approximants using explicit moment computations is inherently numerically unstable, and indeed, in practice, this approach can be employed in a meaningful way only for very moderate values

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

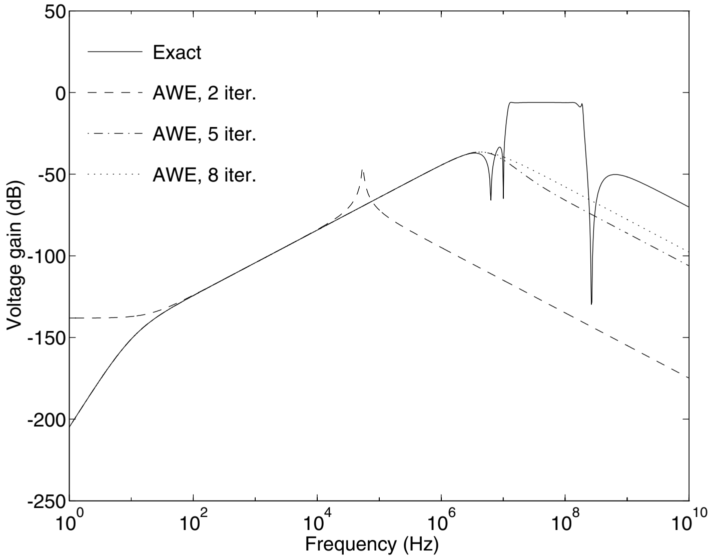

Figure 3.2. Results for simulation of voltage gain with AWE.

of n, such as $n \leq 10$; see Feldmann and Freund (1995a). As we will describe in more detail in Section 5, the numerical problems with AWE can easily be remedied by exploiting the Lanczos–Padé connection (Gragg 1974) and generating the Padé approximant H_n via the classical Lanczos process (Lanczos 1950). This approach was first introduced in Feldmann and Freund (1994) as the *Padé via Lanczos* (PVL) method; see also Gallivan, Grimme and Van Dooren (1994) and Feldmann and Freund (1995a).

While AWE and PVL are mathematically equivalent, their behaviour when run on an actual computer can be vastly different. The reason is that AWE is a numerically unstable algorithm and thus susceptible to round-off errors caused by finite-precision arithmetic. We illustrate the numerical differences between AWE and PVL with a circuit example taken from Feldmann and Freund (1994, 1995a). The circuit simulated here is a voltage filter, where the frequency range of interest is $1 \leq \omega \leq 10^{10}$. This example was first run with AWE, and in Figure 3.2 we show the computed function $|H_n(\mathrm{i}\,\omega)|$, for $n = 2$, 5, 8, together with the exact function $|H(\mathrm{i}\,\omega)|$, each for the frequency range $1 \leq \omega \leq 10^{10}$. Note that H_8 has clearly not yet converged to H. On the other hand, it turned out that the $\{H_n\}$ were hardly changing any more for $n \geq 8$, and in particular, AWE never converged in this example. In Figure 3.3 we show the computed results obtained with PVL for $n = 2$, 8, 28, together with the exact function $|H|$. Note that the results for $n = 8$ (the dotted curves) in Figures 3.1 and 3.2 are vastly different, although they both correspond to the same function H_8. The reason

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

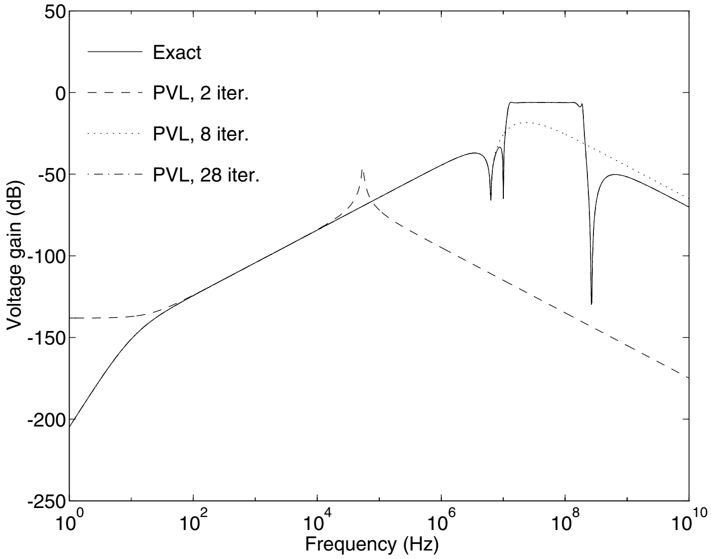

Figure 3.3. Results for simulation of voltage gain with PVL.

for this is that AWE is numerically unstable, while PVL generates H_8 stably. Furthermore, note that PVL converges, with the computed 28th Padé approximant being practically identical to H.

The main reason for the numerical instability of AWE is the explicit generation of the moments (3.9) by means of the formula (3.8). This computation is usually done as follows. We first generate the $2n$ vectors

$$r, M\,r, M^2\,r, \dots, M^{2n-1}\,r, \tag{3.10}$$

and then obtain (3.9) by computing the inner products

$$\mu_j = l^T \cdot (M^j\,r), \quad j = 0, 1, \dots, 2n-1, \tag{3.11}$$

of l with (3.10). An alternative is first to generate the vectors

$$r, M\,r, M^2\,r, \dots, M^{n-1}\,r \quad \text{and} \quad l, M^T\,l, (M^T)^2\,l, \dots, (M^T)^{n-1}\,l, \tag{3.12}$$

and then to obtain (3.9) by computing the inner products

$$\mu_{2j} = \big((M^T)^j\,l\big)^T \cdot (M^j\,r) \quad \text{and} \quad \mu_{2j+1} = \big((M^T)^j\,l\big)^T \cdot \big(M^{j+1}\,r\big) \tag{3.13}$$

for $j = 0, 1, \dots, n-1$. The problem is that the vectors (3.10) quickly converge to an eigenvector corresponding to a dominant eigenvalue of M. As a result, in finite-precision arithmetic, the moments μ_j computed via (3.11), even for rather moderate values of j, contain only information about this dominant eigenvalue. The Padé approximant H_n generated from the moments thus contains only information about part of the spectrum of M.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

The transfer function H, however, in general depends on all eigenvalues of M, and not just the dominant ones. This is the reason why, for AWE, the computed H_n usually does not converge to the transfer function H. The alternative approach suffers from the same problem since the two sequences of vectors in (3.12) quickly converge to a right, respectively left, eigenvector corresponding to a dominant eigenvalue of M.

Note that the space spanned by the first set of vectors in (3.12) is just the nth *right Krylov subspace*

$$\mathcal{K}_n(M, r) := \operatorname{span}\{\, r, M\, r, M^2\, r, \ldots, M^{n-1}\, r \,\} \tag{3.14}$$

induced by the matrix M and the right starting vector r. Similarly, the second set of vectors in (3.12) spans the nth *left Krylov subspace*

$$\mathcal{K}_n(M^T, l) := \operatorname{span}\{\, l, M^T\, l, (M^T)^2\, l, \ldots, (M^T)^{n-1}\, l \,\} \tag{3.15}$$

induced by the matrix M^T and the left starting vector l. While Krylov subspaces are very useful for large-scale matrix computations, the vectors in the definitions (3.14) and (3.15) are in general unsuitable as basis vectors. Indeed, as we just mentioned, they quickly converge, and in particular, they quickly become almost linearly dependent. The remedy is to construct more suitable basis vectors

$$v_1, v_2, \ldots, v_n, \ldots, \quad \text{and} \quad w_1, w_2, \ldots, w_n, \ldots, \tag{3.16}$$

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

$$\mathcal{K}_n(M, r) = \operatorname{span}\{\, v_1, v_2, \ldots, v_n \,\} \tag{3.17}$$

and

$$\mathcal{K}_n(M^T, l) = \operatorname{span}\{\, w_1, w_2, \ldots, w_n \,\}. \tag{3.18}$$

There are two main approaches for constructing basis vectors (3.16), the Lanczos algorithm and the Arnoldi process, which will be discussed in Sections 5 and 6, respectively.

Using the basis vectors (3.16), the explicit moment computations can now easily be avoided. Indeed, instead of the moments (3.9), we now compute so-called *modified moments*

$$w_j^T\, v_j \quad \text{and} \quad w_j^T\, M\, v_j, \quad j = 1, 2, \ldots, n. \tag{3.19}$$

In view of (3.17), (3.18), (3.14), (3.15), and (3.13), the modified moments (3.19) contain the very same information as the moments (3.9), and for each $j = 0, 1, \ldots, 2n-1$, the jth moment μ_j can be expressed as a suitable linear combination of the numbers (3.19).

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

4. Stability and passivity

In this section, we discuss the concepts of stability and passivity of linear dynamical systems.

4.1. Stability

An important property of linear dynamical systems is stability. An actual physical system needs to be stable in order to function properly. If a linear dynamical system (2.1) and (2.2) is used as a description of such a physical system, then clearly it should also be stable. Moreover, when a system (2.1) and (2.2) is replaced by a reduced-order model that is then used in a time-domain analysis, the reduced-order model also needs to be stable.

In this subsection, we present a brief discussion of stability of linear descriptor systems. For a more general survey of the various concepts of stability of dynamical systems, we refer the reader to Anderson and Vongpanitlerd (1973) and Willems (1970).

A descriptor system of the form (2.1) and (2.2) is said to be *stable* if its free-response, that is, the solutions $x(t)$, $t \geq 0$, of

$$\begin{aligned} E\frac{\mathrm{d}x}{\mathrm{d}t} &= Ax, \\ x(0) &= x_0, \end{aligned}$$

remain bounded as $t \to \infty$ for any possible initial vector x_0. Recall from the discussion in Section 2.1 that, for singular E, there are certain restrictions on the possible initial vectors x_0.

Stability can easily be characterized in terms of the finite eigenvalues of the matrix pencil $A - sE$; see, *e.g.*, Masubuchi, Kamitane, Ohara and Suda (1997). More precisely, we have the following theorem.

Theorem 4.1. The descriptor system (2.1) and (2.2) is stable if and only if the following two conditions are satisfied:

(i) all finite eigenvalues $s \in \mathbb{C}$ of the matrix pencil $A - sE$ satisfy $\operatorname{Re} s \leq 0$;

(ii) all finite eigenvalues s of $A - sE$ with $\operatorname{Re} s = 0$ are simple.

We stress that, in view of Theorem 4.1, the infinite eigenvalues of the matrix pencil $A - sE$ have no effect on stability. The reason is that these infinite eigenvalues result only in impulsive motions, which go to zero as $t \to \infty$.

Recall that the transfer function H of the descriptor system (2.1) and (2.2) is of the form

$$H(s) = D + C^T \left(sE - A\right)^{-1} B, \tag{4.1}$$

$$\text{where} \quad A,\, E \in \mathbb{R}^{N \times N}, \;\; B \in \mathbb{R}^{N \times m}, \;\; C \in \mathbb{R}^{N \times m}, \;\; \text{and } D \in \mathbb{R}^{p \times m}. \tag{4.2}$$

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

Note that any pole of H is necessarily an eigenvalue of the matrix pencil $A - sE$. Hence, it is tempting to determine stability via the poles of H. However, in general, not all eigenvalues of $A - sE$ are poles of H. For example, consider the system

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t} &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \\ y(t) &= \begin{bmatrix} 1 & 1 \end{bmatrix} x(t),\end{aligned}$$

which is taken from Anderson and Vongpanitlerd (1973). The pencil associated with this system is

$$A - sI = \begin{bmatrix} 1-s & 0 \\ 0 & -1-s \end{bmatrix}.$$

Its eigenvalues are ± 1, and hence this system is unstable. The transfer function $H(s) = 1/(s+1)$, however, only has the 'stable' pole -1. Therefore, checking conditions (i) and (ii) of Theorem 4.1 only for the poles of H is, in general, not enough to guarantee stability. In order to infer stability of the system (2.1) and (2.2) from the poles of its transfer function, we need an additional condition, which we formulate next.

Let H be a given $(p \times m)$-matrix-valued rational function. Any representation of H of the form (4.1) with matrices (4.2) is called a *realization* of H. Furthermore, a realization (4.1) of H is said to be *minimal* if the dimension N of the matrices (4.2) is as small as possible. We will also say that the state-space description (2.1) and (2.2) is a minimal realization if its transfer function (4.1) is a minimal realization.

The following theorem is the well-known characterization of minimal realizations in terms of conditions on the matrices (4.2); see, *e.g.*, Verghese *et al.* (1981). We also refer the reader to the related results on controllability, observability, and minimal realizations of descriptor systems given in Chapter 2 of Dai (1989).

Theorem 4.2. Let H be a $(p \times m)$-matrix-valued rational function given by a realization (4.1). Then, (4.1) is a minimal realization of H if and only if the matrices (4.2) satisfy the following five conditions:

(i) $\operatorname{rank} \begin{bmatrix} A - sE & B \end{bmatrix} = N$ for all $s \in \mathbb{C}$ (finite controllability);

(ii) $\operatorname{rank} \begin{bmatrix} E & B \end{bmatrix} = N$ (infinite controllability);

(iii) $\operatorname{rank} \begin{bmatrix} A^T - sE^T & C \end{bmatrix} = N$ for all $s \in \mathbb{C}$ (finite observability);

(iv) $\operatorname{rank} \begin{bmatrix} E^T & C \end{bmatrix} = N$ (infinite observability);

(v) $A \ker(E) \subseteq \operatorname{Im}(E)$ (absence of nondynamic modes).

For descriptor systems given by a minimal realization, stability can indeed be checked via the poles of its transfer function.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

Theorem 4.3. Let (2.1) and (2.2) be a minimal realization of a descriptor system, and let H be its transfer function (4.1). Then, the descriptor system (2.1) and (2.2) is stable if and only if all finite poles s_i of H satisfy $\operatorname{Re} s_i \leq 0$ and any pole with $\operatorname{Re} s_i = 0$ is simple.

4.2. Passivity

In circuit simulation, reduced-order modelling is often applied to large passive linear subcircuits, such as RCL networks consisting of only resistors, capacitors, and inductors. 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.*, Chirlian (1967), Rohrer and Nosrati (1981), and Lozano, Brogliato, Egeland and Maschke (2000). Therefore, it is important to have techniques to check the passivity of a given reduced-order model.

Roughly speaking, a system is *passive* if it does not generate energy. For descriptor systems of the form (2.1) and (2.2), passivity is equivalent to positive realness of the transfer function. Moreover, such systems can only be passive if they have identical numbers of inputs and outputs. Thus, for the remainder of this subsection, we assume that $m = p$. Then, a system described by (2.1) and (2.2) is passive, that is, it does not generate energy, if and only if its transfer function (4.1) is *positive real*; see, *e.g.*, Anderson and Vongpanitlerd (1973). A precise definition of positive realness is as follows.

Definition 1. An $(m \times m)$-matrix-valued function $H : \mathbb{C} \mapsto (\mathbb{C} \cup \infty)^{m \times m}$ is called *positive real* if the following three conditions are satisfied:

(i) H is analytic in $\mathbb{C}_+ := \{\, s \in \mathbb{C} \mid \operatorname{Re} s > 0 \,\}$;

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

(iii) $H(s) + (H(s))^H \succeq 0$ for all $s \in \mathbb{C}_+$.

In Definition 1 and hereafter, the notation $M \succeq 0$ means that the matrix M is Hermitian positive semi-definite. Similarly, $M \preceq 0$ means that M is Hermitian negative semi-definite.

For transfer functions H of the form (4.1), condition (ii) of Definition 1 is always satisfied since the matrices (4.2) are assumed to be real. Furthermore, condition (i) simply means that H cannot have poles in $\mathbb{C}_+$, and this can be checked easily. For the special case $m = 1$ of scalar-valued functions H, condition (iii) states that the real part of $H(s)$ is nonnegative for all s with nonnegative real part. In order to check this condition, it is sufficient to show that the real part of $H(s)$ is nonnegative for all purely imaginary s. This can be done by means of relatively elementary means. For example, in Bai and Freund (2000), a procedure based on eigenvalue computations

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

is proposed. For the general matrix-valued case, $m \geq 1$, however, checking condition (iii) is much more involved. One possibility is to employ a suitable extension of the classical positive real lemma (Anderson 1967, Anderson and Vongpanitlerd 1973, Zhou, Doyle and Glover 1996) that characterizes positive realness of regular linear systems via the solvability of certain linear matrix inequalities (LMIs). Such a version of the positive real lemma for general descriptor systems is stated in Theorem 4.4 below.

We remark that any matrix-valued rational function H has an expansion about $s = \infty$ of the form

$$H(s) = \sum_{j=-\infty}^{j_0} M_j s^j, \tag{4.3}$$

where $j_0 \geq 0$ is an integer. Moreover, the function H has a pole at $s = \infty$ if and only if $j_0 \geq 1$ and $M_{j_0} \neq 0$ in (4.3).

A suitable extension of the classical positive real lemma for regular systems to descriptor systems can now be stated as follows.

Theorem 4.4. (Positive real lemma for descriptor systems) Let H be a real $(m \times m)$-matrix-valued rational function of the form (4.1) with matrices (4.2).

(a) (Sufficient condition.) If the LMIs

$$\begin{bmatrix} A^T X + X^T A & X^T B - C \\ B^T X - C^T & -D - D^T \end{bmatrix} \preceq 0 \quad \text{and} \quad E^T X = X^T E \succeq 0 \tag{4.4}$$

have a solution $X \in \mathbb{R}^{N \times N}$, then H is positive real.

(b) (Necessary condition.) Suppose that (4.1) is a minimal realization of H and that the matrix M_0 in the expansion (4.3) satisfies

$$(D - M_0) + (D - M_0)^T \succeq 0. \tag{4.5}$$

If H is positive real, then there exists a solution $X \in \mathbb{R}^{N \times N}$ of the LMIs (4.4).

A proof of Theorem 4.4 can be found in Freund and Jarre (2000).

The result of Theorem 4.4 allows us to check positive realness by solving semi-definite programming problems of the form (4.4). Note that there are N^2 unknowns in (4.4), namely the entries of the $N \times N$ matrix X. Problems of the form (4.4) can be tackled with interior-point methods; see, *e.g.*, Boyd, El Ghaoui, Feron and Balakrishnan (1994) and Freund and Jarre (2003). However, the computational complexity of these methods grows quickly with N, and thus, these methods are viable only for rather small values of N. On the other hand, it is usually known whether a given system is passive, and the need to numerically check passivity mainly arises for reduced-order models of a given passive model. In this case, the dimension N of the

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

semi-definite programming problem (4.4) is equal to the dimension of the reduced-order model, which is usually small enough for techniques based on Theorem 4.4 to become feasible.

For the special case $E = I$, the result of Theorem 4.4 is just the classical positive real lemma (Anderson 1967, Anderson and Vongpanitlerd 1973, Zhou *et al.* 1996). In this case, (4.4) reduces to the problem of finding a symmetric positive semi-definite matrix $X \in \mathbb{R}^{N \times N}$ such that

$$\begin{bmatrix} A^T X + XA & XB - C \\ B^T X - C^T & -D - D^T \end{bmatrix} \preceq 0.$$

Moreover, if $E = I$, the condition (4.5) is always satisfied, since in this case $M_0 = 0$ and $D + D^T \succeq 0$.

4.3. Linear RCL subcircuits

In circuit simulation, an important special case of passive circuits is linear subcircuits that consist only of resistors, capacitors, and inductors. Such linear RCL subcircuits arise in the modelling of a circuit's interconnect and pin package; see, *e.g.*, Cheng *et al.* (2000), Freund and Feldmann (1997, 1998), Kim, Gopal and Pillage (1994), and Pileggi (1995).

The equations describing linear RCL subcircuits are of the form (2.1) and (2.2) with $D = 0$ and $m = p$. Furthermore, the equations can be formulated such that the matrices A, $E \in \mathbb{R}^{N \times N}$ in (2.1) are symmetric and exhibit a block structure; see Freund and Feldmann (1996*a*, 1998). More precisely, we have

$$A = A^T = \begin{bmatrix} -A_{11} & A_{12} \\ A_{12}^T & 0 \end{bmatrix} \quad \text{and} \quad E = E^T = \begin{bmatrix} E_{11} & 0 \\ 0 & -E_{22} \end{bmatrix}, \tag{4.6}$$

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

If the RCL subcircuit is viewed as an m-terminal component with m inputs and $m = p$ outputs, then the matrices B and C in (2.1) and (2.2) are identical and of the form

$$B = C = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \quad \text{with} \quad B_1 \in \mathbb{R}^{N_1 \times m}. \tag{4.7}$$

In view of (4.6) and (4.7), the transfer function of such an m-terminal RCL subcircuit is given by

$$H(s) = B^T (sE - A)^{-1} B, \quad \text{where} \quad A = A^T, \quad E = E^T. \tag{4.8}$$

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

We call a transfer function H *symmetric* if it is of the form (4.8) with real matrices A, E, and B.

We will also use the following nonsymmetric formulation of (4.8). Let J be the block matrix

$$J = \begin{bmatrix} I_{N_1} & 0 \\ 0 & -I_{N_2} \end{bmatrix}, \tag{4.9}$$

where I_{N_1} and I_{N_2} are the $N_1 \times N_1$ and $N_2 \times N_2$ identity matrix, respectively.

Note that, by (4.7) and (4.9), we have $B = JB$. Using this relation, as well as (4.6), we can rewrite (4.8) as follows:

$$\begin{aligned} &H(s) = B^T \bigl(s\widetilde{E} - \widetilde{A}\bigr)^{-1} B, \\ \text{where} \quad &\widetilde{A} = \begin{bmatrix} -A_{11} & A_{12} \\ -A_{12}^T & 0 \end{bmatrix}, \quad \widetilde{E} = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix}. \end{aligned} \tag{4.10}$$

In this formulation, the matrix $\widetilde{A}$ is no longer symmetric, but now

$$\widetilde{A} + \widetilde{A}^T \preceq 0 \quad \text{and} \quad \widetilde{E} \succeq 0. \tag{4.11}$$

It turns out that the properties (4.11) are the key to ensure positive realness. Indeed, in Freund (2000*b*), we established the following result.

Theorem 4.5. *Let $\widetilde{A}$, $\widetilde{E} \in \mathbb{R}^{N \times N}$, and $B \in \mathbb{R}^{N \times m}$. Assume that $\widetilde{A}$ and $\widetilde{E}$ satisfy (4.11), and that the matrix pencil $\widetilde{A} - s\widetilde{E}$ is regular. Then, the $(m \times m)$-matrix-valued function*

$$H(s) = B^T \bigl(s\widetilde{E} - \widetilde{A}\bigr)^{-1} B$$

is positive real.

5. Approaches based on Lanczos-type methods

In this section, we discuss the use of Lanczos-type methods for the construction of Padé and Padé-type reduced-order models of time-invariant linear dynamical systems.

5.1. *Block Krylov subspaces*

We consider general descriptor systems of the form (2.1) and (2.2). As discussed in Section 3.2, the key to using Krylov-subspace techniques for reduced-order modelling of such systems is to first replace the matrix pair A and E by a single matrix M. To this end, let $s_0 \in \mathbb{C}$ be any given point such that the matrix $A - s_0 E$ is nonsingular. Then, with M, R, and L denoting the matrices defined in (3.5), the linear system (2.1) and (2.2) can

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

be rewritten in the following form:

$$M\frac{\mathrm{d}\tilde{x}}{\mathrm{d}t} = (I + s_0 M)\,\tilde{x} + Ru(t), \tag{5.1}$$

$$y(t) = L^T\tilde{x}(t) + Du(t). \tag{5.2}$$

Here, $\tilde{x} := F_2 x$, where F_2 is the matrix from the factorization (3.3). Note that $M \in \mathbb{C}^{N\times N}$, $R \in \mathbb{C}^{N\times m}$, and $L \in \mathbb{C}^{N\times p}$, where N is the state-space dimension of the system, m is the number of inputs, and p is the number of outputs.

The transfer function H of the rewritten system (5.1) and (5.2) is given by (3.4). By expanding (3.4) about s_0, we obtain

$$H(s) = D - \sum_{j=0}^{\infty} L^T M^j R\,(s - s_0)^j. \tag{5.3}$$

Recall from Section 3 that Padé and Padé-type reduced-order models are defined via the leading coefficients of an expansion of H about s_0. In view of (5.3), the jth coefficient of such an expansion can be expressed as follows:

$$-L^T M^j R = -\big((M^{j-i})^T L\big)^T \big(M^i R\big), \quad i = 0, 1, \ldots, j. \tag{5.4}$$

Notice that the factors on the right-hand side of (5.4) are blocks of the *right* and *left block Krylov matrices*

$$\begin{aligned} &\begin{bmatrix} R & MR & M^2R & \cdots & M^iR & \cdots \end{bmatrix} \\ \text{and} \quad &\Big[L \quad M^TL \quad \big(M^T\big)^2 L \quad \cdots \quad \big(M^T\big)^k L \quad \cdots \Big], \end{aligned} \tag{5.5}$$

respectively. As a result, all the information needed to generate Padé and Padé-type reduced-order models is contained in the block Krylov matrices (5.5). However, simply computing the blocks M^iR and $\big(M^T\big)^i L$ in (5.5) and then generating the leading coefficients of the expansion (5.3) from these blocks is not a viable numerical procedure. The reason is that, in finite-precision arithmetic, as i increases, the blocks M^iR and $\big(M^T\big)^i L$ quickly contain only information about the eigenspaces of the dominant eigenvalue of M. Instead, we need to employ suitable Krylov-subspace methods that generate numerically better basis vectors for the subspaces associated with the block Krylov matrices (5.5).

Next, we give a formal definition of the subspaces induced by (5.5). For the special case $m = p = 1$ of single-input single-output systems, the 'blocks' of the Krylov matrices (5.5) reduce to vectors, and the Krylov subspaces spanned by these vectors are just the standard Krylov subspaces that we introduced in (3.17) and (3.18). For the general case $m,\ p \geq 1$ of multi-input multi-output systems, however, the definition of subspaces induced by (5.5) is more involved. First, note that each block M^iR consists of m column

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

vectors of length N. By scanning these column vectors of the right block Krylov matrix in (5.5) from left to right and by deleting any column that is linearly dependent on columns to its left, we obtain the *deflated* right block Krylov matrix

$$\left[\, R_1 \quad M R_2 \quad M^2 R_3 \quad \cdots \quad M^{i_{\max}-1} R_{i_{\max}} \,\right]. \tag{5.6}$$

This process of detecting and deleting the linearly dependent columns is called *exact deflation*. We remark that the matrix (5.6) is finite, since at most N of the column vectors can be linearly independent. Furthermore, a column $M^i r$ being linearly dependent on columns to its left in (5.5) implies that any column $M^{i'} r$, $i' \geq i$, is linearly dependent on columns to its right. Therefore, in (5.6), for each $i = 1, 2, \ldots, i_{\max}$, the matrix R_i is a submatrix of R_{i-1}, where, for $i = 1$, we set $R_0 = R$.

Let m_i denote the number of columns of R_i. The matrix (5.6) thus has

$$n_{\max}^{(\mathrm{r})} := m_1 + m_2 + \cdots + m_{i_{\max}},$$

columns. For each integer n with $1 \leq n \leq n_{\max}^{(\mathrm{r})}$, we define the nth *right block Krylov subspace* $\mathcal{K}_n(M, R)$ (induced by M and R) as the subspace spanned by the first n columns of the deflated right block Krylov matrix (5.6).

Analogously, by deleting the linearly independent columns of the left block Krylov matrix in (5.5), we obtain a deflated left block Krylov matrix of the form

$$\left[\, L_1 \quad M^T L_2 \quad \left(M^T\right)^2 L_3 \quad \cdots \quad \left(M^T\right)^{i_{\max}-1} L_{k_{\max}} \,\right]. \tag{5.7}$$

Let $n_{\max}^{(\mathrm{l})}$ be the number of columns of the matrix (5.7). Then, for each integer n with $1 \leq n \leq n_{\max}^{(\mathrm{l})}$, we define the nth *left block Krylov subspace* $\mathcal{K}_n(M^T, L)$ (induced by M^T and L) as the subspace spanned by the first n columns of the deflated left block Krylov matrix (5.7).

For a more detailed discussion of block Krylov subspaces and deflation, we refer the reader to Aliaga, Boley, Freund and Hernández (2000) and Freund (2000*b*).

Next, we discuss reduced-order modelling approaches that employ Lanczos and Lanczos-type methods for the construction of suitable basis vectors for the right and left block Krylov subspaces $\mathcal{K}_n(M, R)$ and $\mathcal{K}_n(M^T, L)$.

5.2. The MPVL algorithm

For the special case $m = p = 1$ of single-input single-output linear dynamical systems, each of the 'blocks' R and L only consists of a single vector, say r and l, and $\mathcal{K}_n(M, r)$ and $\mathcal{K}_n(M^T, l)$ are just the standard nth right and left Krylov subspaces induced by single vectors. The classical Lanczos process (Lanczos 1950) is a well-known procedure for computing two sets of bi-orthogonal basis vectors for $\mathcal{K}_n(M, r)$ and $\mathcal{K}_n(M^T, l)$. Moreover,

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

these vectors are generated by means of three-term recurrences the coefficients of which define a tridiagonal matrix T_n. It turns out that T_n contains all the information that is needed to set up an nth Padé reduced-order model of a given single-input single-output time-invariant linear dynamical system. The associated computational procedure is called the *Padé via Lanczos* (PVL) algorithm (Feldmann and Freund 1994, 1995*a*).

Here, we describe in some detail an extension of the PVL algorithm to the case of general m-input p-output time-invariant linear dynamical systems. The underlying block Krylov subspace method is the *nonsymmetric band Lanczos algorithm* (Freund 2000*a*) for constructing two sets of right and left Lanczos vectors,

$$v_1, v_2, \dots, v_n \quad \text{and} \quad w_1, w_2, \dots, w_n, \tag{5.8}$$

respectively. These vectors span the nth right and left block Krylov subspaces (induced by M and R, and M^T and L, respectively):

$$\begin{aligned} \operatorname{span}\{\, v_1, v_2, \dots, v_n \,\} &= \mathcal{K}_n(M, R) \\ \text{and} \quad \operatorname{span}\{\, w_1, w_2, \dots, w_n \,\} &= \mathcal{K}_n(M^T, L). \end{aligned} \tag{5.9}$$

Moreover, the vectors (5.8) are constructed to be bi-orthogonal:

$$w_j^T v_k = \begin{cases} 0 & \text{if } j \neq k, \\ \delta_j & \text{if } j = k, \end{cases} \quad \text{for all} \quad j, k = 1, 2, \dots, n. \tag{5.10}$$

It turns out that the Lanczos vectors (5.8) can be constructed by means of recurrence relations of length at most $m+p+1$. The recurrence coefficients for the first n right Lanczos vectors define an $n \times n$ matrix $T_n^{(\mathrm{pr})}$ that is 'essentially' a band matrix with total bandwidth $m+p+1$. Similarly, the recurrence coefficients for the first n left Lanczos vectors define an $n \times n$ band matrix $\widetilde{T}_n^{(\mathrm{pr})}$ with total bandwidth $m+p+1$. For a more detailed discussion of the structure of $T_n^{(\mathrm{pr})}$ and $\widetilde{T}_n^{(\mathrm{pr})}$, we refer the reader to Aliaga *et al.* (2000) and Freund (2000*a*).

Algorithm 5.1 below gives a complete description of the numerical procedure that generates the Lanczos vectors (5.8) with properties (5.9) and (5.10). In order to obtain a Padé reduced-order model based on this algorithm, we do not need the Lanczos vectors themselves, but rather the matrix of right recurrence coefficients $T_n^{(\mathrm{pr})}$, the matrices $\rho_n^{(\mathrm{pr})}$ and $\eta_n^{(\mathrm{pr})}$ that contain the recurrence coefficients from processing the starting blocks R and L, respectively, and the diagonal matrix

$$\Delta_n = \operatorname{diag}(\delta_1, \delta_2, \dots, \delta_n),$$

whose diagonal entries are the δ_j's from (5.10). The following algorithm produces the matrices $T_n^{(\mathrm{pr})}$, $\rho_n^{(\mathrm{pr})}$, $\eta_n^{(\mathrm{pr})}$, and Δ_n as output.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

Algorithm 5.1. (Nonsymmetric band Lanczos algorithm)

```
INPUT: A matrix $M \in \mathbb{C}^{N\times N}$;
       A block of $m$ right starting vectors $R = \begin{bmatrix} r_1 & r_2 & \cdots & r_m \end{bmatrix} \in \mathbb{C}^{N\times m}$;
       A block of $p$ left starting vectors $L = \begin{bmatrix} l_1 & l_2 & \cdots & l_p \end{bmatrix} \in \mathbb{C}^{N\times p}$.
OUTPUT: The $n \times n$ Lanczos matrix $T_n^{(\mathrm{pr})}$, and the matrices $\rho_n^{(\mathrm{pr})}$, $\eta_n^{(\mathrm{pr})}$,
        and $\Delta_n$.

(0) For $k = 1, 2, \ldots, m$, set $\hat{v}_k = r_k$.
    For $k = 1, 2, \ldots, p$, set $\hat{w}_k = l_k$.
    Set $m_{\mathrm{c}} = m$, $p_{\mathrm{c}} = p$, and $\mathcal{I}_v = \mathcal{I}_w = \emptyset$.

For $n = 1, 2, \ldots,$ until convergence or $m_{\mathrm{c}} = 0$ or $p_{\mathrm{c}} = 0$ or $\delta_n = 0$ do:

(1) (If necessary, deflate $\hat{v}_n$.)
    Compute $\|\hat{v}_n\|_2$.
    Decide if $\hat{v}_n$ should be deflated. If yes, do the following:

    (a) Set $\hat{v}_{n-m_{\mathrm{c}}}^{\mathrm{defl}} = \hat{v}_n$ and store this vector. Set $\mathcal{I}_v = \mathcal{I}_v \cup \{\, n - m_{\mathrm{c}} \,\}$.
    (b) Set $m_{\mathrm{c}} = m_{\mathrm{c}} - 1$. If $m_{\mathrm{c}} = 0$, set $n = n - 1$ and stop.
    (c) For $k = n, n+1, \ldots, n + m_{\mathrm{c}} - 1$, set $\hat{v}_k = \hat{v}_{k+1}$.
    (d) Repeat all of step (1).

(2) (If necessary, deflate $\hat{w}_n$.)
    Compute $\|\hat{w}_n\|_2$.
    Decide if $\hat{w}_n$ should be deflated. If yes, do the following:

    (a) Set $\hat{w}_{n-p_{\mathrm{c}}}^{\mathrm{defl}} = \hat{w}_n$ and store this vector. Set $\mathcal{I}_w = \mathcal{I}_w \cup \{\, n - p_{\mathrm{c}} \,\}$.
    (b) Set $p_{\mathrm{c}} = p_{\mathrm{c}} - 1$. If $p_{\mathrm{c}} = 0$, set $n = n - 1$ and stop.
    (c) For $k = n, n+1, \ldots, n + p_{\mathrm{c}} - 1$, set $\hat{w}_k = \hat{w}_{k+1}$.
    (d) Repeat all of step (2).

(3) (Normalize $\hat{v}_n$ and $\hat{w}_n$ to obtain $v_n$ and $w_n$.)
    Set

        $t_{n,n-m_{\mathrm{c}}} = \|\hat{v}_n\|_2, \quad \tilde{t}_{n,n-p_{\mathrm{c}}} = \|\hat{w}_n\|_2,$

        $v_n = \dfrac{\hat{v}_n}{t_{n,n-m_{\mathrm{c}}}}, \quad \text{and} \quad w_n = \dfrac{\hat{w}_n}{\tilde{t}_{n,n-p_{\mathrm{c}}}}.$

(4) (Compute $\delta_n$ and check for possible breakdown.)
    Set $\delta_n = w_n^T v_n$. If $\delta_n = 0$, set $n = n - 1$ and stop.
```

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

```
(5) (Orthogonalize the right candidate vectors against $w_n$.)
    For $k = n+1, n+2, \ldots, n+m_c-1$, set
        $t_{n,k-m_c} = \frac{w_n^T \hat{v}_k}{\delta_n}$   and   $\hat{v}_k = \hat{v}_k - v_n\, t_{n,k-m_c}$.
(6) (Orthogonalize the left candidate vectors against $v_n$.)
    For $k = n+1, n+2, \ldots, n+p_c-1$, set
        $\tilde{t}_{n,k-p_c} = \frac{\hat{w}_k^T v_n}{\delta_n}$   and   $\hat{w}_k = \hat{w}_k - w_n\, \tilde{t}_{n,k-p_c}$.
(7) (Advance the right block Krylov subspace to get $\hat{v}_{n+m_c}$.)
    (a) Set $\hat{v}_{n+m_c} = M\, v_n$.
    (b) For $k \in \mathcal{I}_w$ (in ascending order), set
            $\tilde{\sigma} = \bigl(\hat{w}_k^{\mathrm{defl}}\bigr)^T v_n$,   $\tilde{t}_{n,k} = \frac{\tilde{\sigma}}{\delta_n}$,
        and, if $k > 0$, set
            $t_{k,n} = \frac{\tilde{\sigma}}{\delta_k}$   and   $\hat{v}_{n+m_c} = \hat{v}_{n+m_c} - v_k\, t_{k,n}$.
    (c) Set $k_v = \max\{\, 1,\ n - p_c \,\}$.
    (d) For $k = k_v, k_v+1, \ldots, n-1$, set
            $t_{k,n} = \tilde{t}_{n,k}\, \frac{\delta_n}{\delta_k}$   and   $\hat{v}_{n+m_c} = \hat{v}_{n+m_c} - v_k\, t_{k,n}$.
    (e) Set
            $t_{n,n} = \frac{w_n^T \hat{v}_{n+m_c}}{\delta_n}$   and   $\hat{v}_{n+m_c} = \hat{v}_{n+m_c} - v_n\, t_{n,n}$.
(8) (Advance the left block Krylov subspace to get $\hat{w}_{n+p_c}$.)
    (a) Set $\hat{w}_{n+p_c} = M^T w_n$.
    (b) For $k \in \mathcal{I}_v$ (in ascending order), set
            $\sigma = w_n^T \hat{v}_k^{\mathrm{defl}}$,   $t_{n,k} = \frac{\sigma}{\delta_n}$,
        and, if $k > 0$, set
            $\tilde{t}_{k,n} = \frac{\sigma}{\delta_k}$   and   $\hat{w}_{n+p_c} = \hat{w}_{n+p_c} - w_k\, \tilde{t}_{k,n}$.
    (c) Set $k_w = \max\{\, 1,\ n - m_c \,\}$.
    (d) For $k = k_w, k_w+1, \ldots, n-1$, set
            $\tilde{t}_{k,n} = t_{n,k}\, \frac{\delta_n}{\delta_k}$   and   $\hat{w}_{n+p_c} = \hat{w}_{n+p_c} - w_k\, \tilde{t}_{k,n}$.
```

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

(e) Set

$$\tilde{t}_{n,n} = t_{n,n} \quad \text{and} \quad \hat{w}_{n+p_c} = \hat{w}_{n+p_c} - w_n \, \tilde{t}_{n,n}.$$

(9) Set

$$\begin{aligned} T_n^{(\mathrm{pr})} &= \bigl[t_{i,k}\bigr]_{i,k=1,2,\ldots,n}, \\ \rho_n^{(\mathrm{pr})} &= \bigl[t_{i,k-m}\bigr]_{i=1,2,\ldots,n;\, k=1,2,\ldots,k_\rho} \quad \text{where} \quad k_\rho = m + \min\{\,0,\, n - m_c\,\}, \\ \eta_n^{(\mathrm{pr})} &= \bigl[\tilde{t}_{i,k-p}\bigr]_{i=1,2,\ldots,n;\, k=1,2,\ldots,k_\eta} \quad \text{where} \quad k_\eta = p + \min\{\,0,\, n - p_c\,\}, \\ \Delta_n &= \operatorname{diag}\bigl(\delta_1, \delta_2, \ldots, \delta_n\bigr). \end{aligned}$$

(10) Check if n is large enough. If yes, stop.

Remark 1. When applied to single starting vectors, that is, for the special case $m = p = 1$, Algorithm 5.1 reduces to the classical nonsymmetric Lanczos process (Lanczos 1950).

Remark 2. It can be shown that, at step n of Algorithm 5.1, exact deflation of a vector in the right, respectively left, block Krylov matrix (5.5) occurs if and only if $\hat{v}_n = 0$, respectively $\hat{w}_n = 0$, in step (1), respectively step (2). Therefore, to run Algorithm 5.1 with exact deflation only, we deflate $\hat{v}_n$ if $\|\hat{v}_n\|_2 = 0$ in step (1), and we deflate $\hat{w}_n$ if $\|\hat{w}_n\|_2 = 0$ in step (2). In finite-precision arithmetic, however, so-called *inexact deflation* is employed. This means that, in step (1), $\hat{v}_n$ is deflated if $\|\hat{v}_n\|_2 \le \varepsilon$, and, in step (2), $\hat{w}_n$ is deflated if $\|\hat{w}_n\|_2 \le \varepsilon$, where $\varepsilon = \varepsilon(M) > 0$ is a suitably chosen small constant.

Remark 3. The occurrence of $\delta_n = 0$ in step (4) of Algorithm 5.1 is called a *breakdown*. In finite-precision arithmetic, in step (4) we should also check for *near-breakdowns*, that is, if $\delta_n \approx 0$. In general, it cannot be excluded that breakdowns or near-breakdowns occur, although they are very unlikely. Furthermore, by using so-called *look-ahead* techniques, it is possible to remedy the problem of possible breakdowns or near-breakdowns. For the sake of simplicity, we have stated the band Lanczos algorithm without look-ahead only. A look-ahead version of Algorithm 5.1 is described in Aliaga *et al.* (2000).

The matrices $T_n^{(\mathrm{pr})}$, $\rho_n^{(\mathrm{pr})}$, and $\eta_n^{(\mathrm{pr})}$ produced by Algorithm 5.1 can be viewed as oblique projections of the input data M, R, and L onto the right block Krylov subspace $\mathcal{K}_n(M, R)$ and orthogonally to the left block Krylov subspace $\mathcal{K}_n(M^T, L)$. To give a precise statement of these projection properties, we let

$$V_n := \bigl[v_1 \quad v_2 \quad \cdots \quad v_n\bigr] \quad \text{and} \quad W_n := \bigl[w_1 \quad w_2 \quad \cdots \quad w_n\bigr] \tag{5.11}$$

denote the matrices whose columns are the first n right and left Lanczos

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

vectors, respectively. Then the matrices $T_n^{(\mathrm{pr})}$, $\rho_n^{(\mathrm{pr})}$, $\eta_n^{(\mathrm{pr})}$, and Δ_n generated by Algorithm 5.1 are related to the input data M, R, and L as follows:

$$\begin{aligned} T_n^{(\mathrm{pr})} &= \Delta_n^{-1} W_n^T M V_n, \\ \rho_n^{(\mathrm{pr})} &= \Delta_n^{-1} W_n^T R, \\ \eta_n^{(\mathrm{pr})} &= \Delta_n^{-T} V_n^T L, \\ \Delta_n &= W_n^T V_n. \end{aligned} \tag{5.12}$$

The relations (5.12) can be employed to set up a reduced-order model of dimension n of the linear system (5.1) and (5.2). To this end, we restrict the state vector $\tilde{x}(t)$ in (5.1) and (5.2) to vectors in $\mathcal{K}_n(M,R)$. In view of (5.11), these restricted vectors can be written as

$$\tilde{x}(t) = V_n z(t), \tag{5.13}$$

where $z(t)$ has length n. By applying the oblique projections stated in (5.12) to the linear dynamical system (5.1) and (5.2), and by using (5.13), we obtain the following reduced-order model:

$$T_n^{(\mathrm{pr})} \frac{\mathrm{d}z}{\mathrm{d}t} = \bigl(I + s_0 T_n^{(\mathrm{pr})}\bigr) z + \rho_n^{(\mathrm{pr})} u(t), \tag{5.14}$$

$$y(t) = \bigl(\eta_n^{(\mathrm{pr})}\bigr)^T \Delta_n z(t) + D u(t). \tag{5.15}$$

Note that the transfer function of this reduced-order model is given by

$$H_n(s) = D - \bigl(\eta_n^{(\mathrm{pr})}\bigr)^T \Delta_n \bigl(I - (s - s_0) T_n^{(\mathrm{pr})}\bigr)^{-1} \rho_n^{(\mathrm{pr})}. \tag{5.16}$$

The *matrix-Padé via Lanczos* (MPVL) algorithm (Feldmann and Freund 1995*b*, Freund 1995) consists of applying Algorithm 5.1 to the matrices M, R, and L defined in (3.5), and running it for n steps. The matrices $T_n^{(\mathrm{pr})}$, $\rho_n^{(\mathrm{pr})}$, $\eta_n^{(\mathrm{pr})}$, and Δ_n produced by Algorithm 5.1 are then used to set up the reduced-order model (5.14) and (5.15) of the original linear dynamical system (2.1) and (2.2).

It turns out that the reduced-order model (5.14) and (5.15) is indeed a matrix-Padé model of the original system.

Theorem 5.2. (Matrix-Padé model) Suppose that Algorithm 5.1 is run with exact deflation only and that $n \geq \max\{m, p\}$. Then, the reduced-order model (5.14) and (5.15) is a matrix-Padé model of the linear dynamical system (2.1) and (2.2). More precisely, the Taylor expansions about s_0 of the transfer functions H (2.8) and H_n (5.16) agree to as many leading coefficients as possible, that is,

$$H(s) = H_n(s) + \mathcal{O}\bigl((s - s_0)^{q(n)}\bigr),$$

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

where $q(n)$ is as large as possible. In particular,

$$q(n) \geq \left\lfloor \frac{n}{m} \right\rfloor + \left\lfloor \frac{n}{p} \right\rfloor.$$

A proof of Theorem 5.2 is given in Freund (1995). Earlier related results, which required additional assumptions, can be found in de Villemagne and Skelton (1987) and Feldmann and Freund (1995*b*).

5.3. A connection with shifted Krylov-subspace solvers

The representation (3.4) of the transfer function H suggests to employ the machinery of shifted Krylov-subspace methods (Freund 1993) for reduced-order modelling. Indeed, let us define the new variable

$$\sigma(s) := \frac{1}{s - s_0}, \quad s \in \mathbb{C}, \quad s \neq s_0. \tag{5.17}$$

Using (5.17), we can rewrite (3.4) as follows:

$$H(s) = D + \sigma(s)\, L^T (M - \sigma(s)\, I)^{-1} R, \quad s \neq s_0. \tag{5.18}$$

For any $\sigma \in \mathbb{C}$ that is not an eigenvalue of the matrix M, let $X(\sigma)$ denote the unique solution of the block linear system

$$\bigl(M - \sigma I\bigr) X(\sigma) = R. \tag{5.19}$$

By (5.18) and (5.19), we have

$$H(s) = D + \sigma(s)\, L^T X\left(\sigma(s)\right) \tag{5.20}$$

for any $s \in \mathbb{C}$ such that $s \neq s_0$ and $\sigma(s)$ is not an eigenvalue of M.

In view of (5.20), we can compute the values $H(s)$ via solution of block linear systems of the form (5.19). Furthermore, (5.19) is a family of shifted systems, that is, the coefficient matrices of (5.19) differ from the fixed matrix M only by scalar multiples of the identity matrix. It is well known that Krylov-subspace methods for the solution of linear equations can exploit this shift structure; see, *e.g.*, Freund (1993) and the references given there. The basic observation is that Krylov subspaces are invariant under additive shifts by scalar multiples of the identity matrix. The underlying Krylov-subspace method thus has to be run only once, and approximate solutions of any shifted system can then be obtained by solving small shifted problems.

Next, we describe one such method, namely a variant of the block *biconjugate gradient* (BCG) method (O'Leary 1980), in a little more detail. Our variant of block BCG is based on the band Lanczos method. Recall that, after n steps, Algorithm 5.1 (applied to the matrices M, R, and L) has generated the matrices $T_n^{(\mathrm{pr})}$, $\rho_n^{(\mathrm{pr})}$, and $\eta_n^{(\mathrm{pr})}$ and that these satisfy (5.12). In terms of these matrices, the nth block BCG iterate, $X_n(\sigma)$, for the block

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

system (5.19) can be expressed as follows:

$$X_n(\sigma) = V_n Z_n(\sigma), \tag{5.21}$$

where $Z_n(\sigma)$ is the solution of the shifted block Lanczos system

$$\bigl(T_n^{(\mathrm{pr})} - \sigma I_n\bigr) Z_n(\sigma) = \rho_n^{(\mathrm{pr})}. \tag{5.22}$$

Recall that V_n is the matrix of right Lanczos vectors defined in (5.11). Also, note that the coefficient matrix of the system (5.22) is of size $n \times n$. By choosing $\sigma = \sigma(s)$ and inserting the associated block BCG iterate into (5.20), we obtain the approximation

$$H^{(n)}(s) := D + \sigma(s)\, L^T X_n\left(\sigma(s)\right) \approx H(s) \tag{5.23}$$

for the value of the transfer function H at s. Using (5.21), (5.22), (5.12), and (5.17), it readily follows from (5.23) that

$$\begin{aligned} H^{(n)}(s) &:= D + \sigma(s)\left(\eta_n^{(\mathrm{pr})}\right)^T \Delta_n \bigl(T_n^{(\mathrm{pr})} - \sigma(s)\bigr)^{-1} \rho_n^{(\mathrm{pr})} \\ &= D - \left(\eta_n^{(\mathrm{pr})}\right)^T \Delta_n \bigl(I - (s - s_0) T_n^{(\mathrm{pr})}\bigr)^{-1} \rho_n^{(\mathrm{pr})}. \end{aligned} \tag{5.24}$$

By comparing (5.24) and (5.16), we conclude that

$$H_n(s) = H^{(n)}(s). \tag{5.25}$$

This means that computing approximate values of $H(s)$ via n iterations of the shifted block BCG method is equivalent to matrix-Padé approximation.

Of course, this equivalence no longer holds true when shifted variants of other Krylov-subspace solvers, such as block QMR (Freund and Malhotra 1997), are employed.

5.4. *The SyMPVL algorithm*

A disadvantage of Padé models is that, in general, they do not preserve the stability and possibly passivity of the original linear dynamical system. In part, these problems can be overcome by means of suitable post-processing techniques, such as the ones described in Bai, Feldmann and Freund (1998), and Bai and Freund (2001*a*). However, the reduced-order models obtained by post-processing of Padé models are necessarily no longer optimal in the sense of Padé approximation. Furthermore, post-processing techniques are not guaranteed to result always in stable and possibly passive reduced-order models.

For special cases, however, Padé models can be shown to be stable and passive. In particular, this is the case for linear dynamical systems describing RC subcircuits, RL subcircuits, and LC subcircuits; see Bai and Freund (2001*b*) and Freund and Feldmann (1996*a*, 1997, 1998).

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

Next, we describe the SyMPVL algorithm (Freund and Feldmann 1996a, 1997, 1998), which is a special version of MPVL tailored to linear RCL subcircuits.

Recall from Section 4.3 that linear RCL subcircuits can be described by linear dynamical systems (2.1) and (2.2) with $D = 0$, symmetric matrices A and E of the form (4.6), and matrices $B = C$ of the form (4.7). Furthermore, the transfer function (4.8), H, is symmetric.

We now assume that the expansion point s_0 for the Padé approximation is chosen to be real and nonnegative, that is, $s_0 \geq 0$. Together with (4.6) it follows that the matrix $A - s_0 E$ is symmetric indefinite, with N_1 nonpositive and N_2 nonnegative eigenvalues. Thus, $A - s_0 E$ admits a factorization of the following form:

$$A - s_0 E = -F_1 J F_1^T, \tag{5.26}$$

where J is the block matrix defined in (4.9). Instead of the general factorization (3.3), we now use (5.26). By (5.26) and (3.5), the matrices M, R, and L are then of the following form:

$$M = F_1^{-1} E F_1^{-T} J, \quad R = F_1^{-1} B, \quad \text{and} \quad L = -J F_1^{-1} C.$$

Since $E = E^T$ and $B = C$, it follows that

$$JM = M^T J \quad \text{and} \quad L = -JR.$$

This means that M is J-symmetric and the left starting block L is (up to its sign) the J-multiple of the right starting block R. These two properties imply that all the right and left Lanczos vectors generated by the band Lanczos Algorithm 5.1 are J-multiples of each other:

$$w_j = J v_j \quad \text{for all} \quad j = 1, 2, \ldots, n.$$

Consequently, Algorithm 5.1 simplifies, in that only the right Lanczos vectors need to be computed. The resulting version of MPVL for computing matrix-Padé models of RCL subcircuits is just the SyMPVL algorithm. The computational costs of SyMPVL are half of that of the general MPVL algorithm.

Let $H_n^{(1)}$ denote the matrix-Padé model generated by SyMPVL after n Lanczos steps. For general RCL subcircuits, however, $H_n^{(1)}$ will not preserve the passivity of the original system.

An additional reduced-order model that is guaranteed to be passive can be obtained as follows, provided that all right Lanczos vectors are stored. Let

$$V_n = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$$

denote the matrix that contains the first n right Lanczos vectors as columns. Then, by projecting the matrices in the representation (4.10) of the transfer

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

function H of the original RCL subcircuit onto the columns of V_n, we obtain the following reduced-order transfer function:

$$H_n^{(2)}(s) = \left(V_n^T B\right)^T \left(sV_n^T \tilde{E} V_n - V_n^T \tilde{A} V_n\right)^{-1} V_n^T B. \tag{5.27}$$

The passivity of the original RCL subcircuit, together with Theorem 4.5, implies that the reduced-order model defined by $H_n^{(2)}$ is indeed passive. Furthermore, in Freund (2000*b*), it is shown that $H_n^{(2)}$ is a matrix-Padé-type approximation of the original transfer function and that, at the expansion point s_0, $H_n^{(2)}$ matches half as many leading coefficients of H as the matrix-Padé approximant $H_n^{(1)}$.

Next, we illustrate the behaviour of SyMPVL with two circuit examples.

5.5. A package model

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

In Freund and Feldmann (1997), SyMPVL was used to compute a Padé-based reduced-order model of the package, and it was found that a model $H_n^{(1)}$ of order $n = 80$ is sufficient to match the transfer-function components of interest. However, the model $H_n^{(1)}$ 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 $H_n^{(2)}$ given by (5.27). The expansion point $s_0 = 5\pi \times 10^9$ was used. Recall that $H_n^{(2)}$ is only a Padé-type approximant and thus less accurate than the Padé approximant $H_n^{(2)}$. Therefore, we now have to go to order $n = 112$ to obtain a projected reduced-order model $H_n^{(2)}$ that matches the transfer-function components of interest. Figures 5.1 and 5.2 show the voltage-to-voltage transfer function between the external terminal of pin no. 1 and the internal terminals of the same pin and the neighbouring pin no. 2, respectively. The plots show results with the projected model $H_n^{(2)}$ and the Padé model $H_n^{(2)}$, both of order $n = 112$, compared with an exact analysis.

In Figure 5.3 we compare the relative error of the projected model $H_{112}^{(2)}$ and the Padé model $H_{112}^{(1)}$ of the same size. Clearly, the Padé model is more

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

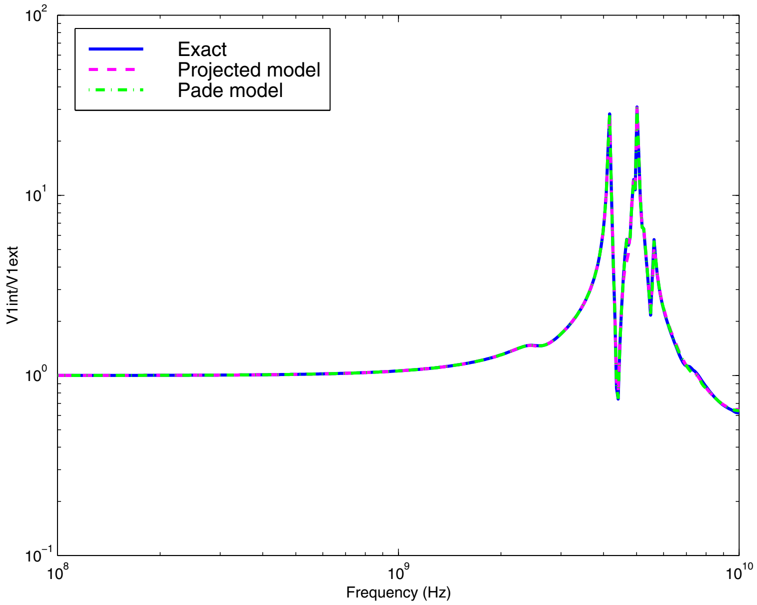

Figure 5.1. Package: pin no. 1 external to pin no. 1 internal, exact, projected model, and Padé model.

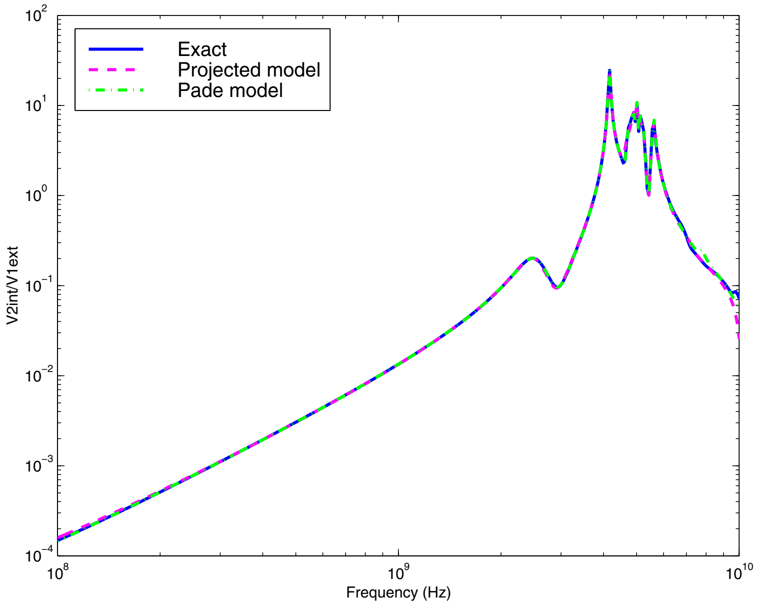

Figure 5.2. Package: pin no 1 external to pin no. 2 internal, exact, projected model, and Padé model.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

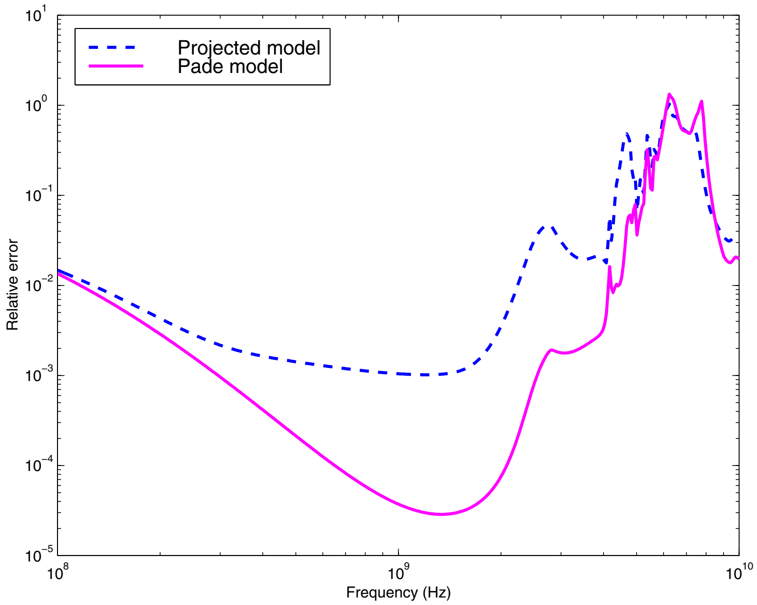

Figure 5.3. Relative error of projected model and Padé model.

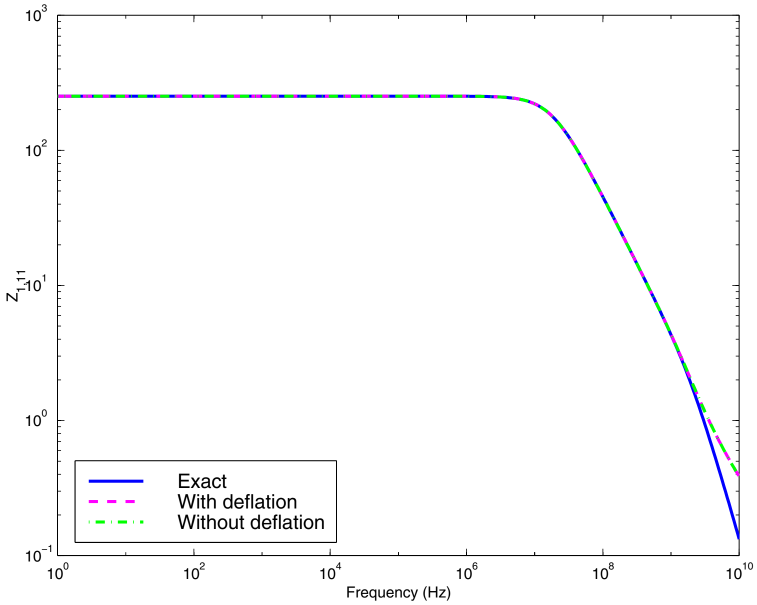

Figure 5.4. Impedance $H_{1,11}$.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

accurate. However, out of the 112 poles of $H_{112}^{(1)}$, 22 have positive real parts, violating the passivity of the Padé model. On the other hand, the projected model is passive.

5.6. 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. Sweeps through the first two Krylov blocks, R and MR, of the block Krylov matrix (5.5) 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 Figure 5.4, we show the $H_{1,11}$ 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.

6. Approaches based on the Arnoldi process

The Arnoldi process (Arnoldi 1951) is another widely used Krylov-subspace method. A band version of the Arnoldi process that is suitable for multiple starting vectors can also be used for reduced-order modelling. However, the models generated from the band Arnoldi process are only Padé-type models.

In contrast to the band Lanczos algorithm, the band Arnoldi process only involves one of the starting blocks, namely R, and it only uses matrix-vector products with M. Moreover, the band Arnoldi process only generates one set of vectors, $v_1, v_2, \ldots, v_n$, instead of the two sequences of right and left vectors produced by the band Lanczos algorithm. The Arnoldi vectors span the nth right block Krylov subspace (induced by M and R):

$$\operatorname{span}\{\, v_1, v_2, \ldots, v_n \,\} = \mathcal{K}_n(M, R).$$

The Arnoldi vectors are constructed to be orthonormal:

$$V_n^H V_n = I, \quad \text{where} \quad V_n := \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}.$$

After n iterations, the Arnoldi process has generated the first n Arnoldi vectors, namely the n columns of the matrix V_n, as well as an $n \times n$ matrix

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

$G_n^{(\mathrm{pr})}$ of recurrence coefficients, and, provided that $n \geq m$, an $n \times m$ matrix $\rho_n^{(\mathrm{pr})}$. The matrices $G_n^{(\mathrm{pr})}$ and $\rho_n^{(\mathrm{pr})}$ are projections of the matrices M and R onto the subspace spanned by the columns of V_n, which is just the block Krylov subspace $\mathcal{K}_n(M, R)$. More precisely, we have

$$G_n^{(\mathrm{pr})} = V_n^H M V_n \quad \text{and} \quad \rho_n^{(\mathrm{pr})} = V_n^H R. \tag{6.1}$$

The band Arnoldi process can be stated as follows.

Algorithm 6.1. (Band Arnoldi process)

```
INPUT: A matrix $M \in \mathbb{C}^{n \times n}$;
       A block of $m$ right starting vectors $R = \begin{bmatrix} r_1 & r_2 & \cdots & r_m \end{bmatrix} \in \mathbb{C}^{n \times m}$.
OUTPUT: The $n \times n$ Arnoldi matrix $G_n^{(\mathrm{pr})}$.
        The matrix $V_n = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$ containing the first $n$
        Arnoldi vectors, and the matrix $\rho_n^{(\mathrm{pr})}$.

(0) For $k = 1, 2, \ldots, m$, set $\hat{v}_k = r_k$.
    Set $m_{\mathrm{c}} = m$ and $\mathcal{I} = \emptyset$.

For $n = 1, 2, \ldots,$ until convergence or $m_{\mathrm{c}} = 0$ do:

  (1) (If necessary, deflate $\hat{v}_n$.)
      Compute $\|\hat{v}_n\|_2$.
      Decide if $\hat{v}_n$ should be deflated. If yes, do the following:
      (a) Set $\hat{v}_{n-m_{\mathrm{c}}}^{\mathrm{defl}} = \hat{v}_n$ and store this vector. Set $\mathcal{I} = \mathcal{I} \cup \{\, n - m_{\mathrm{c}} \,\}$.
      (b) Set $m_{\mathrm{c}} = m_{\mathrm{c}} - 1$. If $m_{\mathrm{c}} = 0$, set $n = n - 1$ and stop.
      (c) For $k = n, n+1, \ldots, n + m_{\mathrm{c}} - 1$, set $\hat{v}_k = \hat{v}_{k+1}$.
      (d) Repeat all of step (1).
  (2) (Normalize $\hat{v}_n$ to obtain $v_n$.)
      Set
          $g_{n,n-m_{\mathrm{c}}} = \|\hat{v}_n\|_2 \quad \text{and} \quad v_n = \dfrac{\hat{v}_n}{g_{n,n-m_{\mathrm{c}}}}.$
  (3) (Orthogonalize the candidate vectors against $v_n$.)
      For $k = n+1, n+2, \ldots, n + m_{\mathrm{c}} - 1$, set
          $g_{n,k-m_{\mathrm{c}}} = v_n^H \hat{v}_k \quad \text{and} \quad \hat{v}_k = \hat{v}_k - v_n \, g_{n,k-m_{\mathrm{c}}}.$
  (4) (Advance the block Krylov subspace to get $\hat{v}_{n+m_{\mathrm{c}}}$.)
      (a) Set $\hat{v}_{n+m_{\mathrm{c}}} = M \, v_n$.
      (b) For $k = 1, 2, \ldots, n$, set
          $g_{k,n} = v_k^H \hat{v}_{n+m_{\mathrm{c}}} \quad \text{and} \quad \hat{v}_{n+m_{\mathrm{c}}} = \hat{v}_{n+m_{\mathrm{c}}} - v_k \, g_{k,n}.$
```

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

(5) (a) For $k \in \mathcal{I}$, set $g_{n,k} = v_n^H \hat{v}_k^{\text{defl}}$.

(b) Set

$$G_n^{(\text{pr})} = \bigl[g_{i,k}\bigr]_{i,k=1,2,\dots,n}\,,$$
$$k_\rho = m + \min\{\,0,\; n - m_{\text{c}}\,\},$$
$$\rho_n^{(\text{pr})} = \bigl[g_{i,k-m}\bigr]_{i=1,2,\dots,n;\,k=1,2,\dots,k_\rho}.$$

(6) Check if n is large enough. If yes, stop.

Note that, in contrast to the band Lanczos algorithm, the band Arnoldi process requires the storage of all previously computed Arnoldi vectors.

Like the band Lanczos algorithm, the band Arnoldi process can also be employed for reduced-order modelling. Let M, R, and L be the matrices defined in (3.5). After running Algorithm 6.1 (applied to M and R) for n steps, we have obtained the matrices $G_n^{(\text{pr})}$ and $\rho_n^{(\text{pr})}$, as well as the matrix V_n of Arnoldi vectors. The transfer function H_n of a reduced-order model H_n can now be defined as follows:

$$H_n(s) = \bigl(V_n^H L\bigr)^H \bigl(I - (s - s_0)V_n^H M V_n\bigr)^{-1}\bigl(V_n^H R\bigr).$$

Using the relations (6.1) for $G_n^{(\text{pr})}$ and $\rho_n^{(\text{pr})}$, the formula for H_n reduces to

$$H_n(s) = \bigl(V_n^H L\bigr)^H \bigl(I - (s - s_0)G_n^{(\text{pr})}\bigr)^{-1}\rho_n^{(\text{pr})}. \tag{6.2}$$

The matrices $G_n^{(\text{pr})}$ and $\rho_n^{(\text{pr})}$ are directly available from Algorithm 6.1. In addition, we also need to compute the matrix

$$\eta_n^{(\text{pr})} = V_n^H L.$$

It turns out that the transfer function (6.2) defines a matrix-Padé-type reduced-order model.

Theorem 6.2. (Matrix-Padé-type model) Suppose that Algorithm 6.1 is run with exact deflation only and that $n \geq m$. Then, the reduced-order model associated with the reduced-order transfer function (6.2) is a matrix-Padé-type model of the linear dynamical system (2.1) and (2.2). More precisely, the Taylor expansions about s_0 of the transfer functions (2.8), H, and (6.2), H_n, agree in at least

$$q'(n) \geq \left\lfloor \frac{n}{m} \right\rfloor$$

leading coefficients:

$$H(s) = H_n(s) + \mathcal{O}\bigl((s - s_0)^{q'(n)}\bigr). \tag{6.3}$$

A proof of Theorem 6.2 is given in Freund (2000*b*).

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

Remark 4. The number $q'(n)$ is the exact number of terms matched in the expansion (6.3) provided that no exact deflations occur in Algorithm 6.1. In the case of exact deflations, the number of matching terms is somewhat higher, but so is the number of matching terms for the matrix-Padé model of Theorem 5.2; see Freund (2000*b*). In particular, the matrix-Padé model is always more accurate than the matrix-Padé-type model obtained from Algorithm 6.1. On the other hand, the band Arnoldi process is certainly simpler than the band Lanczos process. Furthermore, the true orthogonality of the Arnoldi vectors generally results in better numerical behaviour than the bi-orthogonality of the Lanczos vectors.

Remark 5. For the special case of RCL subcircuits, the algorithm PRIMA proposed by Odabasioglu (1996) and Odabasioglu, Celik and Pileggi (1997) can be interpreted as a special case of the Arnoldi reduced-order modelling procedure described here. Furthermore, in Freund (1999*a*) and (2000*b*) it is shown that the reduced-order model produced by PRIMA is mathematically equivalent to the additional passive model produced by SyMPVL. In contrast to PRIMA, however, SyMPVL also produces a true matrix-Padé model, and thus PRIMA does not appear to have any real advantage over – or even be competitive with – SyMPVL.

Remark 6. It is also possible to devise a two-sided Arnoldi procedure and then generate Padé models from it. Such an approach is described in Cullum and Zhang (2002).

7. Circuit-noise computations

In this section, we discuss the use of reduced-order modelling for *circuit-noise computations*. In particular, we show how noise-type transfer functions can be rewritten so that reduced-order modelling techniques for linear dynamical systems can be applied. The material in this section is based on the paper by Feldmann and Freund (1997).

7.1. The problem

Noise in electronic circuits is caused by the stochastical fluctuations in currents and voltages that occur within the devices of the circuit. We refer the reader to Chapter 8 of Davidse (1991) or to van der Ziel (1986) for an introduction to circuit noise and the main noise mechanisms. Noise-analysis algorithms for circuits in DC steady-state have been available for a long time in traditional circuit simulators such as SPICE (Rohrer, Nagel, Meyer and Weber 1971). As we will now describe, simulation techniques based on reduced-order modelling, such as PVL and MPVL, can easily be extended to include noise computations.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

Noise in circuit devices is modelled by stochastic processes. In the time domain, a stochastic process is characterized in terms of statistical averages, such as the mean and autocorrelation, and in the frequency domain, it is described by the spectral power density. The main types of noise in integrated circuits are *thermal noise*, *shot noise*, and *flicker noise*. Thermal and shot noise represent *white noise*, that is, their *spectral power densities* do not depend on the frequency ω. Flicker noise is modelled by a stochastic process with a spectral power density that is proportional to $(1/\omega)^\beta$ where β is a constant of about one.

Next, we describe the problem of noise computation for circuits with constant excitation in steady-state (DC). Moreover, we assume that all time-varying circuit elements are independent sources. In this case, the general system of circuit equations (1.1) simplifies to a system of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\, q(\hat{x}) + f(\hat{x}) = b_0. \tag{7.1}$$

Here, b_0 denotes the constant excitation vector. Let $\hat{x}_0$ be a DC operating point of the circuit, that is, $\hat{x}_0$ is a constant vector that satisfies $f(\hat{x}_0) = b_0$. Adding noise sources to (7.1) gives

$$\frac{\mathrm{d}}{\mathrm{d}t} q(\hat{x} + x) + f(\hat{x} + x) = b_0 + B\,\nu(t), \tag{7.2}$$

where $\nu(t)$ is a vector stochastic process of length m that describes the noise sources, $B \in \mathbb{R}^{N \times m}$ is the noise-source incidence matrix, and m denotes the number of noise sources. The vector function $x = x(t)$ in (7.2) represents the stochastical deviations of the circuit variables from the DC operating point $\hat{x}_0$ that are caused by the noise sources. By linearizing (7.2) about $\hat{x}_0$ and using the fact that $f(\hat{x}_0) = b_0$, we obtain the following linear system of DAEs:

$$E \frac{\mathrm{d}x}{\mathrm{d}t} = Ax + B\nu(t), \tag{7.3}$$

$$y(t) = C^T x(t). \tag{7.4}$$

Here,

$$A = -\mathrm{D}_x f(\hat{x}_0) \quad \text{and} \quad E = \mathrm{D}_x q(\hat{x}_0) \tag{7.5}$$

that is, A is the negative of the Jacobian matrix of f at the point $\hat{x}_0$ and E is the Jacobian matrix of q at the point $\hat{x}_0$. Furthermore, in (7.4), $y(t)$ is a vector stochastic process of length p describing the stochastical deviations at the outputs of interest due to the noise sources, and $C \in \mathbb{R}^{N \times p}$ is a constant matrix that selects the outputs of interest. Note that (7.3) and (7.4) describe a linear dynamical system of the form (1.2) and (1.3) with m inputs and p outputs. Thus we can use MPVL or, if $m = p = 1$, PVL to generate reduced-order models for (7.3) and (7.4).

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

For noise computations in the frequency domain, the goal is to compute the $(p \times p)$-cross-spectral power density matrix $S_y(\omega)$ of the vector stochastic process y in (7.4). It turns out that

$$S_y(\omega) = C^T \, (\mathrm{i}\,\omega E - A)^{-1} B \, S_\nu(\omega) \, B^T \, (\mathrm{i}\,\omega E - A)^{-H} C \tag{7.6}$$

for all $\omega \geq 0$. Here, ω denotes frequency, and $S_\nu(\omega)$ is the given $m \times m$ cross-spectral power density matrix of the noise sources $\nu(t)$ in (7.2). We remark that the diagonal entries of $S_\nu(\omega)$ are the spectral power densities of the noise sources, and that nonzero off-diagonal entries of $S_\nu(\omega)$ occur only if there is coupling between some of the noise sources. Moreover, if all noise sources are white, then S_ν is a constant matrix.

7.2. Reformulation as a transfer function

Clearly, the matrix-valued function (7.6), S_y, does not have the form of a transfer function (2.8). Consequently, the reduced-order modelling techniques we discussed so far cannot be applied directly to S_y. However, for the physical relevant values $\omega \geq 0$ and under some mild assumptions on the form of S_ν, we can rewrite (7.6) as a function of the type (2.8). More precisely, we assume that

$$S_\nu(\omega) = (P(\mathrm{i}\,\omega))^{-1} \quad \text{for all} \quad \omega \geq 0, \tag{7.7}$$

where

$$P(s) = P_0 + P_1 s + \cdots + P_M s^M, \quad P_i \in \mathbb{C}^{m \times m}, \quad 0 \leq i \leq M, \tag{7.8}$$

is any matrix polynomial of degree M (that is, $P_M \neq 0$). In particular, for the important special case that all noise sources are white, as in the case of thermal and shot noise, we have

$$P(s) = P_0 = S_\nu^{-1} \quad \text{and} \quad M = 0. \tag{7.9}$$

If $S_\nu(\omega)$ does depend on the frequency, as in the case of flicker noise, then the assumption (7.7) is satisfied at least approximately, see Feldmann and Freund (1997).

By inserting (7.7) into (7.6) and setting

$$H(s) := C^T \, (sE - A)^{-1} B \, (P(s))^{-1} B^T \, (sE - A)^{-H} C, \quad s \in \mathbb{C}, \tag{7.10}$$

it follows that

$$H(\mathrm{i}\,\omega) = S_y(\omega) \quad \text{for all} \quad \omega \geq 0. \tag{7.11}$$

The relation (7.11) suggests first generating an approximation H_n to the function H in (7.10) and then using

$$S_y(\omega) \approx H_n(\mathrm{i}\,\omega) \tag{7.12}$$

as an approximation to S_y. It turns out that the function H can be rewritten

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

as a transfer function of the type (2.8), and thus we can employ MPVL (or PVL if $p = 1$) to obtain H_n as an nth matrix-Padé approximant to H. More precisely, in Feldmann and Freund (1997), it is shown that

$$H(s) = \widetilde{C}^T \left(s\,\widetilde{E} - \widetilde{A} \right)^{-1} \widetilde{C} \quad \text{for all} \quad s \in \mathbb{C}. \tag{7.13}$$

Here, $\widetilde{C} \in \mathbb{C}^{\widetilde{N} \times p}$ and $\widetilde{A},\ \widetilde{E} \in \mathbb{C}^{\widetilde{N} \times \widetilde{N}}$ are matrices given by

$$\begin{gathered}
\widetilde{C} := \begin{bmatrix} C \\ 0_{N\times p} \\ 0_{m\times p} \\ 0_{m\times p} \\ \vdots \\ 0_{m\times p} \end{bmatrix}, \quad
\widetilde{A} := \begin{bmatrix} 0 & A^T & 0 & 0 & \cdots & 0 \\ A & 0 & -B & 0 & \cdots & 0 \\ 0 & -B^T & -P_0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & I & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 0 & I \end{bmatrix}, \\
\widetilde{E} := \begin{bmatrix} 0 & -E^T & 0 & 0 & \cdots & 0 \\ E & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & P_1 & P_2 & \cdots & P_M \\ 0 & 0 & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & I & 0 \end{bmatrix},
\end{gathered} \tag{7.14}$$

and $\widetilde{N} := 2 \cdot N + m \cdot M$.

If the matrix polynomial P is linear, that is, $M = 1$ in (7.8), the matrices (7.14) reduce to

$$\widetilde{C} := \begin{bmatrix} C \\ 0 \\ 0 \end{bmatrix}, \quad \widetilde{A} := \begin{bmatrix} 0 & A^T & 0 \\ A & 0 & -B \\ 0 & -B^T & -P_0 \end{bmatrix}, \quad \widetilde{E} := \begin{bmatrix} 0 & -E^T & 0 \\ E & 0 & 0 \\ 0 & 0 & P_1 \end{bmatrix}. \tag{7.15}$$

The important special case (7.9) of white noise is also covered by (7.15) with $P_0 := S_\nu^{-1}$ and $P_1 := 0$. In this case, by eliminating the third block rows and columns in (7.15), the matrices $\widetilde{C}$, $\widetilde{A}$, and $\widetilde{E}$ can be further reduced to

$$\widetilde{C} = \begin{bmatrix} C \\ 0 \end{bmatrix}, \quad \widetilde{A} = \begin{bmatrix} 0 & A^T \\ A & B^T S_\nu B \end{bmatrix}, \quad \widetilde{E} = \begin{bmatrix} 0 & -E^T \\ E & 0 \end{bmatrix}. \tag{7.16}$$

7.3. A PVL simulation

We now present results of a typical simulation with the noise-computation algorithm described in Section 7.2.

The example is a 5th-order Cauer filter that uses ten 741 operational amplifiers as building blocks. The total size of the problem is 463 variables. The noise sources are all white. The circuit has a single input and a single output, and we employ PVL to compute an nth Padé approximant to the

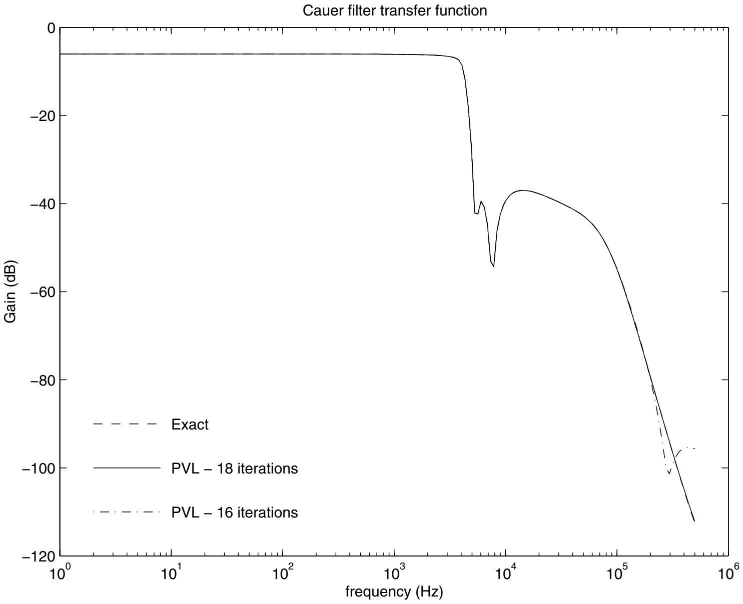

Figure 7.1. Transfer characteristic of the Cauer filter.

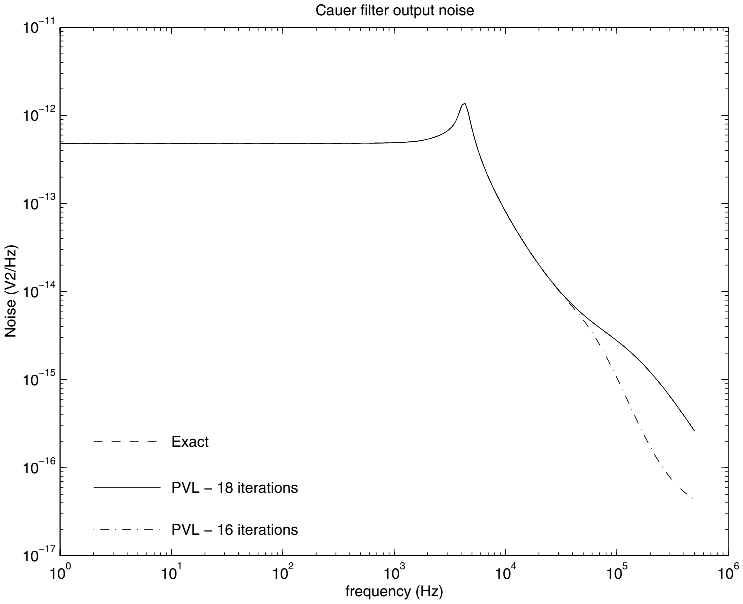

Figure 7.2. Spectral power density of the Cauer filter.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

transfer function of the circuit. In addition, we also compute the spectral power density of the noise at the single output, by applying PVL to the rewritten noise-type transfer function (7.13) with matrices C, $\widetilde{A}$, and $\widetilde{E}$ given by (7.16). Figure 7.1 shows the exact transfer function of the 5th order Cauer filter compared to the PVL-computed Padé approximants of order $n = 16$ and $n = 18$. The 18th order approximation captures the behaviour of the circuit almost exactly.

Figure 7.2 shows the spectral power density of the output noise over the same frequency range and for the same approximation order $n = 16$ and $n = 18$. Note that the same number n of PVL iterations is needed to obtain an almost perfect match of both the transfer function and the noise spectrum.

8. Second-order linear dynamical systems

In this section, we describe some reduced-order modelling approaches for second-order linear dynamical systems. Most of the material in this section is taken from the unpublished report by Bai, Dewilde and Freund (2002).

8.1. The problem

Second-order models arise naturally in the study of many types of physical systems, such as electrical and mechanical systems. A *time-invariant multi-input multi-output second-order system* is described by equations of the form

$$M\frac{\mathrm{d}^2q}{\mathrm{d}t^2} + D\frac{\mathrm{d}q}{\mathrm{d}t} + Kq = Pu(t), \tag{8.1}$$

$$y(t) = L^T q(t), \tag{8.2}$$

together with initial conditions $q(0) = q_0$ and $\frac{\mathrm{d}}{\mathrm{d}t}q(0) = \dot{q}_0$. Here, $q(t) \in \mathbb{R}^N$ is the vector of state variables, $u(t) \in \mathbb{R}^m$ is the input vector, and $y(t) \in \mathbb{R}^p$ is the output vector. Moreover, M, D, $K \in \mathbb{R}^{N\times N}$ are system matrices, such as mass, damping, and stiffness matrices in structural dynamics, $P \in \mathbb{R}^{N\times m}$ is the input distribution matrix, and $L \in \mathbb{R}^{N\times p}$ is the output distribution matrix. Finally, N is the state-space dimension, and m and p are the number of inputs and outputs, respectively. In most practical cases, m and p are much smaller than N.

The second-order system (8.1) and (8.2) can be reformulated as an equivalent linear first-order system in many different ways. We will use the following equivalent linear system:

$$E\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + Bu(t), \tag{8.3}$$

$$y(t) = C^T x(t), \tag{8.4}$$

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

where

$$x = \begin{bmatrix} q \\ \frac{\mathrm{d}q}{\mathrm{d}t} \end{bmatrix}, \ A = \begin{bmatrix} -K & 0 \\ 0 & W \end{bmatrix}, \ E = \begin{bmatrix} D & M \\ W & 0 \end{bmatrix}, \ B = \begin{bmatrix} P \\ 0 \end{bmatrix}, \ C = \begin{bmatrix} L \\ 0 \end{bmatrix}.$$

Here, $W \in \mathbb{R}^{N \times N}$ can be any nonsingular matrix. A common choice is the identity matrix, $W = I$. If the matrices M, D, and K are all symmetric and M is nonsingular, as is often the case in structural dynamics, we can choose $W = M$. The resulting matrices A and E in the linearized system (8.3) are then symmetric, and thus preserve the symmetry of the original second-order system.

Assume that, for simplicity, we have zero initial conditions, that is, $q(0) = q_0$, $\frac{\mathrm{d}}{\mathrm{d}t}\, q(0) = 0$, and $u(0) = 0$ in (8.1) and (8.2). Then, by taking the Laplace transform of (8.1) and (8.2), we obtain the following algebraic system:

$$\begin{aligned} s^2 MQ(s) + DQ(s) + KQ(s) &= PU(s), \\ Y(s) &= L^T Q(s). \end{aligned}$$

Eliminating $Q(s)$ in this system results in the frequency-domain input-output relation $Y(s) = H(s)U(s)$, where

$$H(s) := L^T \left(s^2 M + sD + K\right)^{-1} P$$

is the transfer function. In view of the equivalent linearized system (8.3) and (8.4), the transfer function can also be written as

$$H(s) = C^T (sE - A)^{-1} B.$$

If the matrix K in (8.1) is nonsingular, then $s_0 = 0$ is guaranteed not to be a pole of H. In this case, H can be expanded about $s_0 = 0$ as follows:

$$H(s) = \mu_0 + \mu_1 s + \mu_2 s^2 + \cdots ,$$

where the matrices μ_j are the so-called *low-frequency moments*. In terms of the matrices of the linearized system (8.3) and (8.4), the moments are given by

$$\mu_j = -C^T \left(A^{-1} E\right)^j A^{-1} B, \quad j = 0, 1, 2, \ldots .$$

8.2. Frequency-response analysis methods

In this subsection, we describe the use of eigensystem analysis to tackle the second-order system (8.1) and (8.2) directly.

We assume that the input force vector $u(t)$ of (8.1) is time-harmonic:

$$u(t) = \tilde{u}(\omega)\mathrm{e}^{\mathrm{i}\,\omega t},$$

where ω is the frequency of the system. Correspondingly, we assume that the state variables of the second-order system can be represented as follows:

$$q(t) = \tilde{q}(\omega)\mathrm{e}^{\mathrm{i}\,\omega t}.$$

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

The problem of solving the system of second-order differential equations (8.1) then reduces to solving the parametrized linear system of equations

$$(-\omega^2 M + \mathrm{i}\,\omega D + K)\,\tilde{q}(\omega) = P\tilde{u}(\omega) \tag{8.5}$$

for $\tilde{q}(\omega)$. This approach is called the *direct frequency-response analysis method*. For a given frequency ω_0, we can use a linear system solver, either direct or iterative, to obtain the desired vector $\tilde{q}(\omega_0)$.

Alternatively, we can try to reduce the cost of solving the large-scale parametrized linear system of equations (8.5) by first applying an eigensystem analysis. This approach is called the *modal frequency-response analysis* in structural dynamics. The basic idea is to first transfer the coordinates $\tilde{q}(\omega)$ of the state vector $q(t)$ to new coordinates $p(\omega)$ as follows:

$$q(t) \cong W_k p(\omega) \mathrm{e}^{\mathrm{i}\,\omega t}.$$

Here, W_k consists of k selected modal shapes to retain the modes whose resonant frequencies lie within the range of forcing frequencies. More precisely, W_k consists of k selected eigenvectors of the underlying quadratic eigenvalue problem $(\lambda^2 M + \lambda D + K)\,w = 0$. Equation (8.5) is then approximated by

$$(-\omega^2 M W_k + \mathrm{i}\,\omega D W_k + K W_k)\,p(\omega) = P\tilde{u}(\omega).$$

Multiplying this equation from the left by W_k^T, we obtain a $k \times k$ parametrized linear system of equations for $p(\omega)$:

$$(-\omega^2\,(W_k^T M W_k) + \mathrm{i}\,\omega\,(W_k^T D W_k) + (W_k^T K W_k))\,p(\omega) = W_k^T P(\omega).$$

Typically, $k \ll n$. The main question now is how to obtain the desired modal shapes W_k. One possibility is to simply extract W_k from the matrix pair (M, K) by ignoring the contribution of the damping term. This is called the *modal superposition method* in structural dynamics. This approach is applicable under the assumption that the damping term is of a certain form. For example, this is the case for so-called Rayleigh damping $D = \alpha M + \beta K$, where α and β are scalars (Clough and Penzien 1975). In general, however, we may need to solve the full quadratic eigenvalue problem $(\lambda^2 M + \lambda D + K)\,w = 0$ in order to obtain the desired modal shapes W_k. Some of these techniques have been reviewed in the recent survey paper (Tisseur and Meerbergen 2001) on the quadratic eigenvalue problem.

8.3. *Reduced-order modelling based on linearization*

An obvious approach to constructing reduced-order models of the second-order system (8.1) and (8.2) is to apply any of the model-reduction techniques for linear systems to the linearized system (8.3) and (8.4). In particular, we can employ the Krylov-subspace techniques discussed in Sections 5 and 6.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

The resulting approach can be summarized as follows.

1. Linearize the second-order system (8.1) and (8.2) by properly defining the $2N \times 2N$ matrices A and E of the equivalent linear system (8.3) and (8.4). Select an expansion point s_0 'close' to the frequency range of interest and such that the matrix $A - s_0 E$ is nonsingular.

2. Apply a suitable Krylov process, such as the nonsymmetric band Lanczos algorithm described in Section 5, to the matrix $M := (A - s_0 E)^{-1} E$ and the blocks of right and left starting vectors $R := (A - s_0 E)^{-1} B$ and $L := C$ to obtain bi-orthogonal Lanczos basis matrices V_n and W_n for the nth right and left block-Krylov subspaces $\mathcal{K}_n(M, R)$ and $\mathcal{K}_n(M^T, L)$.

3. Approximate the state vector $x(t)$ by $V_n z(t)$, where $z(t)$ is determined by the following linear reduced-order model of the linear system (8.3) and (8.4):
$$\begin{aligned} E_n \frac{\mathrm{d}z}{\mathrm{d}t} &= A_n z + B_n u(t), \\ y(t) &= C_n^T z(t). \end{aligned}$$
Here, $E_n = T_n^{(\mathrm{pr})}$, $A_n = I_n + s_0 T_n^{(\mathrm{pr})}$, $B_n = \rho_n^{(\mathrm{pr})}$, $C_n = \Delta_n^T \eta_n^{(\mathrm{pr})}$, and $T_n^{(\mathrm{pr})}$, $\rho_n^{(\mathrm{pr})}$, $\eta_n^{(\mathrm{pr})}$, and Δ_n are the matrices generated by the nonsymmetric band Lanczos Algorithm 5.1.

In Figure 8.1, we show the results of this approach applied to the linear-drive multi-mode resonator structure described in Clark, Zhou and Pister (1998). The solid lines are the Bode plots of the frequency response of the original second-order system, which is of dimension $N = 63$. The dashed lines are the Bode plots of the frequency response of the reduced-order model of dimension $n = 12$. The relative error between the transfer functions of the original system and the reduced-order model of dimension $n = 12$ is less than 10^{-4} over the frequency range shown in Figure 8.1.

There are a couple of advantages to the linearization approach. First, we can directly employ existing reduced-order modelling techniques developed for linear systems. Second, we can also exploit the structures of the linearized system matrices A and E in a Krylov process to reduce the computational cost. However, the linearization approach also has disadvantages. In particular, it ignores the physical meaning of the original system matrices, and more importantly, the reduced-order models are no longer in a second-order form. For engineering design and control of structural systems, it is often desirable to have reduced-order models that preserve the second-order form: see, *e.g.*, Su and Craig, Jr. (1991).

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

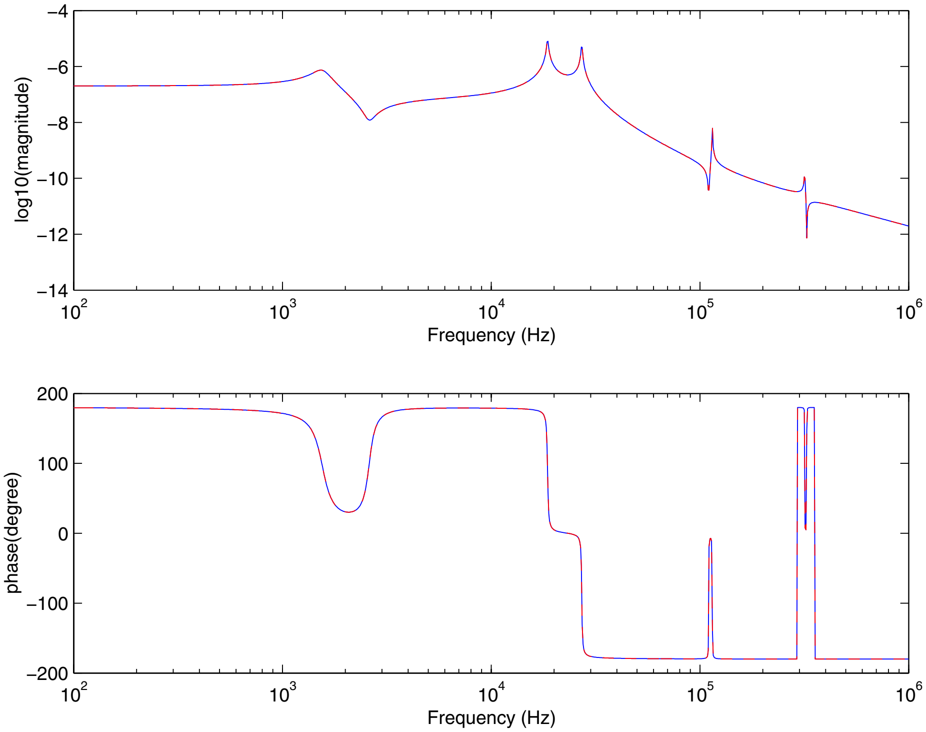

Figure 8.1. Bode plots for the original system and the reduced-order model of dimension $n = 12$.

8.4. Reduced-order modelling based on second-order systems

In this section, we discuss a Krylov-subspace technique that produces a reduced-order model of second-order form. This approach is based on the work by Su and Craig, Jr. (1991).

The key observation is the following. In view of the linearization (8.3) and (8.4) of the second-order system (8.1) and (8.2), the desired Krylov subspace for reduced-order modelling is

$$\operatorname{span}\left\{ \widetilde{B},\, (A^{-1}E)\,\widetilde{B},\, (A^{-1}E)^2\,\widetilde{B}, \dots,\, (A^{-1}E)^{n-1}\,\widetilde{B} \right\}.$$

Here, $\widetilde{B} := -A^{-1}\begin{bmatrix} B & C \end{bmatrix}$. Moreover, we have assumed that the matrix A in (8.3) is nonsingular. Let us set

$$R_j = \begin{bmatrix} R_j^d \\ R_j^v \end{bmatrix} := (-A^{-1}E)^j\,\widetilde{B},$$

where R_j^d is the vector of length N corresponding to the displacement portion of the vector R_j, and R_j^v is the vector of length N corresponding to the velocity portion of the vector R_j: see Su and Craig, Jr. (1991). Then, in view of the structure of the matrices A and E, we have

$$\begin{bmatrix} R_j^d \\ R_j^v \end{bmatrix} = (-A^{-1}E) \begin{bmatrix} R_{j-1}^d \\ R_{j-1}^v \end{bmatrix} = \begin{bmatrix} K^{-1}DR_{j-1}^d + K^{-1}MR_{j-1}^d \\ -R_{j-1}^d \end{bmatrix}.$$

Note that the jth velocity-portion vector R_j^v is the same (up to its sign) as the $(j-1)$st displacement-portion vector R_{j-1}^d. In other words, the second portion R_j^v of R_j is the 'one-step' delay of the first portion R_{j-1}^d of R_j. This suggests that we may simply choose

$$\operatorname{span}\left\{ R_0^d, R_1^d, R_2^d, \ldots, R_{n-1}^d \right\} \tag{8.6}$$

as the projection subspace used for reduced-order modelling.

In practice, for numerical stability, we may opt to employ the Arnoldi process to generate an orthonormal basis Q_n of the subspace (8.6). The resulting procedure can be summarized as follows.

Algorithm 8.1. (Algorithm by Su and Craig Jr.)

```
(0) (Initialization.)
    Set $R_0^d = K^{-1}\begin{bmatrix} P & L \end{bmatrix}$,  $R_0^v = 0$,  $U_0 S_0 V_0^T = (R_0^d)^T K R_0^d$
    (by computing an SVD),
    $Q_1^d = R_0^d U_0 S_0^{-1/2}$, and $Q_1^v = 0$.
(1) (Arnoldi loop.)
    For $j = 1, 2, \ldots, n-1$ do:
        Set $R_j^d = K^{-1}\left(DQ_{j-1}^d + MQ_{j-1}^v\right)$ and $R_j^v = -Q_{j-1}^d$.
(2) (Orthogonalization.)
    For $i = 1, 2, \ldots, j$ do:
        Set $T_i = (Q_i^d)^T K R_j^d$, $R_j^d = R_j^d - Q_i^d T_i$, and $R_j^v = R_j^v - Q_i^v T_i$.
(3) (Normalization.)
    Set $U_0 S_0 V_0^T = (R_j^d)^T K R_j^d$ (by computing an SVD),
    $Q_{j+1}^d = R_j^d U_0 S_0^{-1/2}$, and $Q_{j+1}^v = R_j^v U_0 S_0^{-1/2}$.
```

An approximation of the state vector $q(t)$ can then be obtained by constraining $q(t)$ to the subspace spanned by the columns of Q_n, that is, $q(t) \approx Q_n z(t)$. Moreover, the reduced-order state vector $z(t)$ is defined as the solution of the following second-order system:

$$M_n \frac{\mathrm{d}^2 q}{\mathrm{d}t^2} + D_n \frac{\mathrm{d}q}{\mathrm{d}t} + K_n q = P_n u(t), \tag{8.7}$$

$$y(t) = L_n^T q(t), \tag{8.8}$$

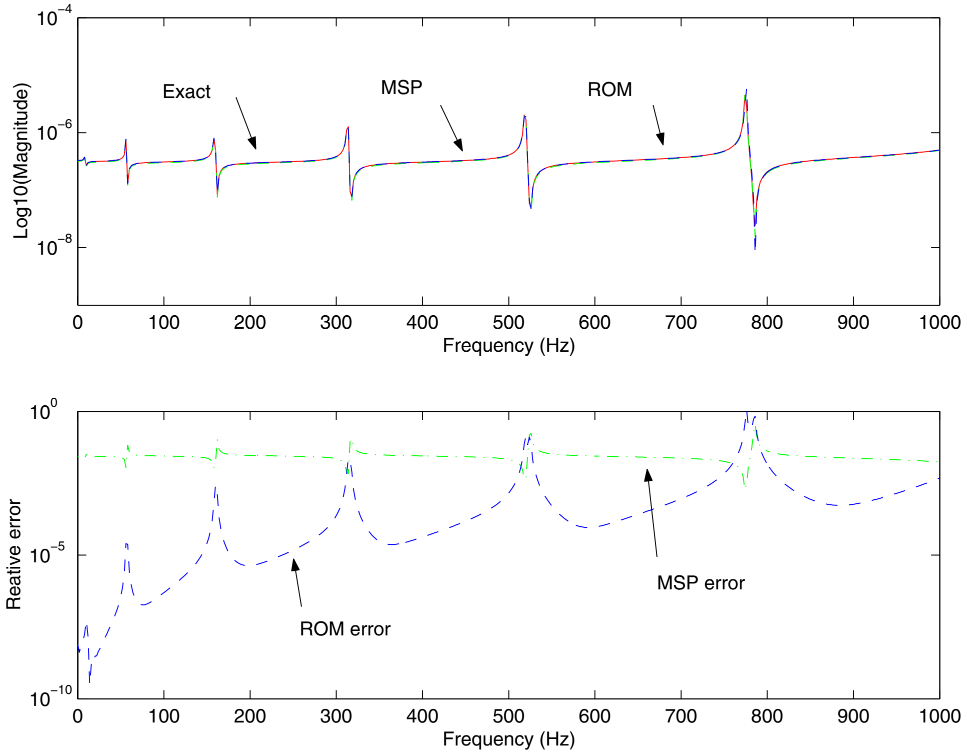

Figure 8.2. Frequency-response analysis (top plot) and relative errors (bottom plot) of a finite-element model of a shaft.

where $M_n := Q_n^T M Q_n$, $D_n := Q_n^T D Q_n$, $K_n := Q_n^T K Q_n$, $P_n := Q_n^T P$, and $L_n := Q_n^T L$. Note that (8.7) and (8.8) describe a reduced-order model in second-order form of the original second-order system (8.1) and (8.2).

Su and Craig, Jr. (1991) describe several advantages of this approach. Here, we present some numerical results of a frequency-response analysis of a second-order system of order $N = 400$, which arises from a finite-element model of a shaft on bearing support with a damper. In the top plot of Figure 8.2, we plot the magnitudes of the transfer function H computed exactly, approximated by the model-superposition (MSP) method, and approximated by the Krylov-subspace technique (ROM). For the MSP method, we used the 80 modal shapes W_{80} from the matrix pencil $\lambda^2 M + K$. The reduced-order model (8.7) and (8.8) is also of dimension $n = 80$. The bottom plot of Figure 8.2 shows the relative errors between the exact transfer function and its approximations based on the MSP method (dash-dotted line) and the ROM method (dashed line). The plots indicate that no accuracy has been lost by the Krylov subspace-based method.

9. Concluding remarks

We have presented a survey of the most common techniques for reduced-order modelling of large-scale linear dynamical systems. By and large, the area of linear reduced-order modelling is fairly well explored, and we have a number of efficient techniques at our disposal. Still, some open problems remain. One such problem is the construction of reduced-order models that preserve stability or passivity and, at the same time, have optimal approximation properties. Particularly in circuit simulation, reduced-order modelling is used to substitute large linear subsystems within the simulation of even larger, generally nonlinear systems. It would be important to better understand the effects of these substitutions on the overall nonlinear simulation.

The systems arising in the simulation of electronic circuits are nonlinear in general, and it would be highly desirable to apply nonlinear reduced-order modelling techniques directly to these nonlinear systems. However, the area of nonlinear reduced-order modelling is in its infancy compared to the state of the art of linear reduced-order modelling. We expect that further progress in model reduction will mainly occur in the area of nonlinear reduced-order modelling.

In this survey, we have focused solely on Krylov subspace-based model-reduction techniques for time-invariant systems. There are of course many other order-reduction approaches that do not fall into this limited category. Methods that we have not treated here include balanced realizations (Moore 1981), Hankel-norm optimal approximations (Glover 1984), order reduction of time-varying systems (Roychowdhury 1999), and proper orthogonal decomposition, which is also known as Karhunen–Loève decomposition (Holmes, Lumley and Berkooz 1996, Glavaški, Marsden and Murray 1998, Rathinam and Petzold 2002).

Acknowledgements

I am grateful to Peter Feldmann, who first introduced me to circuit simulation; many of the results surveyed in this paper are based on joint work with him. I would like to thank Zhaojun Bai for his help with the material on second-order systems and for providing the two numerical examples in Section 8.

REFERENCES

J. I. Aliaga, D. L. Boley, R. W. Freund and V. Hernández (2000), 'A Lanczos-type method for multiple starting vectors', *Math. Comp.* **69**, 1577–1601.

B. D. O. Anderson (1967), 'A system theory criterion for positive real matrices', *SIAM J. Control.* **5**, 171–182.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

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

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

Z. Bai (2002), 'Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems', *Appl. Numer. Math.* **43**, 9–44.

Z. Bai and R. W. Freund (2000), 'Eigenvalue-based characterization and test for positive realness of scalar transfer functions', *IEEE Trans. Automat. Control* **45**, 2396–2402.

Z. Bai and R. W. Freund (2001*a*), 'A partial Padé-via-Lanczos method for reduced-order modeling', *Linear Algebra Appl.* **332–334**, 139–164.

Z. Bai and R. W. Freund (2001*b*), 'A symmetric band Lanczos process based on coupled recurrences and some applications', *SIAM J. Sci. Comput.* **23**, 542–562.

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

Z. Bai, P. M. Dewilde and R. W. Freund (2002), Reduced-order modeling, Numerical Analysis Manuscript No. 02–4–13, Bell Laboratories, Lucent Technologies, Murray Hill, NJ. Also available online from: `http://cm.bell-labs.com/cs/doc/02`.

G. A. Baker, Jr. and P. Graves-Morris (1996), *Padé Approximants*, 2nd edn, Cambridge University Press, New York.

D. J. Bender and A. J. Laub (1987), 'The linear-quadratic optimal regulator for descriptor systems', *IEEE Trans. Automat. Control* **32**, 672–688.

S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan (1994), *Linear Matrix Inequalities in System and Control Theory*, SIAM Publications, Philadelphia, PA.

A. Bultheel and B. De Moor (2000), 'Rational approximation in linear systems and control', *J. Comput. Appl. Math.* **121**, 355–378.

A. Bultheel and M. Van Barel (1986), 'Padé techniques for model reduction in linear system theory: a survey', *J. Comput. Appl. Math.* **14**, 401–438.

S. L. Campbell (1980), *Singular Systems of Differential Equations*, Pitman, London.

S. L. Campbell (1982), *Singular Systems of Differential Equations II*, Pitman, London.

C.-K. Cheng, J. Lillis, S. Lin and N. H. Chang (2000), *Interconnect Analysis and Synthesis*, Wiley, New York.

E. Chiprout and M. S. Nakhla (1994), *Asymptotic Waveform Evaluation*, Kluwer Academic Publishers, Norwell, MA.

P. M. Chirlian (1967), *Integrated and Active Network Analysis and Synthesis*, Prentice-Hall, Englewood Cliffs, NJ.

J. V. Clark, N. Zhou and K. S. J. Pister (1998), MEMS simulation using SUGAR v0.5, in *Proc. Solid-State Sensors and Actuators Workshop*, Hilton Head Island, SC, pp. 191–196.

R. W. Clough and J. Penzien (1975), *Dynamics of Structures*, McGraw-Hill.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

J. Cullum and T. Zhang (2002), 'Two-sided Arnoldi and nonsymmetric Lanczos algorithms', *SIAM J. Matrix Anal. Appl.* **24**, 303–319.

L. Dai (1989), *Singular Control Systems*, Vol. 118 of *Lecture Notes in Control and Information Sciences*, Springer, Berlin, Germany.

J. Davidse (1991), *Analog Electronic Circuit Design*, Prentice-Hall, New York.

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

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

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

P. Feldmann and R. W. Freund (1997), Circuit noise evaluation by Padé approximation based model-reduction techniques, in *Technical Digest of the 1997 IEEE/ACM Int. Conf. on Computer-Aided Design*, IEEE Computer Society Press, Los Alamitos, CA, pp. 132–138.

L. Fortuna, G. Nunnari and A. Gallo (1992), *Model Order Reduction Techniques with Applications in Electrical Engineering*, Springer, London.

R. W. Freund (1993), Solution of shifted linear systems by quasi-minimal residual iterations, in *Numerical Linear Algebra* (L. Reichel, A. Ruttan and R. S. Varga, eds), W. de Gruyter, Berlin, Germany, pp. 101–121.

R. W. Freund (1995), Computation of matrix Padé approximations of transfer functions via a Lanczos-type process, in *Approximation Theory VIII, Vol. 1: Approximation and Interpolation* (C. Chui and L. Schumaker, eds), World Scientific, Singapore, pp. 215–222.

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

R. W. Freund (1999*a*), Passive reduced-order models for interconnect simulation and their computation via Krylov-subspace algorithms, in *Proc. 36th ACM/IEEE Design Automation Conference*, ACM, New York, pp. 195–200.

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

R. W. Freund (2000*a*), Band Lanczos method (Section 7.10), in *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide* (Z. Bai, J. Demmel, J. Dongarra, A. Ruhe and H. van der Vorst, eds), SIAM Publications, Philadelphia, PA, pp. 205–216. Also available online from: `http://cm.bell-labs.com/cs/doc/99`.

R. W. Freund (2000*b*), 'Krylov-subspace methods for reduced-order modeling in circuit simulation', *J. Comput. Appl. Math.* **123**, 395–421.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

R. W. Freund and P. Feldmann (1996*a*), 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, pp. 280–287.

R. W. Freund and P. Feldmann (1996*b*), 'Small-signal circuit analysis and sensitivity computations with the PVL algorithm', *IEEE Trans. Circuits and Systems, II: Analog and Digital Signal Processing* **43**, 577–585.

R. W. Freund and P. Feldmann (1997), The SyMPVL algorithm and its applications to interconnect simulation, in *Proc. 1997 International Conference on Simulation of Semiconductor Processes and Devices*, IEEE, Piscataway, NJ, pp. 113–116.

R. W. Freund and P. Feldmann (1998), 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, pp. 530–537.

R. W. Freund and F. Jarre (2000), An extension of the positive real lemma to descriptor systems, Numerical Analysis Manuscript No. 00–3–09, Bell Laboratories, Murray Hill, NJ. Also available online from:
`http://cm.bell-labs.com/cs/doc/00`.

R. W. Freund and F. Jarre (2003), Numerical computation of nearby positive real systems in the descriptor case, Numerical analysis manuscript, Bell Laboratories, Murray Hill, NJ. In preparation.

R. W. Freund and M. Malhotra (1997), 'A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides', *Linear Algebra Appl.* **254**, 119–157.

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

S. Glavaški, J. E. Marsden and R. M. Murray (1998), Model reduction, centering, and the Karhunen–Loève expansion, in *Proc. 37th IEEE Conference on Decision and Control*, IEEE, Piscataway, NJ, pp. 2071–2076.

K. Glover (1984), 'All optimal Hankel-norm approximations of linear multivariable systems and their l^∞-error bounds.', *Internat. J. Control* **39**, 1115–1193.

W. B. Gragg (1974), 'Matrix interpretations and applications of the continued fraction algorithm', *Rocky Mountain J. Math.* **4**, 213–225.

P. Holmes, J. L. Lumley and G. Berkooz (1996), *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, Cambridge.

S.-Y. Kim, N. Gopal and L. T. Pillage (1994), 'Time-domain macromodels for VLSI interconnect analysis', *IEEE Trans. Computer-Aided Design* **13**, 1257–1270.

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

R. Lozano, B. Brogliato, O. Egeland and B. Maschke (2000), *Dissipative Systems Analysis and Control*, Springer, London.

I. Masubuchi, Y. Kamitane, A. Ohara and N. Suda (1997), 'H_∞ control for descriptor systems: Matrix inequalities approach', *Automatica J. IFAC* **33**, 669–673.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press

B. C. Moore (1981), ‘Principal component analysis in linear systems: Controllability, observability, and model reduction’, *IEEE Trans. Automat. Control* **26**, 17–31.

A. Odabasioglu (1996), ‘Provably passive RLC circuit reduction’, MS thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University.

A. Odabasioglu, M. Celik and L. T. Pileggi (1997), 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, pp. 58–65.

D. P. O’Leary (1980), ‘The block conjugate gradient algorithm and related methods’, *Linear Algebra Appl.* **29**, 293–322.

L. T. Pileggi (1995), 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, pp. 246–253.

L. T. Pillage and R. A. Rohrer (1990), ‘Asymptotic waveform evaluation for timing analysis’, *IEEE Trans. Computer-Aided Design* **9**, 352–366.

V. Raghavan, R. A. Rohrer, L. T. Pillage, J. Y. Lee, J. E. Bracken and M. M. Alaybeyi (1993), AWE-inspired, in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 18.1.1–18.1.8.

M. Rathinam and L. R. Petzold (2002), A new look at proper orthogonal decomposition. Submitted manuscript, University of California, Santa Barbara.

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

R. A. Rohrer, L. Nagel, R. Meyer and L. Weber (1971), ‘Computationally efficient electronic-circuit noise calculations’, *IEEE J. Solid-State Circuits* **6**, 204–213.

J. Roychowdhury (1999), ‘Reduced-order modeling of time-varying systems’, *IEEE Trans. Circuits and Systems, II: Analog and Digital Signal Processing* **46** 1273–1288.

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

T.-J. Su and R. R. Craig, Jr. (1991), ‘Model reduction and control of flexible structures using Krylov vectors’, *J. Guidance Control Dynamics* **14**, 260–267.

F. Tisseur and K. Meerbergen (2001), ‘The quadratic eigenvalue problem’, *SIAM Rev.* **43**, 235–286.

G. C. Verghese, B. C. Lévy and T. Kailath (1981), ‘A generalized state-space for singular systems’, *IEEE Trans. Automat. Control* **26**, 811–831.

C. de Villemagne and R. E. Skelton (1987), ‘Model reductions using a projection formulation’, *Internat. J. Control* **46**, 2141–2169.

J. Vlach and K. Singhal (1994), *Computer Methods for Circuit Analysis and Design*, 2nd edn, Van Nostrand Reinhold, New York.

J. L. Willems (1970), *Stability Theory of Dynamical Systems*, Wiley, New York.

K. Zhou, J. C. Doyle and K. Glover (1996), *Robust and Optimal Control*, Prentice-Hall, Upper Saddle River, NJ.

A. van der Ziel (1986), *Noise in Solid State Devices and Circuits*, Wiley, New York.

https://doi.org/10.1017/S0962492902000120 Published online by Cambridge University Press