Rational approximations of pre-filtered transfer functions via the Lanczos algorithm

Kyle Gallivan [a] and Paul Van Dooren [b]

[a] *Department of Computer Science, Florida State University, Tallahassee, FL-32306 USA*
E-mail: gallivan@cs.fsu.edu
[b] *CESAME, Université Catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium*
E-mail: vdooren@csam.ucl.ac.be

Given a single-input single-output system $\{A, b, c\}$ with strictly proper transfer function $g(s)$, we derive a Lanczos-based method to construct a tridiagonal state-space model $\{\hat{A}, \hat{b}, \hat{c}\}$ approximating the "pre-filtered" transfer function $f(s)g(s)$, where $f(s)$ is given in factored form $f(s) \doteq \prod_{i=1}^{\ell}(s - z_i)/\prod_{i=1}^{\ell}(s - p_i)$. We also show how to apply this idea to the Arnoldi process and mention a few other extensions.

1. Introduction

In this short paper we develop Lanczos-based methods to fit the first $2k$ moments of a transfer function

$$h(s) = h_1 s^{-1} + h_2 s^{-2} + \cdots + h_{2k} s^{-2k} + \cdots \tag{1}$$

via a Lanczos-based method operating directly on the state-space model $\{A, b, c\}$ of the transfer function

$$h(s) = c(sI_n - A)^{-1}b. \tag{2}$$

Notice that we use expansions around $s = \infty$ in (2) and that $h_0 = 0$ by assumption. The relation between (1) and (2) is easily seen to be

$$h_i = cA^{i-1}b, \tag{3}$$

and the aim is to construct a *reduced order* model $\{\hat{A}, \hat{b}, \hat{c}\}$ of order k and with transfer function

$$\hat{h}(s) = \hat{c}(sI_k - \hat{A})^{-1}\hat{b}, \tag{4}$$

so that

$$h(s) - \hat{h}(s) = c_{2k+1}s^{-(2k+1)} + \mathrm{O}\!\left(s^{-(2k+2)}\right). \tag{5}$$

In other words, $h(s)$ and $\hat{h}(s)$ match in their *first $2k$ moments* of their expansion around $s = \infty$, and $\hat{h}(s)$ is constrained to be of *degree k*.

This basic problem has been addressed in several papers already [5,9] but here we consider the problem where $h(s)$ is given as a *product* of two functions:

$$h(s) = f(s)g(s), \tag{6}$$

where $g(\infty) = 0$ and $f(\infty) = f_0 < \infty$. This implies that there exist expansions

$$g(s) = g_1 s^{-1} + g_2 s^{-2} + \cdots + g_{2k}s^{-2k} + \cdots, \tag{7}$$
$$f(s) = f_0 + f_1 s^{-1} + f_2 s^{-2} + \cdots. \tag{8}$$

Moreover, we *assume* that we already have a kth order model $\{\tilde{A}, \tilde{b}, \tilde{c}\}$ fitting the first $2k$ moments of $g(s)$:

$$g(s) - \tilde{c}\!\left(sI_k - \tilde{A}\right)^{-1}\tilde{b} = \mathrm{O}\!\left(s^{-(2k+1)}\right), \tag{9}$$

and also that $f(s)$ is a transfer function of which we know the poles and zeros:

$$f(s) = \frac{(s - z_1)\cdots(s - z_\ell)}{(s - p_1)\cdots(s - p_\ell)}. \tag{10}$$

The motivation of this problem as follows: suppose we have a function $g(s)$ that is well approximated by a kth order system $\hat{g}(s) = \tilde{c}(sI_k - \tilde{A})^{-1}\tilde{b}$. We then want to "pre-filter" $g(s)$ by $f(s)$ – whose poles and zeros are known – and derive from this a kth order approximation of the compound system $f(s)g(s)$:

$$f(s)g(s) - \hat{c}\!\left(sI_k - \hat{A}\right)^{-1}\hat{b} = \mathrm{O}\!\left(s^{-(2k+1)}\right). \tag{11}$$

The role of the filter $f(s)$ is typically to emphasize certain frequencies or to stabilize the system.

2. Background

Here we develop some basic facts about realizations of transfer functions and their product. Any linear time-invariant system

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

can be described equivalently in the Laplace domain:

$$\begin{cases} sx(s) = Ax(s) + Bu(s) \\ y(s) = Cx(s) + Du(s). \end{cases}$$

Eliminating $x(s)$ from this then yields the associated transfer function

$$R(s) \doteq C(sI - A)^{-1}B + D,$$

which describes the relation between inputs and outputs in the Laplace domain. Essentially the same result is obtained for discrete-time linear time-invariant systems when using the z-transform. For simplicity we restrict ourselves to the continuous-time case, but everything below also holds for discrete-time systems. We also assume – for the sake of simplicity – that the models $\{A, B, C, D\}$ are *real*, but point out that everything in this paper trivially extends to the complex case. As described in the introduction, we are interested in approximating this transfer function by another one of lower degree and such that both match in their first $2k$ terms (called *moments*) of an expansion around $s = \infty$. Since pre-filtering amounts to pre-multiplying a transfer function, the following lemma will be useful.

Lemma 1 (See [11] for a proof). Let $\{A_i, B_i, C_i, D_i\}$ for $i = 1, 2$ be realizations of transfer functions $R_i(s)$, then a realization for the product $R_1(s)R_2(s)$ is specified by $\{A, B, C, D\}$, where

$$R_1(s)R_2(s) \Leftrightarrow \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|c} A_1 & B_1C_2 & B_1D_2 \\ 0 & A_2 & B_2 \\ \hline C_1 & D_1C_2 & D_1D_2 \end{array}\right]$$

$$= \left[\begin{array}{cc|c} A_1 & & B_1 \\ & I_{n_2} & \\ \hline C_1 & & D_1 \end{array}\right] \left[\begin{array}{cc|c} I_{n_1} & & \\ & A_2 & B_2 \\ \hline & C_2 & D_2 \end{array}\right]. \qquad (12)$$

The realization quadruple can thus be obtained by a mere product of "expanded" realizations and we can expect sparsity to be preserved to a large extent, especially in the case that B_i and C_i are vectors and D_i scalars. As an example, let us take the single-input single-output (SISO) case and let us choose $\ell = 1$, i.e., $f(s) = (s - z)/(s - p)$. A realization for $f(s)$ is easily seen to be

$$f(s) \Leftrightarrow \left[\begin{array}{c|c} A_1 & b_1 \\ \hline c_1 & d_1 \end{array}\right] = \left[\begin{array}{c|c} p & 1 \\ \hline p - z & 1 \end{array}\right]. \qquad (13)$$

Assume $g(s)$ also scalar and $g(\infty) = 0$, then

$$g(s) \Leftrightarrow \left[\begin{array}{c|c} A_2 & b_2 \\ \hline c_2 & 0 \end{array}\right] \qquad (14)$$

and the following realization for $f(s)g(s)$ is obtained from lemma 1:

$$f(s)g(s) \Leftrightarrow \left[\begin{array}{c|c} A & b \\ \hline c & 0 \end{array}\right] = \left[\begin{array}{cc|c} p & c_2 & 0 \\ 0 & A_2 & b_2 \\ \hline p - z & c_2 & 0 \end{array}\right], \qquad (15)$$

which clearly preserves sparsity. We show later on that this simple case of a first degree filter $f(s)$ is not restrictive since we can repeat the same argument to add on additional first order pre-filters and build recursively one of arbitrary degree ℓ.

How can this be used in the construction of $\hat{h}(s)$ when $\hat{g}(s)$ is given? It turns out that one can start from $\hat{h}(s) = f(s)\hat{g}(s)$ rather than from $h(s) = f(s)(s)$ for the following reason. Let

$$g(s) - \hat{g}(s) = \mathrm{O}\big(s^{-(2k+1)}\big), \tag{16}$$

then it follows that

$$f(s)g(s) - f(s)\hat{g}(s) = \mathrm{O}\big(s^{-(2k+1)}\big), \tag{17}$$

since $f(s)$ is regular at $s = \infty$. But this is the same as the desired result

$$h(s) - \hat{h}(s) = \mathrm{O}\big(s^{-(2k+1)}\big). \tag{18}$$

So it is sufficient to find a kth order model matching the first $2k$ moments of $f(s)\hat{g}(s)$, which is a $(k+\ell)$th order function. So we have to run the Lanczos–Markov algorithm on a realization of $f(s)\hat{g}(s)$ which is a reasonably low order system. This process is briefly explained in the next section.

3.   A Lanczos-based algorithm

We assume here that we are starting from a sparse system $\{A_g, b_g, c_g\}$ realizing the transfer function $g(s)$:

$$g(s) = c_g(sI_n - A_g)^{-1}b_g, \quad n = \text{degree } g(s) \gg k. \tag{19}$$

A kth order model $\{\tilde{A}, \tilde{b}, \tilde{c}\}$ such that

$$\tilde{c}\tilde{A}^{i-1}\tilde{b} = c_g A_g^{i-1} b_g, \quad i = 1, \ldots, 2k,$$

is obtained from the Lanczos algorithm [5,8] which constructs $n \times k$ matrices V_k and W_k such that

$$\begin{aligned}
\mathrm{Im}V_k &= \mathrm{Im}\big[b_g, A_g b_g, \ldots, A_g^{k-1} b_g\big], \\
\mathrm{Im}W_k &= \mathrm{Im}\big[c_g^\mathrm{T}, A_g^\mathrm{T} c_g^\mathrm{T}, \ldots, A_g^{\mathrm{T}k-1} c_g^\mathrm{T}\big], \\
W_k^\mathrm{T} V_k &= I_k,
\end{aligned}$$

giving

$$\hat{g}(s) \Leftrightarrow \left[\begin{array}{c|c} \tilde{A} & \tilde{b} \\ \hline \tilde{c} & 0 \end{array}\right] = \left[\begin{array}{c|c} W_k^\mathrm{T} A_g V_k & W_k^\mathrm{T} b_g \\ \hline c_g V_k & 0 \end{array}\right] = \left[\begin{array}{cccc|c} \alpha_1 & \gamma_2 & & & \beta_1 \\ \beta_2 & \ddots & \ddots & & 0 \\ & \ddots & \ddots & \gamma_k & \vdots \\ & & \beta_k & \alpha_k & 0 \\ \hline \gamma_1 & 0 & \ldots & 0 & 0 \end{array}\right]. \tag{20}$$

All parameters α_i, β_i and γ_i here are scalar if the Lanczos tridiagonalization did not encounter any breakdowns. If either β_i or γ_i is zero, then we found an uncontrollable or unobservable mode, which is easy to deflate [4,9]. Other breakdowns lead to a modification of the algorithm (using so-called look-ahead steps) and result in parameters α_i, β_i and γ_i that are *blocks* of compatible size [5,7,9], see also section 6. For the sake of simplicity, we consider the scalar and tridiagonal case first and will come back to the more general case later.

If we now construct the realization for the product $f(s)\hat{g}(s)$ and assume that $f(s)$ is of degree one,

$$f(s) \Leftrightarrow \left[\begin{array}{c|c} p & 1 \\ \hline p-z & 1 \end{array} \right],$$

then we have

$$f(s)\hat{g}(s) \Leftrightarrow \left[\begin{array}{c|ccccc|c} p & \gamma_1 & 0 & \ldots & 0 & & 0 \\ \hline 0 & \alpha_1 & \gamma_2 & & & & \beta_1 \\ 0 & \beta_2 & \ddots & \ddots & & & 0 \\ \vdots & & \ddots & \ddots & \gamma_k & & \vdots \\ 0 & & & \beta_k & \alpha_k & & 0 \\ \hline p-z & \gamma_1 & 0 & \ldots & 0 & & 0 \end{array} \right] \triangleq \left[\begin{array}{c|c} A_+ & b_+ \\ \hline c_+ & 0 \end{array} \right]. \tag{21}$$

It is easy to see that this is *almost* in the required tridiagonal form (20) resulting from the Lanczos algorithm. It suffices to find an updating state-space transformation T such that

$$\left[\begin{array}{c|c} T^{-1}A_+T & T^{-1}b_+ \\ \hline c_+T & 0 \end{array} \right] = \left[\begin{array}{cccc|c} \hat{\alpha}_1 & \hat{\gamma}_2 & & 0 & \hat{\beta}_1 \\ \hat{\beta}_2 & \ddots & \ddots & & 0 \\ & \ddots & \ddots & \hat{\gamma}_{k+1} & \vdots \\ 0 & & \hat{\beta}_{k+1} & \hat{\alpha}_{k+1} & 0 \\ \hline \hat{\gamma}_1 & 0 & \ldots & 0 & 0 \end{array} \right], \tag{22}$$

and this can be achieved by a sequence of "bulge chasing" LR steps that propagate the superfluous elements $\hat{\beta}_1$ and $\hat{\gamma}_1$ in (21) to the bottom, resulting finally in (22). Of course, it is possible that an LR step breaks down, in which case (22) will require a look-ahead step [9], but for the sake of simplicity we leave this discussion for later.

If one chooses

$$M_1 = \left[\begin{array}{cc} 0 & \gamma_1 \\ 1 & p-z \end{array} \right], \qquad M_1^{-1} = \left[\begin{array}{cc} \dfrac{p-z}{\gamma_1} & 1 \\ \dfrac{1}{\gamma_1} & 0 \end{array} \right],$$

then, clearly,

$$M_1^{-1} \begin{bmatrix} 0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \beta_1 \\ 0 \end{bmatrix} \doteq \begin{bmatrix} \hat{\beta}_1 \\ 0 \end{bmatrix}, \qquad M_1^{\mathrm{T}} \begin{bmatrix} (p-z) \\ \gamma_1 \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ 0 \end{bmatrix} \doteq \begin{bmatrix} \hat{\gamma}_1 \\ 0 \end{bmatrix}.$$

Applying the state-space transformation

$$T_1 = \mathrm{diag}\{M_1, I_{k-1}\}$$

then yields

$$\begin{bmatrix}
\hat{\alpha}_1 & x_2 & y_2 & & & & \hat{\beta}_1 \\
u_2 & z_2 & t_2 & 0 & & & 0 \\
v_2 & w_2 & \alpha_2 & \gamma_3 & & & \vdots \\
 & 0 & \beta_3 & \alpha_3 & \ddots & & \vdots \\
 & & & \ddots & \ddots & \gamma_k & 0 \\
 & & & & \beta_k & \alpha_k & 0 \\
\hat{\gamma}_1 & 0 & \ldots & \ldots & 0 & 0 & 0
\end{bmatrix}. \tag{23}$$

In the next step we define a transformation M_2 such that

$$M_2^{-1} \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} \hat{\beta}_2 \\ 0 \end{bmatrix}, \qquad M_2^{\mathrm{T}} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \hat{\gamma}_2 \\ 0 \end{bmatrix},$$

which also defines $\hat{\beta}_2$ and $\hat{\gamma}_2$. This transformation (almost) always exists since we only impose two vector conditions. It is easy to check that it always exists if and only if $u_2 x_2 + v_2 y_2 \neq 0$ since this also equals the product $\hat{\beta}_2 \hat{\gamma}_2$. Applying the state-space transformation

$$T_2 = \mathrm{diag}\{1, M_2, I_{k-2}\}$$

then moves the 3×3 "bulge" one step downwards. We illustrate this by showing the leading 4×4 submatrix of (23) only:

$$\begin{bmatrix} 1 & & \\ & M_2^{-1} & \\ & & 1 \end{bmatrix} \begin{bmatrix} \hat{\alpha}_1 & x_2 & y_2 & \\ u_2 & z_2 & t_2 & 0 \\ v_2 & w_2 & \alpha_2 & \gamma_3 \\ & 0 & \beta_3 & \alpha_3 \end{bmatrix} \begin{bmatrix} 1 & & \\ & M_2 & \\ & & 1 \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_1 & \hat{\gamma}_2 & 0 & \\ \hat{\beta}_2 & \hat{\alpha}_2 & x_3 & y_3 \\ 0 & u_3 & z_3 & t_3 \\ & v_3 & w_3 & \alpha_3 \end{bmatrix},$$

which also defines $x_3, y_3, z_3, t_3, u_3, v_3, w_3$ to be used in later steps. This is continued until the "bulge" disappears in the bottom right corner. After that, one only needs to keep the leading $k \times k$ subsystem.

The operation count for this procedure is 40 flops per 2×2 transformation, resulting in a total of $40k$ flops since there are k transformations. If one wants to update the spaces V_k and W_k as well, then the operation count increases by $12nk$

flops. For each additional degree section $(s - z_i)/(s - p_i)$ one has to repeat the same process and, hence, the total amount is

$$
\begin{cases}
40kp & \text{flops to obtain } \{\hat{A}, \hat{b}, \hat{c}\}, \\
12nkp & \text{flops to obtain } V_k, W_k.
\end{cases}
$$

4. Continuing the Lanczos process

The process described in the previous section only explains how to obtain a kth order model $\hat{h}(s)$ matching the first $2k$ moments of $h(s)$, starting from a similar model $\hat{g}(s)$ for $g(s)$. But what to do if one wants a model of degree higher than k, matching more than the first $2k$ moments? From the discussion of the last section it is clear that it is no longer sufficient to start from $\hat{g}(s)$. It is precisely here that the updating of V_k and W_k enters the picture. Assume we start from the standard Lanczos recurrence for A_g, b_g, c_g at step k:

$$
\begin{cases}
A_g V_k = V_k \tilde{A} + q_k e_k^{\mathrm{T}}, & c_g = [\gamma_1\, 0\, \ldots\, 0] W_k^{\mathrm{T}}, \quad W_k^{\mathrm{T}} V_k = I_k, \\
W_k^{\mathrm{T}} A_g = \tilde{A} W_k^{\mathrm{T}} + e_k r_k^{\mathrm{T}}, & b_g^{\mathrm{T}} = [\beta_1\, 0\, \ldots\, 0] V_k^{\mathrm{T}},
\end{cases}
\tag{24}
$$

then this completely specifies the reduced model (20). From this, one also easily derives the following updated equation:

$$
\begin{bmatrix} p & c_g \\ \hline 0 & \\ \vdots & A_g \\ 0 & \end{bmatrix}
\begin{bmatrix} 1 & 0 \ldots 0 \\ \hline 0 & \\ \vdots & V_k \\ 0 & \end{bmatrix}
=
\begin{bmatrix} 1 & 0 \ldots 0 \\ \hline 0 & \\ \vdots & V_k \\ 0 & \end{bmatrix}
\begin{bmatrix} p & \gamma_1\, 0 \ldots 0 \\ \hline 0 & \\ \vdots & \tilde{A} \\ 0 & \end{bmatrix}
+
\begin{bmatrix} 0 \\ \\ q_k \\ \\ \end{bmatrix}
e_{k+1}^{\mathrm{T}},
$$

$$
\begin{bmatrix} 1 & 0 \ldots 0 \\ \hline 0 & \\ \vdots & W_k^{\mathrm{T}} \\ 0 & \end{bmatrix}
\begin{bmatrix} p & c_g \\ \hline 0 & \\ \vdots & A_g \\ 0 & \end{bmatrix}
=
\begin{bmatrix} p & \gamma_1\, 0 \ldots 0 \\ \hline 0 & \\ \vdots & \tilde{A} \\ 0 & \end{bmatrix}
\begin{bmatrix} 1 & 0 \ldots 0 \\ \hline 0 & \\ \vdots & W_k^{\mathrm{T}} \\ 0 & \end{bmatrix}
+
e_{k+1}
\begin{bmatrix} 0 \\ \\ r_k \\ \\ \end{bmatrix}^{\mathrm{T}},
$$

$$
\begin{bmatrix} p - z \\ \hline \\ c_g^{\mathrm{T}} \\ \\ \end{bmatrix}
=
\begin{bmatrix} 1 & 0 \ldots 0 \\ \hline 0 & \\ \vdots & W_k \\ 0 & \end{bmatrix}
\begin{bmatrix} p - z \\ \hline \gamma_1 \\ 0 \\ 0 \end{bmatrix},
$$

$$
\begin{bmatrix} 0 \\ \hline \\ b_g \\ \\ \end{bmatrix}
=
\begin{bmatrix} 1 & 0 \ldots 0 \\ \hline 0 & \\ \vdots & V_k \\ 0 & \end{bmatrix}
\begin{bmatrix} 0 \\ \hline \beta_1 \\ 0 \\ 0 \end{bmatrix}.
$$

When one updates the model as in the previous section it is obvious that the embedded matrices $\mathrm{diag}\{1, V_k\}$ and $\mathrm{diag}\{1, W_k\}$ have to be updated at the same time and that the 2×2 transformations have to be applied to these matrices as well. Notice that in the very last step of this update (step k) the *residual vectors* q_k and r_k^T also enter the picture and that one will obtain something of the following type:

$$
\begin{bmatrix} p & c_g \\ \hline 0 & \\ \vdots & A_g \\ 0 & \end{bmatrix}
\begin{bmatrix} \widehat{V}_k & \begin{matrix} * \\ \vdots \\ * \end{matrix} \end{bmatrix}
=
\begin{bmatrix} \widehat{V}_k & \begin{matrix} * \\ \vdots \\ * \end{matrix} \end{bmatrix}
\begin{bmatrix} \hat{\alpha}_1 & \hat{\gamma}_2 & & & 0 \\ \hat{\beta}_2 & \ddots & \ddots & & \vdots \\ & \ddots & \ddots & \hat{\gamma}_k & 0 \\ & & \hat{\beta}_k & \hat{\alpha}_k & * \\ \hline 0 & \dots & 0 & * & * \end{bmatrix}
$$

$$
+ \, \hat{q}_k \big(\gamma e_k^{\mathrm{T}} + \delta e_{k+1}^{\mathrm{T}} \big),
$$

$$
\begin{bmatrix} \widehat{W}_k^{\mathrm{T}} \\ \hline * \, \dots \, * \end{bmatrix}
\begin{bmatrix} p & c_g \\ \hline 0 & \\ \vdots & A_g \\ 0 & \end{bmatrix}
=
\begin{bmatrix} \hat{\alpha}_1 & \hat{\gamma}_2 & & & 0 \\ \hat{\beta}_2 & \ddots & \ddots & & \vdots \\ & \ddots & \ddots & \hat{\gamma}_k & 0 \\ & & \hat{\beta}_k & \hat{\alpha}_k & * \\ \hline 0 & \dots & 0 & * & * \end{bmatrix}
\begin{bmatrix} \widehat{W}_k^{\mathrm{T}} \\ \hline * \, \dots \, * \end{bmatrix}
$$

$$
+ \, (\zeta e_k + \eta e_{k+1}) \hat{r}_{k+1}^{\mathrm{T}}.
$$

(25)

If we now delete the last column and row of these equations, we have again the setup of stage k of a Lanczos process, as described in (24). From thereon, one can continue the process with spaces and vectors that are now in $\mathbb{R}^{n+1}$. The updated "residuals" $\hat{r}_k$ and $\hat{q}_k$ are directly obtained from (25).

5. Relation to implicit restarts

Starting from a particular Krylov subspace

$$
\mathcal{K}_k(A, b) \doteq \mathrm{Im} \left[b, Ab, \dots, A^{k-1}b \right],
$$

spanned by the columns of a matrix V_k (i.e., $\mathrm{Im}\, V_i = \mathcal{K}_i(A, b)$, $i = 1, \dots, k$), it is often useful to derive a modified space $\mathrm{Im}\, V_{f,k} = \mathcal{K}_k(A, b_f)$, where $b_f = f(A)b$ and $f(s) = \prod_{i=1}^{\ell}(s - z_i)$ is an appropriately chosen polynomial. In [10] this is used to pre-filter the basis

$$
\mathrm{Im}\, V_{f,k} = \mathcal{K}_k(A, b_f) = f(A)\mathcal{K}_k(A, b)
$$

so that the eigenvalues of the projected matrix $V_{f,k}^{\mathrm{T}} A V_{f,k}$ tend faster to the spectrum of A in a specified region. It is also shown in that paper that passing from V_k to $V_{f,k}$ can be done *implicitly*, via the application of QR-steps on the Hessenberg matrix $V_k^{\mathrm{T}} A V_k$, rather than building the space *explicitly* from the pair $\{A, b_f\}$. The same idea was

subsequently applied to the Lanczos tridiagonalization in [6], where it was shown that HR-steps applied to the tridiagonal matrix $W_k^T A V_k^T$ implicitly pre-filter the Krylov spaces $\operatorname{Im} V_k = \mathcal{K}_k(A, b)$ and $\operatorname{Im} W_k = \mathcal{K}_k(A^T, c^T)$ to yield new spaces

$$\operatorname{Im} V_{f_1,k} = \mathcal{K}_k(A, b_{f_1}) = f_1(A)\mathcal{K}_k(A, b)$$

and

$$\operatorname{Im} W_{f_2,k} = \mathcal{K}_k\left(A^T, c_{f_2}^T\right) = f_2\left(A^T\right)\mathcal{K}_k\left(A^T, c^T\right),$$

where $f(s) = f_1(s)f_2(s)$ is again a given polynomial. Instead of fitting the moments $cA^{i-1}b$, $i = 1, \ldots, 2k$, of the function

$$g(s) = \sum_{i=1}^{\infty} cA^{i-1}bs^{-i},$$

the new spaces now yield a Padé approximation of degree k of the pre-filtered function $f(s)g(s)$ with *modified* moments $cf(A)A^{i-1}b$, $i = 1, \ldots, 2k$.

From this discussion, it is clear that the present paper shows how to extend these ideas to *rational* pre-filters. We now show how to extend rational pre-filtering as well to the Arnoldi process. In this case, one builds an orthogonal basis V_k such that

$$\begin{cases} A_g V_k = V_k \tilde{A} + q_k e_k^T, & V_k^T V_k = I_k, \\ b_g^T = [\beta_1\, 0\, \ldots\, 0]V_k^T, \end{cases} \tag{26}$$

and $\tilde{A}$ is in upper Hessenberg form. In order to apply the present ideas one needs to define a transfer function $g(s) = c_g(sI - A_g)^{-1}b_g$ and, hence, to *choose* as well a vector c_g. The updating equations of the previous section then become (with $\tilde{c} \doteq c_g V_k$)

$$\begin{bmatrix} p & c_g \\ \hline 0 & \\ \vdots & A_g \\ 0 & \end{bmatrix} \begin{bmatrix} 1 & 0 \ldots 0 \\ \hline 0 & \\ \vdots & V_k \\ 0 & \end{bmatrix} = \begin{bmatrix} 1 & 0 \ldots 0 \\ \hline 0 & \\ \vdots & V_k \\ 0 & \end{bmatrix} \begin{bmatrix} p & \tilde{c} \\ \hline 0 & \\ \vdots & \tilde{A} \\ 0 & \end{bmatrix} + \begin{bmatrix} 0 \\ \hline q_k \end{bmatrix} e_{k+1}^T,$$

$$\begin{bmatrix} 0 \\ \hline b_g \end{bmatrix} = \begin{bmatrix} 1 & 0 \ldots 0 \\ \hline 0 & \\ \vdots & V_k \\ 0 & \end{bmatrix} \begin{bmatrix} 0 \\ \hline \beta_1 \\ 0 \\ 0 \end{bmatrix}. \tag{27}$$

In order to transform this again to the standard form (26) we first need to permute β_1 from position 2 to position 1 in the transformed b_g vector, and to apply this permutation to the columns of $\operatorname{diag}\{1, V_k\}$. The corresponding similarity applied to

$$\begin{bmatrix} p & c_g \\ \hline 0 & \\ \vdots & A_g \\ 0 & \end{bmatrix}$$

destroys its Hessenberg form by introducing an element in the (3,1) position. A classical bulge chasing algorithm then pushes this element downwards until the Hessenberg form is restored. In the last step k, a rotation is applied that affects the vector e_{k+1}^{T} in a similar manner as in the Lanczos process described in (26):

$$
\begin{bmatrix} p & c_g \\ 0 & \\ \vdots & A_g \\ 0 & \end{bmatrix} \begin{bmatrix} \widehat{V}_k & \begin{matrix} * \\ \vdots \\ * \end{matrix} \end{bmatrix} = \begin{bmatrix} \widehat{V}_k & \begin{matrix} * \\ \vdots \\ * \end{matrix} \end{bmatrix} \begin{bmatrix} \hat{h}_{1,1} & \hat{h}_{1,2} & \dots & \hat{h}_{1,k} & * \\ \hat{h}_{2,1} & \ddots & & \vdots & \vdots \\ & \ddots & \ddots & \vdots & \vdots \\ & & \hat{h}_{k,k-1} & \hat{h}_{k,k} & * \\ 0 & \dots & 0 & * & * \end{bmatrix}
$$

$$
+ \hat{q}_k \left(\gamma e_k^{\mathrm{T}} + \delta e_{k+1}^{\mathrm{T}} \right). \tag{28}
$$

Again, we retrieve stage k of an Arnoldi process, after deleting the last column and row of this equation. Although this looks like a promising method for applying a rational pre-filter to a single Krylov space $\mathcal{K}_k(A, b)$ it is still unclear for the moment what the role of the c_g vector is in this procedure. This vector remains free to be chosen, while it clearly affects the resulting Hessenberg form. On the other hand, one clearly *needs* additional information to be able to apply a rational pre-filter to a single Krylov space, since $f(A)\mathcal{K}_k(A, b)$ obviously depends on *all* vectors of $\mathcal{K}_n(A, b)$ if $f(s)$ is rational.

6.   Extensions

In this concluding section, we mention a few extensions of the "simplified" theory presented earlier.

First, we address the issue of breakdowns in the SISO case. It is well known that the tridiagonal form resulting from the Lanczos process may not always exist [4,7,9]. If one of the parameters β_i of γ_i becomes zero, this implies that the system $\{A_g, b_g, c_g\}$ is not completely controllable or observable, and, hence, that the original state-space description of the transfer function was not minimal. This (rather *fortunate*) breakdown will not occur if one starts with a minimal state-space representation for $g(s)$.

The more serious breakdown is when the inner product of the vectors q_k and r_k in (24) equals zero ($r_k^{\mathrm{T}} q_k = 0$). Since these vectors are equal (up to a scaling factor) to the next vectors v_{k+1} and w_{k+1}, respectively, it becomes impossible to satisfy the biorthogonality $W_{k+1}^{\mathrm{T}} V_{k+1} = I_{k+1}$ in the next step. The standard procedure is then to perform look-ahead, in which case some of the parameters α_i, β_i and γ_i become *blocks* of compatible dimension [7,9]. The same holds true for the LR-steps. Even when starting from a scalar tridiagonal model (20), it is possible that the bulge chasing described in (23) fails. The vectors $[y_i, \; t_i]^{\mathrm{T}}$ and $[v_i, \; w_i]^{\mathrm{T}}$ now play the role of q_i and r_i, and if these vectors are orthogonal the process breaks down. Again one can get around these breakdowns by performing *block LR*-steps. We point out that even though a block form is encountered at some stage, it may disappear again after applying

another pre-multiplication with a first degree pre-filter. The use of look-ahead of course affects the complexity of the method and the moment matching property, which now holds only for those values of k that correspond to "completed" look-ahead steps [7]. For more details on look-ahead techniques, we refer to [7,9].

Next, we address the multi-input multi-output (MIMO) case. Since the derivation of the look-ahead Lanczos process, there have been several attempts to extend this also to the MIMO case (see, e.g., [1]). The same ideas of using block tridiagonal forms apply here as well, but the algorithm becomes rather intricate and many variants are possible. The pre-multiplication with a first degree transfer function described in lemma 1 still applies to MIMO systems [11] and can thus be merged with MIMO look-ahead techniques. We refer to [1] for more details of this Lanczos variant.

We also point out that the present ideas are not restricted to expansions around $s = \infty$ or to standard state-space systems. Assume we have a MIMO generalized state-space system described by

$$\begin{cases} sEx(s) = Ax(s) + Bu(s) \\ y(s) = Cx(s) + Du(s), \end{cases}$$

in the Laplace domain. The transfer function $R(s) = D + C(sE - A)^{-1}B$ can be expanded around a finite point $s = \sigma$ (which is *not* a pole of $R(s)$), as follows [2,3]:

$$R(s) = D + C\left[(s - \sigma)E - (A - \sigma E)\right]^{-1}B$$
$$= D + \sum_{i=0}^{\infty} C\left[(A - \sigma E)^{-1}E\right]^i (\sigma E - A)^{-1}B(s - \sigma)^i.$$

It is clear that in this context the role of the matrices A_g, b_g and c_g is now replaced by $(A - \sigma E)^{-1}E$, $(\sigma E - A)^{-1}B$ and C, respectively. For more details on these connections, we refer to [2,3].

A further extension is the application of the above ideas to the stabilization of an unstable system via pole/zero dislocations (see [11]). Assume we have a kth order model $\{\tilde{A}, \tilde{b}, \tilde{c}\}$ with transfer function $\tilde{g}(s) = \tilde{c}(sI_k - \tilde{A})^{-1}\tilde{b}$, but this model has unstable – and, hence, undesired – poles. We would like to construct another transfer function $\hat{g}(s)$ which has only stable poles and yet the same frequency response (i.e., $|\hat{g}(j\omega)| = |\tilde{g}(j\omega)|$, $\forall \omega$). It turns out that it is always possible to achieve this by pre-multiplying $\tilde{g}(s)$ with

$$f(s) \doteq \frac{\prod_{i=1}^{\ell}(s - p_i)}{\prod_{i=1}^{\ell}(s - \overline{p}_i)},$$

where the p_i are the *unstable poles* of $\tilde{g}(s)$. Since these are zeros of $f(s)$, they will cancel in the product $\hat{g}(s) = f(s)\tilde{g}(s)$. Moreover, the special choice of $f(s)$ ensures that $|f(j\omega)| = 1$, $\forall \omega$, and hence that $|\hat{g}(j\omega)| = |\tilde{g}(j\omega)|$, $\forall \omega$. That these ideas also extend to the MIMO case or to the discrete-time case was shown in [11].

Acknowledgements

References

[1] J. Aliaga, V. Hernandez and D. Boley, Using the block clustered nonsymmetric Lanczos algorithm to solve control problems for MIMO linear systems, in: *Proc. of the Cornelius Lanczos Internat. Centenary Conf.*, eds. J. Brown, M. Chu, D. Ellison and R. Plemmons, Raleigh, NC (1994) pp. 387–389.

[2] K. Gallivan, E. Grimme, D. Sorensen and P. Van Dooren, On some modifications of the Lanczos algorithm and the relation with Padé approximations, in: *Mathematical Research Series 7* (Akademie Verlag, Berlin, 1996) pp. 87–116.

[3] K. Gallivan, E. Grimme and P. Van Dooren, A rational Lanczos algorithm for model reduction, Numer. Algorithms 12 (1995) 33–63.

[4] G. Golub, B. Kågström and P. Van Dooren, Direct block tridiagonalization of single-input single-output systems, Systems Control Lett. 18 (1992) 109–120.

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

[6] E. Grimme, D. Sorensen and P. Van Dooren, Model reduction of state-space systems via an implicitly restarted Lanczos method, Numer. Algorithms 12 (1995) 1–31.

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

[8] 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.

[9] B.N. Parlett, Reduction to tridiagonal form and minimal realizations, SIAM J. Matrix Anal. Appl. 13 (1992) 567–593.

[10] D. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, SIAM J. Matrix Anal. Appl. 13 (1992) 357–385.

[11] P. Van Dooren, Rational and polynomial matrix factorizations via recursive pole-zero cancellation, Linear Algebra Appl. 137/138 (1990) 663–697.