Computing Minimal Partial Realizations Via a Lanczos-Type Algorithm for Multiple Starting Vectors*

Roland W. Freund

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

E-mail: freund@research.bell-labs.com

Abstract

We describe a Lanczos-type procedure that reduces a given realization of a finite sequence of (moment) matrices to a minimal partial realization. A key feature of this procedure is that the underlying Lanczos-type algorithm is directly applied to the matrix triplet describing the given realization, rather than to the moment matrices. It thus avoids explicit formulation of and the usually unstable computation with the moment matrices.

1 Introduction

Given a finite sequence of matrices,

$$\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_{r-1} \in \mathbb{C}^{p\times m}, \tag{1}$$

the *partial realization problem* is to find a triplet of matrices

$$\mathbf{A}_n \in \mathbb{C}^{n\times n}, \quad \mathbf{B}_n \in \mathbb{C}^{n\times m}, \quad \mathbf{C}_n \in \mathbb{C}^{n\times p}, \tag{2}$$

such that

$$\mathbf{M}_i = \mathbf{C}_n^{\mathrm{T}} \mathbf{A}_n^i \mathbf{B}_n \quad \text{for all} \quad i = 0, 1, \ldots, r-1. \tag{3}$$

A partial realization (2) is called *minimal* if the state-space dimension n in (2) and (3) is as small as possible. The minimal partial realization problem has been studied extensively; see, e.g., [2, 3, 5, 6, 7, 8, 15, 18, 22, 23] and the references given therein.

However, there is still a need to develop robust and numerically stable algorithms for the solution of minimal partial realization problems, especially those arising in the context of

*Numerical Analysis Manuscript No. 97–3–07, Bell Laboratories, Murray Hill, New Jersey, September 1997. Available on WWW at {`http://cm.bell-labs.com/cs/doc/97`}.

large-scale dynamical systems. For example, the simulation of VLSI circuits often involves linear time-invariant dynamical systems with state-space dimensions N that can be of the order of millions. One way of tackling these huge systems is to approximate them by much smaller reduced-order models that are obtained via the solution of minimal partial realization problems; see, e.g., [9, 10, 11] and the references given therein. In these cases, the matrices (1) are defined as the Markov parameters of a linear time-invariant dynamical system of the form

$$\begin{aligned} \mathbf{C}\frac{d\mathbf{x}}{dt} &= -\mathbf{G}\,\mathbf{x} + \mathbf{B}\,\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{E}^{\mathrm{T}}\mathbf{x}(t). \end{aligned} \tag{4}$$

Here, $\mathbf{C}$, $\mathbf{G} \in \mathbb{C}^{N\times N}$, $\mathbf{B} \in \mathbb{C}^{N\times m}$, and $\mathbf{E} \in \mathbb{C}^{N\times p}$ are given matrices, and m and p denote the numbers of input and outputs, respectively. The behavior of the linear system (4) is characterized by its Laplace-domain transfer function,

$$\mathbf{H}(z) = \mathbf{E}^{\mathrm{T}}\left(\mathbf{G} + z\mathbf{C}\right)^{-1}\mathbf{B}. \tag{5}$$

In the case of VLSI circuits, the matrices $\mathbf{G}$ and $\mathbf{C}$ are singular in general, however, $\mathbf{G}+z\mathbf{C}$ is a regular pencil. Hence, there exists $z_0 \in \mathbb{C}$ such that the matrix $\mathbf{G} + z_0\mathbf{C}$ is nonsingular. Selecting such a point z_0 and using a factorization of the type

$$\mathbf{G} + z_0\mathbf{C} = \mathbf{F}_1 \cdot \mathbf{F}_2, \quad \mathbf{F}_1, \mathbf{F}_2 \in \mathbb{C}^{N\times N}, \tag{6}$$

we can rewrite (5) as follows:

$$\mathbf{H}(z) = \mathbf{L}^{\mathrm{T}}\left(\mathbf{I} - (z - z_0)\mathbf{A}\right)^{-1}\mathbf{R}. \tag{7}$$

Here, $\mathbf{A}$ is an $N \times N$ matrix, $\mathbf{R}$ is an $N \times m$ matrix, and $\mathbf{L}$ is an $N \times p$ matrix defined by

$$\mathbf{A} := -\mathbf{F}_1^{-1}\mathbf{C}\mathbf{F}_2^{-1}, \quad \mathbf{R} := \mathbf{F}_1^{-1}\mathbf{B}, \quad \mathbf{L} := \mathbf{F}_2^{-\mathrm{T}}\mathbf{E}. \tag{8}$$

Moreover, $\mathbf{I}$ denotes the $N \times N$ identity matrix. From (7), we obtain the expansion

$$\mathbf{H}(z) = \sum_{i=0}^{\infty} \mathbf{M}_i (z - z_0)^i$$

of $\mathbf{H}$ about the point z_0. Here,

$$\mathbf{M}_i = \mathbf{L}^{\mathrm{T}}\mathbf{A}^i\mathbf{R}, \quad i = 0, 1, \ldots\,, \tag{9}$$

are the corresponding Markov parameters.

We remark that, for VLSI circuits, the matrices $\mathbf{G}$ and $\mathbf{C}$ are sparse, and typically, the factorization (6) can be chosen as a sparse LU decomposition. By (8), the matrix $\mathbf{A}$ can then be represented as a product of sparse matrices and inverses of sparse matrices. Using this representation, matrix-vector products with $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$ can be computed relative cheaply, while it would be prohibitive to actually form and store the $N \times N$ matrix $\mathbf{A}$ for large N. This property could, in principle, be used to first compute the matrices $\mathbf{M}_i$,

$i = 0, 1, \ldots, r-1$, by means of (9), and then to solve the minimal partial realization problem starting from the data (1). However, any approach based on explicitly forming the matrices (9) is numerically unstable and can only be employed for very moderate values of r, typically $r \leq 20$; see the discussion in [9, 10].

For the case of single-input single-output systems, i.e., $m = p = 1$, it is well known that explicitly forming the matrices (9) can easily be avoided by employing the so-called Lanczos-Padé connection [16, 17]. The Lanczos algorithm [19] is an iterative procedure that is applied to an $N \times N$ matrix $\mathbf{A}$ and two right and left starting vectors $\mathbf{R}$ and $\mathbf{L}$ of length N. After n iterations, the Lanczos process has generated an $n \times n$ matrix $\mathbf{T}_n$ that represents the oblique projection of $\mathbf{A}$ onto the nth right Krylov subspace (generated by $\mathbf{A}$ and $\mathbf{R}$) and orthogonally to the nth left Krylov subspace (generated by $\mathbf{A}^{\mathrm{T}}$ and $\mathbf{L}$). The Lanczos-Padé connection then states that, for $r = 2n$, a solution of the minimal partial realization problem (for $m = p = 1$) is given by the nth Lanczos matrix $\mathbf{A}_n = \mathbf{T}_n$ together with suitably scaled first unit vectors $\mathbf{B}_n$ and $\mathbf{C}_n$. Using the Lanczos process to compute the solution of minimal partial realization problems in the case $m = p = 1$ was advocated in [9, 10, 14, 21].

In this paper, we present an extension of this Lanczos-based procedure for $m = p = 1$ to the case of general minimal partial realization problems (9) with *arbitrary* $m, p \geq 1$. It is based on a Lanczos-type process for multiple starting vectors, which was recently developed by Aliaga, Boley, Freund, and Hernández [1] and further refined in [13]. This Lanczos-type algorithm is an iterative procedure that is applied to an $N \times N$ matrix $\mathbf{A}$, a block of m right starting vectors $\mathbf{R}$, and a block of p left starting vectors $\mathbf{L}$. The difficulty here is that m and p are in general different, and in fact, the algorithm [1, 13] is the first Lanczos-type algorithm that can handle arbitrary numbers of right and left starting vectors, while all previous algorithms are restricted to $m = p$. After n iterations, this Lanczos-type process has generated an $n \times n$ matrix $\mathbf{T}_n$ that represents the oblique projection of A onto the nth right block Krylov subspace (generated by $\mathbf{A}$ and $\mathbf{R}$) and orthogonally to the nth left block Krylov subspace (generated by $\mathbf{A}^{\mathrm{T}}$ and $\mathbf{L}$). One key features of the algorithm is a built-in deflation procedure to detect and deflate linearly dependent and almost linearly dependent vectors in the underlying block Krylov subspaces. A second feature is a built-in *look-ahead* procedure to remedy the potential breakdowns that cannot be excluded in Lanczos-type algorithms.

In Section 2, we briefly review the appropriate concept of block Krylov subspaces. In Section 3, we state the key properties of the Lanczos-type process. In Section 4, we present a listing of a variant of the Lanczos-type algorithm without look-ahead. In Section 5, a connection to the minimal partial realization problem is formulated. In Section 6, we make some concluding remarks.

2 Block Krylov Subspaces

Let $\mathbf{A} \in \mathbb{C}^{N \times N}$, $\mathbf{R} \in \mathbb{C}^{N \times m}$, and $\mathbf{L} \in \mathbb{C}^{N \times p}$. We now define the *block Krylov subspaces* that are generated by these matrices.

We begin by introducing the *Krylov matrices*

$$\begin{aligned} \mathbf{K}^{(r)} &= \begin{bmatrix} \mathbf{R} & \mathbf{AR} & \cdots & \mathbf{A}^{N-1}\mathbf{R} \end{bmatrix} \\ \text{and } \mathbf{K}^{(l)} &= \begin{bmatrix} \mathbf{L} & \mathbf{A}^{\mathrm{T}}\mathbf{L} & \cdots & (\mathbf{A}^{\mathrm{T}})^{N-1}\mathbf{L} \end{bmatrix}. \end{aligned} \tag{10}$$

By scanning the columns of $\mathbf{K}^{(r)}$, respectively $\mathbf{K}^{(l)}$, from left to right and deleting all vectors that are linearly dependent on vectors to their left, we obtain the *deflated* Krylov matrices

$$\begin{aligned} \mathbf{K}^{(r)}_{\text{defl}} &= \begin{bmatrix} \mathbf{R}_1 & \mathbf{A}\mathbf{R}_2 & \cdots & \mathbf{A}^{j_{\max}-1}\mathbf{R}_{j_{\max}} \end{bmatrix} \\ \text{and } \mathbf{K}^{(l)}_{\text{defl}} &= \begin{bmatrix} \mathbf{L}_1 & \mathbf{A}^{\mathrm{T}}\mathbf{L}_2 & \cdots & (\mathbf{A}^{\mathrm{T}})^{k_{\max}-1}\mathbf{L}_{k_{\max}} \end{bmatrix}. \end{aligned} \tag{11}$$

This process of deleting linearly dependent vectors is called *exact deflation*. Of course, in an actual implementation in finite-precision arithmetic, one would also need to delete the vectors that are almost linearly dependent on vectors to their left; this process is called *inexact deflation*. For the moment, we assume that the matrices (11) are obtained from (10) using only exact deflation. By construction, in (11), the matrices $\mathbf{R}_j \in \mathbb{C}^{N \times m_j}$ and $\mathbf{L}_k \in \mathbb{C}^{N \times p_k}$ are submatrices of R and L, respectively. Moreover, $\mathbf{R}_j$ is a submatrix of $\mathbf{R}_{j-1}$, and $\mathbf{L}_j$ is a submatrix of $\mathbf{L}_{j-1}$, so that

$$\begin{aligned} & m \geq m_1 \geq m_2 \geq \cdots \geq m_{j_{\max}} \geq 1, \\ & p \geq p_1 \geq p_2 \geq \cdots \geq p_{k_{\max}} \geq 1. \end{aligned} \tag{12}$$

Note that

$$\operatorname{rank} \mathbf{K}^{(r)} = \sum_{j=1}^{j_{\max}} m_j \quad \text{and} \quad \operatorname{rank} \mathbf{K}^{(l)} = \sum_{k=1}^{k_{\max}} p_k.$$

We now set $N_0 = \min\{\operatorname{rank} \mathbf{K}^{(r)}, \operatorname{rank} \mathbf{K}^{(l)}\}$. Then, for each integer n with $\max\{m_1, p_1\} \leq n \leq N_0$, we define

$$\begin{aligned} \mathcal{K}_n(\mathbf{A}, \mathbf{R}) &= \operatorname{colspan}_n\left(\mathbf{K}^{(r)}_{\text{defl}}\right) \\ \text{and} \quad \mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L}) &= \operatorname{colspan}_n\left(\mathbf{K}^{(l)}_{\text{defl}}\right). \end{aligned} \tag{13}$$

Here, for any matrix $\mathbf{S} \in \mathbb{C}^{N \times N_0}$ of full column rank N_0 and all $n = 1, 2, \ldots, N_0$, we denote by $\operatorname{colspan}_n(\mathbf{S})$ the subspace of $\mathbb{C}^N$ spanned by the first n columns of $\mathbf{S}$. The matrix $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$, respectively $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$, in (13) is called the *nth right*, respectively *left*, *block Krylov subspace* (generated by $\mathbf{A}$ and $\mathbf{R}$, respectively $\mathbf{A}^{\mathrm{T}}$ and $\mathbf{L}$).

3 A Lanczos-Type Process

For the special case $m = p = 1$, i.e., $\mathbf{R}$, $\mathbf{L} \in \mathbb{C}^N$, the subspaces $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$ and $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$ reduce to the standard Krylov subspaces generated by the matrix $\mathbf{A}$ and the vectors $\mathbf{R}$ and $\mathbf{L}$. The classical Lanczos method [19] is a numerical procedure for generating two sequences of biorthogonal basis vectors for the Krylov subspaces $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$ and $\mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L})$. In [1], Aliaga, Boley, Freund, and Hernández presented a Lanczos-type algorithm for multiple starting vectors that extends the classical Lanczos process to the general block case m, $p \geq 1$.

Their algorithm can also be viewed as an extension of Ruhe's banded Lanczos procedure [20] for Hermitian matrices. A somewhat simpler variant of the algorithm [1] was first described in [11] and later on further refined in [13]. The main advantage of that variant over the one in [1] is that, at each iteration, the matrix-vector products with $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$ are computed for a pair of Lanczos vectors with identical indices. This "synchronization" eliminates the need for the so-called history indices in [1].

In this section, we describe the key properties of the version[11, 13] of the Lanczos-type algorithm for multiple starting vectors. The procedure generates two sequences of *right* and *left Lanczos vectors*,

$$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n, \ldots \quad \text{and} \quad \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n, \ldots, \tag{14}$$

such that, for each $n = 1, 2, \ldots, N_0$,

$$\begin{aligned} \operatorname{span}\{\, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \,\} &= \mathcal{K}_n(\mathbf{A}, \mathbf{R}), \\ \operatorname{span}\{\, \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n \,\} &= \mathcal{K}_n(\mathbf{A}^{\mathrm{T}}, \mathbf{L}), \end{aligned}$$

and the vectors (14) are biorthogonal:

$$\mathbf{w}_i^{\mathrm{T}} \mathbf{v}_l = 0 \quad \text{for all} \quad i \neq l = 1, 2 \ldots, n. \tag{15}$$

In general, it may not be possible to enforce the biorthonogonality condition (15) at every step n. In fact, just as it is the case for the classical Lanczos process, trying to enforce (15) may result in so-called breakdowns or near-breakdowns of the algorithm due to division by zero or a near-zero number. However, these problems can be remedied by incorporating *look-ahead* techniques into the Lanczos-type algorithm. The basic idea is to relax the vector-wise biorthogonality (15) to a cluster-wise biorthogonality between clusters of successive Lanczos vectors whenever a breakdown or near-breakdown would occur in the no-look-ahead algorithm. We remark that a breakdown at step n occurs if, and only if, the matrix

$$\mathbf{D}_n := \mathbf{V}_n^{\mathrm{T}} \mathbf{W}_n$$

is singular. Here,

$$\mathbf{V}_n := \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{bmatrix} \quad \text{and} \quad \mathbf{W}_n := \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_n \end{bmatrix} \tag{16}$$

are $N \times n$ matrices whose columns are the right and left Lanczos vectors (14), respectively. Note that, by (15), the matrix $\mathbf{D}_n$ is diagonal if no look-ahead steps occur. In the case of look-ahead, the matrix $\mathbf{D}_n$ is block diagonal with square blocks of the size of the look-ahead clusters.

The Lanczos vectors (14) can be generated by means of short recurrences. Using the notation (16) and assuming, for simplicity, exact deflation only, these recurrences can be summarized as follows:

$$\begin{aligned} \mathbf{A}\mathbf{V}_n &= \mathbf{V}_n \mathbf{T}_n + \begin{bmatrix} 0 \cdots 0 \; \hat{\mathbf{v}}_{n+1} \cdots \hat{\mathbf{v}}_{n+m_c} \end{bmatrix}, \\ \mathbf{A}^{\mathrm{T}}\mathbf{W}_n &= \mathbf{W}_n \tilde{\mathbf{T}}_n + \begin{bmatrix} 0 \cdots 0 \; \hat{\mathbf{w}}_{n+1} \cdots \hat{\mathbf{w}}_{n+p_c} \end{bmatrix}. \end{aligned} \tag{17}$$

Here, the matrices $\mathbf{T}_n$ and $\tilde{\mathbf{T}}_n$ contain the recurrence coefficients. In the case of no look-ahead, the matrices $\mathbf{T}_n$, respectively $\tilde{\mathbf{T}}_n$, are banded with lower and upper bandwidths m and p, respectively p and m, reflecting the short recurrences. Moreover, in (17), m_c and p_c are the *current* block sizes. Initially, $m_c = m$ and $p_c = p$, and then one sets $m_c := m_c - 1$, respectively $p_c := p_c - 1$, every time a right, respectively left, Krylov vector is deflated. Finally, the vectors $\hat{\mathbf{v}}_{n+1}, \cdots, \hat{\mathbf{v}}_{n+m_c}$ and $\hat{\mathbf{w}}_{n+1}, \cdots, \hat{\mathbf{w}}_{n+p_c}$ are so-called auxiliary vectors. They satisfy the partial biorthogonality conditions

$$\begin{aligned} \mathbf{w}_i^{\mathrm{T}} \hat{\mathbf{v}}_l &= 0 \quad \text{for all} \quad i \leq n, \quad n+1 \leq l \leq n+m_c, \\ \hat{\mathbf{w}}_l^{\mathrm{T}} \mathbf{v}_i &= 0 \quad \text{for all} \quad i \leq n, \quad n+1 \leq l \leq n+p_c, \end{aligned} \tag{18}$$

and they are turned into the final Lanczos vectors $\mathbf{v}_{n+1}, \cdots, \mathbf{v}_{n+m_c}$ and $\mathbf{w}_{n+1}, \cdots, \mathbf{w}_{n+p_c}$ in the course of the next steps of the Lanczos-type algorithm.

The first Lanczos vectors $\mathbf{v}_1, \cdots, \mathbf{v}_{m_1}$ and $\mathbf{w}_1, \cdots, \mathbf{w}_{p_1}$ are obtained by biorthogonalizing the columns of the right and left starting blocks $\mathbf{R}$ and $\mathbf{L}$, respectively. Using (16), the corresponding recurrences can be summarized as follows:

$$\mathbf{R} = \mathbf{V}_{m_1} \boldsymbol{\rho} \quad \text{and} \quad \mathbf{L} = \mathbf{W}_{p_1} \boldsymbol{\eta}, \tag{19}$$

where $\boldsymbol{\rho} \in \mathbb{C}^{m_1 \times m}$ and $\boldsymbol{\eta} \in \mathbb{C}^{p_1 \times p}$ are upper triangular matrices. Note that, by (12), $m \geq m_1$ and $p \geq p_1$.

4 The Algorithm Without Look-Ahead

In this section, we present an explicit statement of the Lanczos-type algorithm without look-ahead, but with deflation.

Algorithm 1 (Lanczos-type method with deflation, but without look-ahead.)
INPUT: A matrix $\mathbf{A} \in \mathbb{C}^{N \times N}$;

A block of m right starting vectors

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_m \end{bmatrix} \in \mathbb{C}^{N \times m};$$

A block of p left starting vectors

$$\mathbf{L} = \begin{bmatrix} \mathbf{l}_1 & \mathbf{l}_2 & \cdots & \mathbf{l}_p \end{bmatrix} \in \mathbb{C}^{N \times p};$$

A deflation tolerance `dtol`.
(Set `dtol`:=0 for exact deflation only.)

0) *For* $i = 1, 2, \ldots, m$, *set* $\hat{\mathbf{v}}_i = \mathbf{r}_i$.
For $i = 1, 2, \ldots, p$, *set* $\hat{\mathbf{w}}_i = \mathbf{l}_i$.
Set $m_{\mathrm{c}} = m$ *and* $p_{\mathrm{c}} = p$.
(m_{c} and p_{c} are the current right and left block sizes, respectively.)
Set $\mathcal{I}_{\mathbf{v}} = \emptyset$ *and* $\mathcal{I}_{\mathbf{w}} = \emptyset$.
($\mathcal{I}_{\mathbf{v}}$ and $\mathcal{I}_{\mathbf{w}}$ record indices of vectors that must be preserved due to inexact deflation. If $i \in \mathcal{I}_{\mathbf{v}}$, respectively $i \in \mathcal{I}_{\mathbf{w}}$, then the newly constructed $\mathbf{v}$, respectively $\mathbf{w}$, vectors need to be biorthogonalized against $\mathbf{w}_i$, respectively $\mathbf{v}_i$.)

For $n = 1, 2, \ldots,$ *do* (Build nth pair of Lanczos vectors $\mathbf{v}_n$ and $\mathbf{w}_n$.) :

1) (If necessary, deflate $\mathbf{v}_n$.) *If* $\|\hat{\mathbf{v}}_n\| > \mathtt{dtol}$, *then continue with step* 2).
Otherwise, deflate $\hat{\mathbf{v}}_n$ *by doing the following*:

1a) *If* $m_{\mathrm{c}} = 1$, *then stop.*
(There are no more right Krylov vectors.)

1b) *If* $\mu = n - m_{\mathrm{c}} > 0$ *and the deflated vector* $\hat{\mathbf{v}}_n$ *is nonzero, then set* $\mathcal{I}_{\mathbf{w}} = \mathcal{I}_{\mathbf{w}} \cup \{\mu\}$ *and save the vectors* $\mathbf{v}_\mu$ *and* $\mathbf{w}_\mu$.

1c) *For* $i = n, n+1, \ldots, n + m_{\mathrm{c}} - 2$, *set* $\hat{\mathbf{v}}_i = \hat{\mathbf{v}}_{i+1}$.
(The vector $\hat{\mathbf{v}}_n$ is deflated. The indices of the remaining right auxiliary vectors are reduced by one.)
Set $m_{\mathrm{c}} = m_{\mathrm{c}} - 1$.
(The current right block size is reduced by one.)

1d) *Repeat all of step* 1).

2) (If necessary, deflate $\mathbf{w}_n$.) *If* $\|\hat{\mathbf{w}}_n\| > \mathtt{dtol}$, *then continue with step* 2).
Otherwise, deflate $\hat{\mathbf{w}}_n$ *by doing the following*:

2a) *If* $p_{\mathrm{c}} = 1$, *then stop.*
(There are no more left Krylov vectors.)

2b) *If* $\phi = n - p_{\mathrm{c}} > 0$ *and the deflated vector* $\hat{\mathbf{w}}_n$ *is nonzero, then set* $\mathcal{I}_{\mathbf{v}} = \mathcal{I}_{\mathbf{v}} \cup \{\phi\}$ *and save the vectors* $\mathbf{v}_\phi$ *and* $\mathbf{w}_\phi$.

2c) *For* $i = n, n+1, \ldots, n + p_{\mathrm{c}} - 1$, *set* $\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i+1}$.
(The vector $\hat{\mathbf{w}}_n$ is deflated. The indices of the remaining left auxiliary vectors are reduced by one.)
Set $p_{\mathrm{c}} = p_{\mathrm{c}} - 1$.
(The current left block size is reduced by one.)

2d) *Repeat all of step* 2).

3) (Normalize $\hat{\mathbf{v}}_n$ and $\hat{\mathbf{w}}_n$ to obtain the Lanczos vectors $\mathbf{v}_n$ and $\mathbf{w}_n$.)
Set $\mu = n - m_{\mathrm{c}}$, $\phi = n - p_{\mathrm{c}}$,

$$\mathbf{v}_n = \frac{\hat{\mathbf{v}}_n}{t_{n,\mu}}, \quad \textit{and} \quad \mathbf{w}_n = \frac{\hat{\mathbf{w}}_n}{\tilde{t}_{n,\phi}},$$

where $t_{n,\mu}$ *and* $\tilde{t}_{n,\phi}$ *are suitable scaling factors, e.g.,*

$$t_{n,\mu} = \|\hat{\mathbf{v}}_n\| \quad \textit{and} \quad \tilde{t}_{n,\phi} = \|\hat{\mathbf{w}}_n\|.$$

4) (Compute δ_n and check for breakdown.)

4a) *Set*

$$\delta_n = \mathbf{w}_n^{\mathrm{T}} \mathbf{v}_n.$$

4b) *If* $\delta_n = 0$, *then stop.*

5) (Advance the right block Krylov subspace and obtain new vector $\hat{\mathbf{v}}_{n+m_c}$.)

5a) *Set*

$$\mathbf{v} = \mathbf{A}\,\mathbf{v}_n.$$

5b) (Biorthogonalize $\mathbf{v}$ against previous Lanczos vectors $\mathbf{w}_i$, $i < n$.)
Set $i_{\mathbf{v}} = \max\{1, n - p_c\}$ *and define the temporary index set*

$$\mathcal{I} = \mathcal{I}_{\mathbf{v}} \cup \{\, i_{\mathbf{v}}, i_{\mathbf{v}} + 1, \ldots, n-1 \,\}.$$

For all $i \in \mathcal{I}$ *(in ascending order), set*

$$\begin{aligned} t_{i,n} &= \begin{cases} \tilde{t}_{n,i}\,\dfrac{\delta_n}{\delta_i} & \text{if } i = n - p_c, \\ \dfrac{\mathbf{w}_i^{\mathrm{T}}\mathbf{v}}{\delta_i} & \text{otherwise,} \end{cases} \\ \mathbf{v} &= \mathbf{v} - \mathbf{v}_i\, t_{i,n}. \end{aligned}$$

Set $\hat{\mathbf{v}}_{n+m_c} = \mathbf{v}$.

6) (Advance the left block Krylov subspace and obtain new vector $\hat{\mathbf{w}}_{n+p_c}$.)

6a) *Set*

$$\mathbf{w} = \mathbf{A}^{\mathrm{T}}\mathbf{w}_n.$$

6b) (Biorthogonalize $\mathbf{w}$ against previous Lanczos vectors $\mathbf{v}_i$, $i < n$.)
Set $i_{\mathbf{w}} = \max\{1, n - m_c\}$ *and define the temporary index set*

$$\mathcal{I} = \mathcal{I}_{\mathbf{w}} \cup \{\, i_{\mathbf{w}}, i_{\mathbf{w}} + 1, \ldots, n-1 \,\}.$$

For all $i \in \mathcal{I}$ *(in ascending order), set*

$$\begin{aligned} \tilde{t}_{i,n} &= \begin{cases} t_{n,i}\,\dfrac{\delta_n}{\delta_i} & \text{if } i = n - m_c, \\ \dfrac{\mathbf{w}^{\mathrm{T}}\mathbf{v}_i}{\delta_i} & \text{otherwise,} \end{cases} \\ \mathbf{w} &= \mathbf{w} - \mathbf{w}_i\, \tilde{t}_{i,n}. \end{aligned}$$

Set $\hat{\mathbf{w}}_{n+p_c} = \mathbf{w}$.

7) (Update the existing auxiliary vectors.)

7a) (Biorthogonalize the vectors $\hat{\mathbf{v}}_{n+1}, \ldots, \hat{\mathbf{v}}_{n+m_c}$ against $\mathbf{w}_n$.)
For $i = n - m_c + 1, n - m_c + 2, \ldots, n$, *set*

$$\begin{aligned} t_{n,i} &= \begin{cases} \dfrac{\mathbf{w}_n^{\mathrm{T}}\hat{\mathbf{v}}_{m_c+i}}{\delta_n} & \text{if } i \le 0 \text{ or } i = n, \\ \tilde{t}_{i,n}\,\dfrac{\delta_i}{\delta_n} & \text{otherwise,} \end{cases} \\ \hat{\mathbf{v}}_{m_c+i} &= \hat{\mathbf{v}}_{m_c+i} - \mathbf{v}_n\, t_{n,i}. \end{aligned}$$

7b) (Biorthogonalize the vectors $\hat{\mathbf{w}}_{n+1}, \ldots, \hat{\mathbf{w}}_{n+p_c}$ against $\mathbf{v}_n$.)
For $i = n - p_c + 1, n - p_c + 2, \ldots, n$, *set*

$$\begin{aligned} \tilde{t}_{n,i} &= \begin{cases} \dfrac{\hat{\mathbf{w}}_{p_c+i}^{\mathrm{T}} \mathbf{v}_n}{\delta_n} & \text{if } i \le 0, \\ t_{i,n} \dfrac{\delta_i}{\delta_n} & \text{otherwise,} \end{cases} \\ \hat{\mathbf{w}}_{p_c+i} &= \hat{\mathbf{w}}_{p_c+i} - \mathbf{w}_n \, \tilde{t}_{n,i}. \end{aligned}$$

8) (In the initial iterations, set up $\boldsymbol{\rho}$ and $\boldsymbol{\eta}$.)
If $n \le m_c$, *set*

$$\rho_{n,i} = t_{n,i-m} \quad \text{for all} \quad i = n - m_c + m, \ldots, m.$$

If $n \le p_c$, *set*

$$\eta_{n,i} = \tilde{t}_{n,i-p} \quad \text{for all} \quad i = n - p_c + p, \ldots, p.$$

5 A Connection to the Realization Problem

We continue to use the notation introduced in Sections 2 and 3. In particular, $\boldsymbol{\rho}$, $\boldsymbol{\eta}$, $\mathbf{D}_n$, and $\mathbf{T}_n$ are the quantities generated by the Lanczos-type algorithm for multiple starting vectors applied to $\mathbf{A}$, $\mathbf{R}$, and $\mathbf{L}$. Moreover, we assume that the Lanczos-type algorithm is run with eaxct deflation only. Recall from Section 2 that the integers m_j and p_k are defined via the matrices $\mathbf{R}_j \in \mathbb{C}^{N \times m_j}$ and $\mathbf{L}_k \in \mathbb{C}^{N \times p_k}$, where $\mathbf{R}_j$ and $\mathbf{L}_k$ are the blocks in the deflated Krylov matrices (11). For each integer n with $\max\{m_1, p_1\} \le n \le N_0$, we define $j = j(n)$ and $k = k(n)$ as the maximal integers such that

$$\begin{aligned} m_1 + m_2 + \cdots + m_j &\le n \\ \text{and} \quad p_1 + p_2 + \cdots + p_k &\le n. \end{aligned} \tag{20}$$

Finally, we denote by $\mathbf{0}_{i \times l}$ the $i \times l$ zero matrix.

Our result about the connection between the Lanczos-type process and the partial realization problem can now be stated as follows.

Theorem 2 *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* (20). *Let* $\boldsymbol{\rho}$, $\boldsymbol{\eta}$, $\mathbf{D}_n$, *and* $\mathbf{T}_n$ *be the matrices generated by* n *steps of the Lanczos-type process applied to* $\mathbf{A}$, $\mathbf{R}$, *and* $\mathbf{L}$. *Then, the triplet*

$$\mathbf{T}_n, \quad \boldsymbol{\rho}_n := \begin{bmatrix} \boldsymbol{\rho} \\ \mathbf{0}_{(n-m_1)\times m} \end{bmatrix}, \quad \boldsymbol{\eta}_n := \mathbf{D}_n^{\mathrm{T}} \begin{bmatrix} \boldsymbol{\eta} \\ 0_{(n-p_1)\times p} \end{bmatrix} \tag{21}$$

is a partial realization of the matrix sequence

$$\mathbf{L}^{\mathrm{T}}\mathbf{R}, \mathbf{L}^{\mathrm{T}}\mathbf{A}\mathbf{R}, \cdots, \mathbf{L}^{\mathrm{T}}\mathbf{A}^{r-1}\mathbf{R}, \quad r = j(n) + k(n),$$

i.e.,

$$\boldsymbol{\eta}_n^{\mathrm{T}} \mathbf{T}_n^i \boldsymbol{\rho}_n = \mathbf{L}^{\mathrm{T}} \mathbf{A}^i \mathbf{R}, \quad \text{for all} \quad i = 0, 1, \ldots, r-1. \tag{22}$$

Moreover, if in (20) *at least one "$\le$" is an "$=$" and if* $\mathbf{D}_n$ *is nonsingular, then the partial realization* (21) *is minimal.*

Here, we only sketch the proof of Thereom 1; details will be presented elsewhere. Using the properties (17)–(19) as well as the band structure of $\mathbf{T}_n$ and $\tilde{\mathbf{T}}_n$, it is relatively straightforward to verify the realization property (22). In remains to show that, under the above additional assumptions on (20) and $\mathbf{D}_n$, the partial realization (21) is minimal. This can be done by using the criterion [15, Theorem 0.1].

6 Concluding Remarks

We described a Lanczos-type procedure for constructing partial realizations. Under some additional assumptions, we showed that this partial realization is minimal. For the case that these assumptions are not satisfied, the partial realization generated by the Lanczos-type algorithm can be furher reduced to a minimal partial realization by a suitable post-processing procedure. This will be described elsewhere.

The minimal partial realization is intimately related to matrix Padé approximation; see, e.g., [4, 23]. The construction of matrix Padé approximants via the Lanczos-type algorithm [1, 13] for multiple starting vectors is described in [11, 12, 13].

References

[1] 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 Labs., Murray Hill, NJ, Sep. 1996. (Available from `http://cm.bell-labs.com/cs/doc/96`)

[2] B.M. Anderson, F.M. Brasch, Jr., and P.V. Lopresti, "The sequential construction of minimal partial realizations from finite input-output data," *SIAM J. Control*, vol. 13, pp. 552–571, 1975.

[3] A.C. Antoulas, "On minimal realizations: A polynomial approach," *Systems Control Lett.*, vol. 14, pp. 319–324, 1990.

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

[5] Y. Bistritz, "Nested basis of invariants for minimal realizations of finite matrix sequences," *SIAM J. Control Optim.*, vol. 21, pp. 804–821, 1983.

[6] O.H. Bosgra, "On parametrizations for the minimal realization problem," *Systems Control Lett.*, vol. 3, pp. 181–187, 1983.

[7] O.H. Bosgra and A.J.J. Van der Weiden, "Input-out invariants for linear multivariable systems," *IEEE Trans. Automat. Control*, vol. AC-25, pp. 20–36, 1980.

[8] B.W. Dickinson, M. Morf, and T. Kailath, "A minimal realization algorithm for matrix sequences," *IEEE Trans. Automat. Control*, vol. AC-19, pp. 31–38, 1974.

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

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

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

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

[13] R.W. Freund, "An extension of the Lanczos-Padé connection to the matrix case," *Numerical Analysis Manuscript*, Bell Labs., Murray Hill, NJ, in preparation.

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

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

[16] W.B. Gragg, "Matrix interpretations and applications of the continued fraction algorithm," *Rocky Mountain J. Math.*, vol. 4, pp. 213–225, 1974.

[17] W.B. Gragg and A. Lindquist, "On the partial realization problem," *Linear Algebra Appl.*, vol. 50, pp. 277–319, 1983.

[18] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, 1980.

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

[20] A. Ruhe, "Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices," *Math. Comp.*, vol. 33, pp. 680–687, 1979.

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

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

[23] M. Van Barel and A. Bultheel, "A canonical matrix continued fraction solution of the minimal (partial) realization problem," *Linear Algebra Appl.*, vol. 122/123/124, pp. 973–1002, 1989.