Circuit Simulation Techniques Based on Lanczos-Type Algorithms[1]

Roland W. Freund

Abstract. Circuit simulation tasks often require the numerical analysis of large linear networks. These networks can become extremely large, and then the use of time-domain differential equation integration as implemented in SPICE-like circuit simulators, would be inefficient or even prohibitive. A much more efficient approach is to generate reduced-order models of the large networks, based on approximations to the frequency-domain transfer function of the network. This paper describes the computation of Padé-based reduced-order models of large linear networks via Lanczos-type algorithms.

1. Introduction

A circuit is a network of electronic devices, such as resistors, capacitors, inductors, diodes, and transistors. Today's integrated circuits are extremely complex, with up to hundreds of thousands or even millions of devices, and prototyping of such circuits is no longer possible. Instead, computational methods are used to simulate and analyze the behavior of the electronic circuit at the design stage. This allows to correct the design before the circuit is actually fabricated in silicon. It is due to extensive circuit simulation that first-time correct circuits in silicon are almost the norm.

Circuits are characterized by three types of equations: Kirchhoff's current law (KCLs), Kirchhoff's voltage law (KVLs), and the branch constitutive relations (BCRs); see, e.g., [VS]. The unknowns in these equations are the currents through the branches of the network, the voltage drops along the branches, and the voltages at the nodes of the network. The KCLs and KVLs are linear algebraic equations that only depend on the topology of the circuit. The KCLs state that, at each node $\mathcal{N}$ of the network, the currents flowing in and out of $\mathcal{N}$ need to sum up to zero. The KVLs state that, for each closed loop $\mathcal{L}$ of the network, the voltage drops along $\mathcal{L}$ sum up to zero. The BCRs are equations that characterize the actual circuit

[1] Numerical Analysis Manuscript No. 96–19, Bell Laboratories, Murray Hill, New Jersey, October 1996. Available on WWW at `http://cm.bell-labs.com/cs/doc/96`.

elements. For example, the BCR of a linear resistor is simply 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.

The KCLs, KVLs, and BCRs can be summarized as a system of first-order differential-algebraic equations (DAEs),

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

together with suitable initial conditions. In (1.1), q and f are given vector-valued functions, and $x = x(t)$ is the unknown vector of circuit variables. Traditional SPICE-like circuit simulators are based on the numerical solution of the system of DAEs (1.1). However, for very large circuits, such a time-domain integration of (1.1) is often inefficient or even prohibitive.

A much more efficient approach is to generate reduced-order models, based on approximations to the frequency-domain transfer function of the linearized circuit or some large linear subcircuit. Instead of the general system (1.1), we now start with the DAEs for a linear(ized) time-invariant (sub)circuit:

$$\begin{aligned} C\frac{dx}{dt} &= -Gx + Bu(t), \\ y(t) &= E^T x(t). \end{aligned} \tag{1.2}$$

Here, $C, G \in \mathbb{C}^{N\times N}$, $B \in \mathbb{C}^{N\times m}$, and $E \in \mathbb{C}^{N\times p}$ are given matrices, m and p is the number of inputs and outputs, respectively, the components of the given vector-valued function $u\colon [0,\infty) \mapsto \mathbb{C}^m$ are the inputs, and $y\colon [0,\infty) \mapsto \mathbb{C}^p$ is the unknown function of outputs. Denoting by

$$\hat{x}(z) = \int_0^\infty x(t)e^{-zt}\,dt, \quad \hat{u}(z) = \int_0^\infty u(t)e^{-zt}\,dt, \quad \hat{y}(z) = \int_0^\infty y(t)e^{-zt}\,dt$$

the Laplace transform of x, u, and y, respectively, and assuming, for simplicity, zero initial conditions, the system of DAEs (1.2) can be stated in frequency-domain as follows:

$$\begin{aligned} zC\hat{x} &= -G\hat{x} + B\hat{u}(z), \\ \hat{y}(z) &= E^T\hat{x}(z). \end{aligned} \tag{1.3}$$

From (1.3), one obtains the input-output relation $\hat{y}(z) = H(z)\hat{u}(z)$, where $H\colon \mathbb{C} \mapsto (\mathbb{C}\cup\{\infty\})^{p\times m}$ is a rational matrix-valued function defined by

$$H(z) = E^T\,(G + zC)^{-1}\,B. \tag{1.4}$$

The function is called the *transfer function* of the m-input p-output time-invariant linear dynamical system given by (1.2). Note that, in general, the transfer function (1.4) has N poles, namely the eigenvalues of the pencil $G + zC$. In particular, if N is large, then the rational function H is of very high order and working directly with H becomes prohibitive. Instead, one approximates H by a rational function H_n of low order $n \ll N$ that describes a reduced-order model of the linear dynamical system (1.2). One possible approach to constructing suitable reduced-order models is Padé approximation; see, e.g. [BG, BV].

For the large linear networks arising in the simulation of electronic circuits, Padé-based reduced-order models have proven to be especially efficient. This was first observed by Huang, Pillage, and Rohrer [H, PR] who, for the single-input single-output case ($m = p = 1$), advocated the use of *asymptotic waveform evaluation* (AWE). The AWE method is based on Padé approximations of the transfer function H, which is now scalar-valued since $m = p = 1$. Furthermore, in AWE, the Padé approximations are generated in the usual way via explicit computation of the underlying moments of the network transfer function. However, this is a numerically unstable procedure. Indeed, in practice, AWE is limited to generating Padé approximants of very low order n only. On the other hand, the numerical problems with AWE can easily be remedied by exploiting the well-known connection [Gr, GL, Gu] between Padé approximants of scalar-valued transfer functions and the classical Lanczos process [L] and generating the Padé approximants via the Lanczos algorithm; see [FeF1, FeF2, GGV]. Feldmann and Freund [FeF3, F2] have extended this approach to the general multi-input multi-output case and shown how matrix Padé approximants to matrix-valued transfer functions can be obtained from a novel Lanczos-type algorithm that was recently developed by Aliaga, Boley, Freund, and Hernández [ABFH].

In this paper, we describe the computation of Padé-based reduced-order models of large linear networks via Lanczos-type algorithms.

2. Padé approximants of transfer functions

Let H be a given matrix-valued transfer function of the form (1.4), where $C, G \in \mathbb{C}^{N\times N}$, $B \in \mathbb{C}^{N\times m}$, and $E \in \mathbb{C}^{N\times p}$ are matrices. Furthermore, we assume that $G+zC$ is a *regular* pencil, i.e., the matrix $G+zC$ is nonsingular for at least one $z \in \mathbb{C}$.

Next, we introduce our notion of an n-th matrix Padé approximant

of H. First, we choose a suitable expansion point $z_0 \in \mathbb{C}$ such that the matrix $G + z_0 C$ is nonsingular. Let L and U be nonsingular $N \times N$ matrices such that

$$L \cdot U = G + z_0 C, \tag{2.1}$$

and set

$$A = -L^{-1} C U^{-1}, \quad R = L^{-1} B, \quad \text{and} \quad L = U^{-T} E. \tag{2.2}$$

Using (2.1) and (2.2), and setting $z = z_0 + \sigma$ in (1.4), we can rewrite (1.4) as follows:

$$H(z_0 + \sigma) = L^T \left(I - \sigma A\right)^{-1} R. \tag{2.3}$$

We remark that in circuit simulation, the matrices G and C are sparse. A factorization (2.1) can then be computed by sparse Gaussian elimination, resulting in factors L and U that are usually also sparse. Although L, C, U are sparse, the matrix A in (2.2) is in general dense. However, Lanczos-type algorithms only use A in the form of matrix-vector products $A \cdot v$ and $A^T \cdot w$, and for computing these, there is no need to explicitly form the matrix A. Indeed, be means of

$$A \cdot v = -L^{-1} \left(C \left(U^{-1} v\right)\right),$$

the product $A \cdot v$ can be obtained by a sparse backsolve with U, a multiplication with the sparse matrix C, and a sparse backsolve with A^T. Computing products $A^T \cdot w$ is done similarly.

We say that an $p \times m$-matrix-valued function $H_n \colon \mathbb{C} \mapsto (\mathbb{C} \cup \{\infty\})^{p \times m}$ is a rational function of *type* $(n-1, n)$ if H_n can be represented as

$$H_n(z_0 + \sigma) = L_n^T \left(I - \sigma A_n\right)^{-1} R_n, \tag{2.4}$$

with matrices $A_n \in \mathbb{C}^{n \times n}$, $R_n \in \mathbb{C}^{n \times m}$, and $L_n \in \mathbb{C}^{n \times p}$. An nth *Padé approximant* (about the expansion point z_0) $H_n(z_0 + \sigma)$ of the $p \times m$-matrix-valued transfer function $H(z_0 + \sigma)$ is a rational $p \times m$-matrix-valued function of type $(n-1, n)$ whose Taylor expansion about $\sigma = 0$ matches as many leading Taylor coefficients of H as possible, i.e.,

$$H_n(z_0 + \sigma) = H(z_0 + \sigma) + \mathcal{O}(\sigma^{m(n)}),$$

where the integer $m(n)$ is maximal. Note that, by (2.3),

$$H(z_0 + \sigma) = \sum_{i=0}^{\infty} M_i \sigma^i, \tag{2.5}$$

where the Taylor coefficients M_i are $p \times m$ matrices given by

$$M_i = L^T A^i R, \quad i = 0, 1, \ldots . \tag{2.6}$$

The coefficients (2.6) are called the *moments* of the transfer function H.

We remark that a matrix-valued rational function of type $(n-1, n)$ can always be written in terms of a pair of matrix polynomials. However, since matrix polynomials do not commute and also for other reasons, such a representation involves a number of difficulties (see the discussion in [XB]) that can be avoided by using (2.4). Furthermore, the representation (2.4) is also natural, in that it reflects the intimate connection between matrix Padé approximants of transfer functions and the minimal partial realization problem in systems theory; see, e.g., [K, T].

3. The single-input single-output case

In this section, we consider the single-input single-output case, $m = p = 1$. Then, in (2.3), $R, L \in \mathbb{C}^N$, and H is a scalar rational function. Furthermore, the nth Padé approximant H_n can be represented as

$$H_n(z_0 + \sigma) = \frac{\varphi_{n-1}(\sigma)}{\psi_n(\sigma)}, \tag{3.1}$$

where φ_{n-1} and ψ_n are polynomials of degree at most $n-1$ and n, respectively. The standard approach (and the one employed in AWE [PR]) to computing H_n is to first generate the $2n$ moments $M_0, M_1, \ldots, M_{2n-1}$, followed by the solution of a linear system with a Hankel matrix whose entries are the moments, to obtain the coefficients of the polynomials φ_{n-1} and ψ_n in (3.1). Unfortunately, this approach is numerically unstable due to the typical extreme ill-conditioning of the Hankel matrix, and also because, in finite-precision arithmetic, the moments M_i with increasing i convey only information about the extreme poles of the transfer function H; see, e.g., [FeF1, FeF2, GGV]. As a result, moment-based approaches, such as AWE, can only be used to compute Padé approximants of very moderate order n, and typically, they are limited to $n \leq 10$.

Next, we describe the *Padé-via-Lanczos* (PVL) algorithm [FeF1, FeF2] to compute H_n. Instead of (3.1), the PVL approach uses the following well-known Lanczos-Padé connection (see, e.g., [Gr, GL, Gu]):

$$H_n(z_0 + \sigma) = \left(\eta D_n^T e_1\right)^T \left(I - \sigma T_n\right)^{-1} \left(\rho e_1\right), \tag{3.2}$$

where $e_1 = [1 \quad 0 \quad \cdots \quad 0]^T \in \mathbb{R}^n$. All the quantities involved in (3.2) are generated in the course of the first n steps of the Lanczos algorithm [L].

Starting with $v_1 = R/\rho$ and $w_1 = L/\eta$, where $\rho, \eta > 0$ are arbitrary normalization factors, the first n Lanczos steps produce two sequences of vectors,

$$v_1, v_2, \ldots, v_n \quad \text{and} \quad w_1, w_2, \ldots, w_n, \tag{3.3}$$

such that

$$\begin{aligned} \text{span}\{\, v_1, v_2, \ldots, v_n \,\} &= \text{span}\{\, R, AR, \ldots, A^{n-1}R \,\}, \\ \text{span}\{\, w_1, w_2, \ldots, w_n \,\} &= \text{span}\{\, L, A^T L, \ldots, (A^T)^{n-1}L \,\}, \end{aligned} \tag{3.4}$$

and the vectors (3.3) are biorthogonal:

$$w_i^T v_l = 0 \quad \text{for all} \quad i \neq l, \; i, l = 1, 2, \ldots, n. \tag{3.5}$$

Setting

$$V_n = [\, v_1 \quad v_2 \quad \cdots \quad v_n \,] \quad \text{and} \quad W_n = [\, w_1 \quad w_2 \quad \cdots \quad w_n \,], \tag{3.6}$$

the matrices D_n and T_n in (3.2) are now defined as

$$D_n = W_n^T V_n \quad \text{and} \quad T_n = D_n^{-1} W_n^T A V_n.$$

Note that, by (3.5), the matrix D_n is diagonal. Moreover, the vectors (3.3) can be generated by three-term recurrences, and the entries of T_n are just the coefficients of these recursions. Consequently, the matrix T_n is tridiagonal. Finally, we remark that, in the classical Lanczos process [L], so-called breakdowns or near-breakdowns due to division by zero or a near-zero number can occur. However, these problems can be remedied by incorporating *look-ahead* techniques into the Lanczos process; see, e.g., [FGN]. The basic idea is to relax the biorthogonality (3.5) slightly whenever a breakdown or near-breakdown would occur in the classical Lanczos algorithm. To keep the exposition as simple as possible, in this paper, we always assume that breakdowns will not occur in the Lanczos process. However, we stress that both the Lanczos-Padé connection (3.2) and the result in Theorem 5.1 below also hold true if look-ahead Lanczos steps are performed to avoid breakdowns or near-breakdowns. Only the structure of the matrices D_n and T_n are affected by look-ahead steps. More precisely, in (3.2), D_n is block diagonal and T_n is block tridiagonal when look-ahead steps occur.

Next, we present two numerical examples (taken from [FeF2]) from circuit simulation to illustrate the behavior of the PVL algorithm for computing nth Padé approximants (3.1). In both cases, we plot the absolute values

$$|H_n(2\pi \mathrm{i}\omega)| \quad \text{and} \quad |H(2\pi \mathrm{i}\omega)|$$

of H_n and the original transfer function H, respectively, over some relevant frequency range $0 \leq \omega_{\min} \leq \omega \leq \omega_{\max}$; here $\mathrm{i} = \sqrt{-1}$.

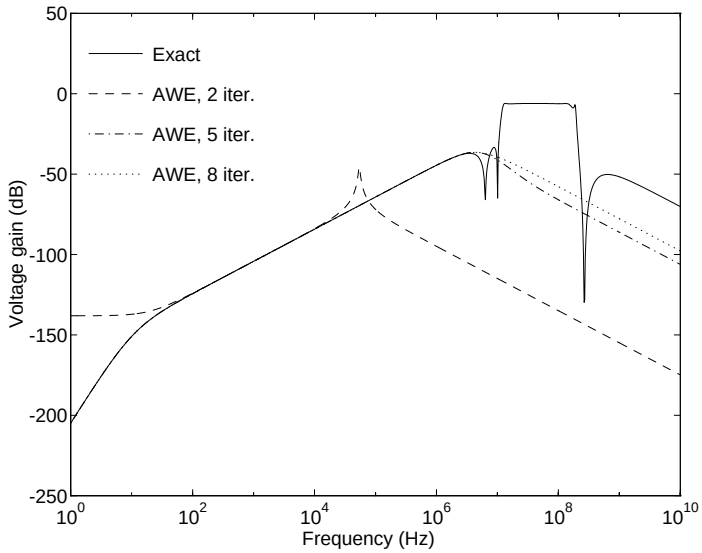

Figure 1: Results for simulation of voltage gain with AWE.

Example 3.1 *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 1, we show the computed* $|H_n|$ *for* $n = 2, 5, 8$, *together with the exact* $|H|$. *Note that* H_8 *clearly has not yet converged to* H. *On the other hand, it turned out that the* H_n*'s practically did not change anymore for* $n \geq 8$, *and in particular, AWE never converged in this example. Next, we ran this example with PVL, and in Figure 2, we show the computed results for* $n = 2, 8, 28$, *together with the exact* $|H|$. *Note that the results for* $n = 8$ *(the dotted curves) in Figures 1 and 2 are vastly different, although they both correspond to the same function* H_8. *The reason for this is that AWE is numerically unstable, while PVL generates* H_8 *stably. Furthermore, note that PVL converges, with the computed* 28*th Padé approximant being practically identical to* H.

Example 3.2 *The second example is a larger circuit (the so-called PEEC circuit) with 2100 capacitors, 172 inductors, and 6990 inductive couplings. Here the frequency range of interest is* $0 \leq \omega \leq 5 \times 10^9$. *In Figure 3, we show the exact transfer function, together with the* 60*-th Padé approximant as computed by PVL.*

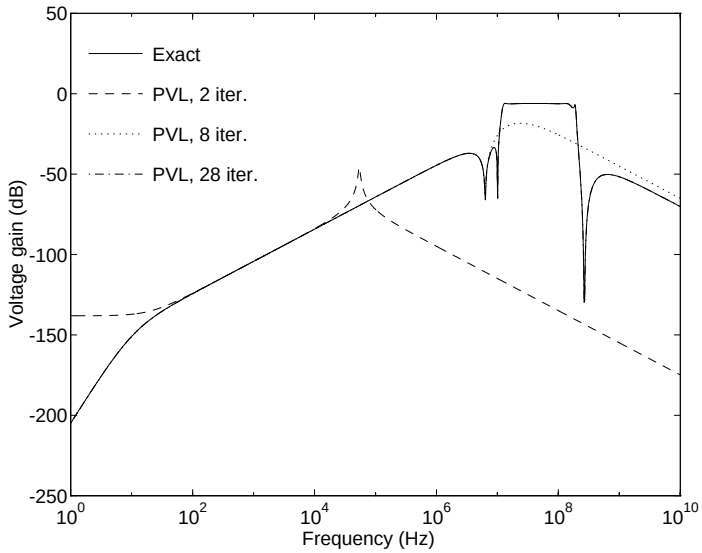

Figure 2: Results for simulation of voltage gain with PVL.

Remark 3.3 *It is well known that, in general, Padé-based reduced-order models need not be stable, even though the original transfer function H was stable. Recall that the function H is stable if all its poles have non-positive real parts and possible poles on the imaginary axis are simple. However, stability of Padé-based reduced-order models can be guaranteed for important special classes of linear networks, namely RC, RL, and LC circuits consisting of only resistors and capacitors, resistors and inductors, and inductors and capacitors, respectively; see* [FrF2].

Remark 3.4 *In circuit simulation, it is often crucial to compute sensitivities of the transfer function or its poles or zeros, with respect to certain circuit parameters. An extension of PVL that allows such sensitivity computations is described in* [FrF1].

4. A Lanczos-type algorithm for multiple starting vectors

We now return to the general m-input p-output case, with *arbitrary* $m, p \geq 1$. A Lanczos-type process connected to matrix Padé approximants of transfer functions H given by (2.3) needs to be able to han-

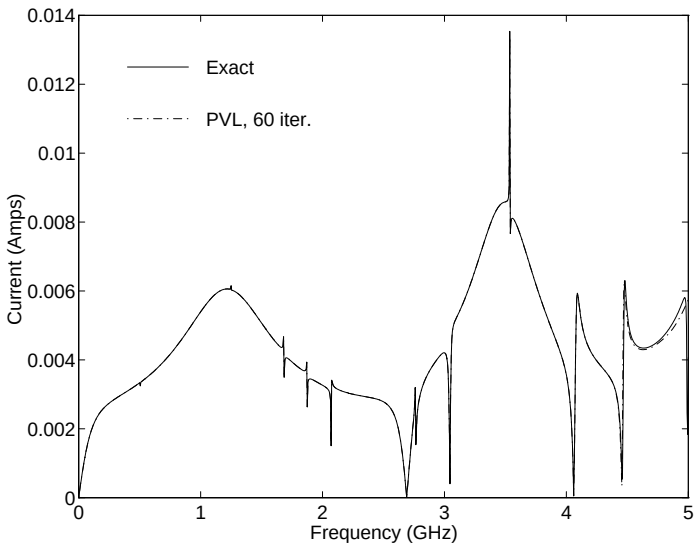

Figure 3: Results for the PEEC circuit, 60 PVL iterations.

dle multiple starting vectors, namely the m, respectively p, columns of the matrices R, respectively L, in (2.3), where, in general, m and p can be different. Such an algorithm was recently developed by Aliaga, Boley, Freund, and Hernández [ABFH], while all earlier algorithms (see, e.g., [KC1, KC2, S, SC]) are restricted to the special case $m = p$. In this section, we briefly describe the method proposed in [ABFH].

Like the classical Lanczos process, the algorithm for multiple starting vectors generates two sequences of vectors (3.3) that are again constructed to fulfill the biorthogonality relations (3.5). However, instead of (3.4), the vectors (3.3) now span the subspaces generated by the first n linearly independent columns of the *Krylov matrices*

$$\begin{aligned} K^{(r)} &= [\,R \quad AR \quad A^2R \quad \cdots \quad A^{N-1}R\,] \\ \text{and} \quad K^{(l)} &= [\,L \quad A^TL \quad (A^T)^2L \quad \cdots \quad (A^T)^{N-1}L\,], \end{aligned} \tag{4.1}$$

respectively. In particular, the process of constructing the vectors (3.3) now also needs to be able to detect linearly dependent columns in (4.1) and to delete these vectors. We refer to this process as *deflation*. In our Lanczos process, a linearly dependent vector occurs if, and only if, $v_n = 0$ or $w_n = 0$, and deflation simply consists of deleting v_n or w_n, respectively.

As a result, the Lanczos vectors (3.3) now span the spaces generated by the first n columns of the *deflated* Krylov matrices

$$\begin{aligned} K_{\text{defl}}^{(r)} &= [\,R_1 \quad AR_2 \quad A^2R_3 \quad \cdots \quad A^{j_{\max}-1}R_{j_{\max}}\,] \\ \text{and} \quad K_{\text{defl}}^{(l)} &= [\,L_1 \quad A^TL_2 \quad (A^T)^2L_3 \quad \cdots \quad (A^T)^{k_{\max}-1}L_{k_{\max}}\,], \end{aligned} \tag{4.2}$$

respectively. The matrices (4.2) are obtained from (4.1) by deleting all linearly dependent vectors that one encounters when scanning the columns of (4.1) from left to right. In (4.2), $R_j \in \mathbb{C}^{N\times m_j}$ and $L_k \in \mathbb{C}^{N\times p_k}$ are submatrices of R and L, respectively. Moreover, R_j is a submatrix of R_{j-1}, and L_j is a submatrix of L_{j-1}, so that $m_1 \geq m_2 \geq \cdots \geq m_{j_{\max}} \geq 1$ and $p_1 \geq p_2 \geq \cdots \geq p_{k_{\max}} \geq 1$.

We now set $N_0 = \min\{\operatorname{rank} K^{(r)}, \operatorname{rank} K^{(l)}\}$. Then, for each integer n with $\max\{m_1, p_1\} \leq n \leq N_0$, we define $j = j(n)$ and $k = k(n)$ as the maximal integers such that

$$m_1 + m_2 + \cdots + m_j \leq n \quad \text{and} \quad p_1 + p_2 + \cdots + p_k \leq n, \tag{4.3}$$

respectively. The integers $j(n)$ and $k(n)$ will be used in §4 to state our main result. We remark that if no deflation occurs, then $j(n) = \lfloor n/m \rfloor$ and $k(n) = \lfloor n/p \rfloor$.

After these preliminaries, we can now present a sketch of the actual algorithm. First, the starting vectors are biorthogonalized, resulting in relations of the following form:

$$R = [\,v_1 \quad v_2 \quad \cdots \quad v_{m_1}\,]\,\rho \quad \text{and} \quad L = [\,w_1 \quad w_2 \quad \cdots \quad w_{p_1}\,]\,\eta,$$

where $\rho \in \mathbb{C}^{m_1\times m}$ and $\eta \in \mathbb{C}^{p_1\times p}$. Moreover, we initialize the indices $m_c = m_1$ and $p_c = p_1$, which are used to keep track of the current block sizes; m_c respectively p_c are reduced by 1 every time deflation occurs.

Then, for $n = 1, 2, \ldots, N_0 - \max\{m_c, p_c\}$, we repeat the following steps:

Set $\quad v_{n+m_c} = Av_n - \displaystyle\sum_{i=n-p_c}^{n+m_c-1} v_i \,\frac{w_i^T A v_n}{w_i^T v_i}$;

If $v_{n+m_c} = 0$, then delete v_{n+m_c} and set $m_c = m_c - 1$;

Set $\quad w_{n+p_c} = A^Tw_n - \displaystyle\sum_{i=n-m_c}^{n+p_c-1} w_i \,\frac{v_i^T A^T w_n}{w_i^T v_i}$;

If $w_{n+p_c} = 0$, then delete w_{n+p_c} and set $p_c = p_c - 1$.

Using the matrices V_n and W_n introduced in (3.6), the above recurrences to generate the Lanczos vectors (3.3) can be summarized in compact

matrix form as follows:

$$\begin{aligned} AV_n &= V_n T_n + \big[\underbrace{0 \quad 0 \quad \cdots \quad 0}_{n-m_c} \quad \underbrace{\star \quad \star \quad \cdots \quad \star}_{m_c}\big], \\ A^T W_n &= W_n \tilde{T}_n^T + \big[\underbrace{0 \quad 0 \quad \cdots \quad 0}_{n-p_c} \quad \underbrace{\star \quad \star \quad \cdots \quad \star}_{p_c}\big]. \end{aligned} \tag{4.4}$$

Here, T_n and $\tilde{T}_n$ are $n \times n$ banded matrices with lower bandwidth $m+1$ and upper bandwidth $p+1$. Moreover, each deflation step reduces the lower or upper bandwidth by 1. Finally, the matrices T_n and $\tilde{T}_n$ are connected by the relation

$$\tilde{T}_n = D_n T_n D_n^{-1}, \quad \text{where} \quad D_n = W_n^T V_n. \tag{4.5}$$

Remark 4.1 *In the above description of the algorithm, we assumed, for simplicity, that only zero vectors are deflated. Of course, in finite-precision arithmetic, also nonzero vectors that are in some sense close to zero need to deflated. Such inexact deflation is described in detail in* [ABFH].

5. A connection to matrix Padé approximation

We continue to use the notation introduced in Section 4. In particular, ρ, η, D_n, and V_n are the quantities generated by the Lanczos-type process. We denote by $0_{i\times l}$ the $i \times l$ zero matrix. We now establish the following connection between the Lanczos-type process and matrix Padé approximants.

Theorem 5.1 *Let* $\max\{m_1, p_1\} \le n \le N_0$, *and let* $j = j(n)$ *and* $k = k(n)$ *be the integers defined in* (4.3). *Then,*

$$H_n(z_0 + \sigma) = \begin{bmatrix} D_{p_1}^T \eta \\ 0_{(n-p_1)\times p} \end{bmatrix}^T (I - \sigma T_n)^{-1} \begin{bmatrix} \rho \\ 0_{(n-m_1)\times m} \end{bmatrix} \tag{5.1}$$

is an nth Padé approximant of the function H *defined in* (2.3), *and*

$$H_n(z_0 + \sigma) = H(z_0 + \sigma) + \mathcal{O}(\sigma^{j+k}). \tag{5.2}$$

Moreover, if $n = m_1 + m_2 + \cdots + m_j$ *or* $n = p_1 + p_2 + \cdots + p_k$, *then* H_n *is a Padé approximant of minimal order* n *that matches the first* $j + k$ *moments* (2.6), $M_0, M_1, \ldots, M_{j+k-1}$, *of* H.

Proof: Using (4.4), the band structure of T_n and $\tilde{T}_n$, the definition of j

and k in (4.3), and (4.5), one can verify the following relations:

$$\begin{aligned} A^{i_1} R &= V_n T_n^{i_1} \begin{bmatrix} \rho \\ 0_{(n-m_1)\times m} \end{bmatrix}, \quad i_1 = 0, 1, \ldots, j-1, \\ L^T A^{i_2} &= [\, \eta^T \quad 0_{p\times(n-p_1)} \,] \tilde{T}_n^{i_2} W_n^T, \\ &= \begin{bmatrix} D_{p_1}^T \eta \\ 0_{(n-p_1)\times p} \end{bmatrix}^T T_n^{i_2} D_n^{-1} W_n^T, \quad i_2 = 0, 1, \ldots, k-1. \end{aligned}$$

These relations, together with (2.6) and $D_n = W_n^T V_n$, imply that

$$M_{i_1+i_2} = L^T A^{i_1+i_2} R = \begin{bmatrix} D_{p_1}^T \eta \\ 0_{(n-p_1)\times p} \end{bmatrix}^T T_n^{i_1+i_2} \begin{bmatrix} \rho \\ 0_{(n-m_1)\times m} \end{bmatrix} \tag{5.3}$$

for all $i_1 + i_2 = 0, 1, \ldots, j+k-2$. Using (4.4) and the fact that the Lanczos vectors are biorthogonal, it is possible to show that (5.3) even holds true for $i_1 + i_2 = j + k - 1$. Thus, by (5.3), the first $j + k$ coefficients of the Taylor expansions about $\sigma = 0$ of the functions (2.5) and (5.1) match, and this concludes the proof of (5.2). In remains to verify that the number $j+k$ of matched coefficients is maximal so that H_n is indeed an nth Padé approximant. This can be done by exploiting the connection between matrix Padé approximants and the partial minimal realization problem and by showing that the criterion given in [GKL, Theorem 0.1] is satisfied. ■

6. Some related work

For the single-input single-output case, Grimme, Sorensen, and Van Dooren [GSV] proposed to use an implicitly restarted Lanczos algorithm to remedy the possible instability of Padé-based reduced-order models. The basic idea here is to adaptively choose a polynomial ψ that dampens out unstable poles of the reduced-order model, and then implicitly run n steps of the Lanczos algorithm applied to A with modified starting vectors $\psi(A)R$ and $\psi(A^T)L$. This results in some Padé-type approximation

$$\begin{aligned} H(z_0 + \sigma) &= L^T (I - \sigma A)^{-1} R \\ &\approx \big(\psi(A^T)L\big)^T (I - \sigma A)^{-1} (\psi(A)R) \\ &\approx L^T R \, e_1^T (I - \sigma T_n)^{-1} e_1 =: \tilde{H}_n(z_0 + \sigma), \end{aligned}$$

where H and $\tilde{H}_n$ match in the $2n$ *modified* moments

$$L^T A^j \left(\psi(A)\right)^2 R, \quad j = 0, 1, \ldots, 2n-1.$$

With the same amount of computational effort, PVL would generate a true Padé approximation that matches the $2(n + \deg \psi)$ moments $LA^j R$, $j = 0, 1, \ldots, 2(l + \deg \psi) - 1$.

An alternative to Padé approximation is interpolation of $H(z_0 + \sigma_j)$ at, say s, carefully selected nonzero points $\sigma_j \in \mathbb{C}$, $j = 1, 2, \ldots, n$. This approach now requires the computation of

$$H(z_0 + \sigma_j) = -\frac{1}{\sigma_j} L^T \left(A - (1/\sigma_j)I\right)^{-1} R = -\frac{1}{\sigma_j} L^T X^{(j)}, \quad j = 1, 2, \ldots, s,$$

where $X^{(j)}$ is the solution the block system

$$\left(A - (1/\sigma_j)I\right) X^{(j)} = R. \tag{6.1}$$

Freund and Malhotra [FM] developed a block-QMR (BL-QMR) algorithm for the iterative solution of block systems such as (6.1). In fact, BL-QMR is also based on the Lanczos-type algorithm sketched in Section 4. Note that s systems of the form (6.1) need to be solved. These systems all have the same right-hand side, R, and their coefficient matrices differ from the fixed matrix A only by different shifts $(1/\sigma_j)I$. The solution of such shifted systems requires only one run of the underlying Lanczos-type algorithm (applied to A), resulting in considerable computational savings when BL-QMR is used to solve the s shifted block systems (6.1); see, e.g., [F1].

References

[ABFH] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernández, A Lanczos-type algorithm for multiple starting vectors, Numerical Analysis Manuscript No. 96–18, Bell Laboratories, Murray Hill, NJ, 1996.

[BG] G. A. Baker, Jr. and P. Graves-Morris, *Padé Approximants, Second Edition*, Cambridge University Press, New York, 1996.

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

[FeF1] P. Feldmann and R. W. Freund, Efficient linear circuit analysis by Padé approximation via the Lanczos process, in: *Proc. EURO-DAC '94 with EURO-VHDL '94*, IEEE Computer Society Press, Los Alamitos, CA, 1994, 170–175.

[FeF2] P. Feldmann and R. W. Freund, Efficient linear circuit analysis by Padé approximation via the Lanczos process, *IEEE Trans. Computer-Aided Design* **14** (1995), 639–649.

[FeF3] P. Feldmann and R. W. Freund, Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm, in: *Proc. 32nd Design Automation Conference*, ACM, New York, 1995, 474–479.

[F1] R. W. Freund, Solution of shifted linear systems by quasi-minimal residual iterations, in: *Numerical Linear Algebra*, ed. L. Reichel, A. Ruttan, and R. S. Varga, W. de Gruyter, Berlin, 1993, 101–121.

[F2] R. W. Freund, Computation of matrix Padé approximations of transfer functions via a Lanczos-type process, in: *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, ed. C. K. Chui and L. L. Schumaker, World Scientific Publishing Co., Inc., Singapore, 1995, 215–222.

[FrF1] R. W. Freund and P. Feldmann, Small-signal circuit analysis and sensitivity computations with the PVL algorithm, *IEEE Trans. Circuits and Systems—II: Analog and Digital Signal Processing* **43** (1996), 577–585.

[FrF2] R. W. Freund and P. Feldmann, Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm, in: *Tech. Dig. 1996 IEEE/ACM International Conference on Computer-Aided Design*, IEEE Computer Society Press, Los Alamitos, CA, 1996, to appear.

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

[FM] R. W. Freund and M. Malhotra, A block-QMR algorithm for non-Hermitian linear systems with multiple right-hand sides, Numerical Analysis Manuscript No. 95–09, AT&T Bell Laboratories, Murray Hill, NJ, 1995. *Linear Algebra Appl.*, to appear.

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

[GKL] I. Gohberg, M. A. Kaashoek, and L. Lerer, On minimality in the partial realization problem, *Systems Control Lett.* **9** (1987), 97–104.

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

[GL] W. B. Gragg and A. Lindquist, On the partial realization problem, *Linear Algebra Appl.* **50** (1983), 277–319.

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

[Gu] M. H. Gutknecht, A completed theory of the unsymmetric Lanczos process and related algorithms, part I, *SIAM J. Matrix Anal. Appl.* **13** (1992), 594–639.

[H] X. Huang, Padé approximation of linear(ized) circuit responses, Ph.D. thesis, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 1990.

[K] T. Kailath, *Linear Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1980.

[KC1] H. M. Kim and R. R. Craig, Jr., Structural dynamics analysis using an unsymmetric block Lanczos algorithm, *Internat. J. Numer. Methods Engrg.* **26** (1988), 2305–2318.

[KC2] H. M. Kim and R. R. Craig, Jr., Computational enhancement of an unsymmetric block Lanczos algorithm, *Internat. J. Numer. Methods Engrg.* **30** (1990), 1083–1089.

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

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

[S] T.-J. Su, A decentralized linear quadratic control design method for flexible structures, Ph.D. thesis, Dept. of Aerospace and Engineering Mechanics, The University of Texas at Austin, 1989.

[SC] T.-J. Su and R. R. Craig, Jr., Model reduction and control of flexible structures using Krylov vectors, *J. Guidance Control Dynamics* **14** (1991), 260–267.

[T] A. J. Tether, Construction of minimal linear state-variable models from finite input-output data, *IEEE Trans. Automat. Control* **AC-15** (1970), 427–436.

[VS] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design, Second Edition*, Van Nostrand Reinhold, New York, 1993.

[XB] G.-L. Xu and A. Bultheel, Matrix Padé approximation: definitions and properties, *Linear Algebra Appl.* **137/138** (1990), 67–136.

Bell Laboratories
Lucent Technologies
Room 2C–420, 700 Mountain Avenue
Murray Hill, New Jersey 07974–0636