Look-ahead methods for block Hankel systems*

Marc Van Barel and Adhemar Bultheel
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200 A
B-3001 Heverlee, Belgium
e-mail: Marc.VanBarel@cs.kuleuven.ac.be

October 30, 1995

Abstract

In this paper, a so-called auxiliary matrix polynomial $X_n(z)$ and a first right formal orthogonal matrix polynomial (FOMP) $A_n^1(z)$ is connected to each well-conditioned leading principal block submatrix of a given block Hankel matrix. From these two matrix polynomials, all other (inner) right FOMPs of block n of a system of block biorthogonal matrix polynomials with respect to the block Hankel moment matrix can be computed in an efficient way. The two matrix polynomials $X_n(z)$ and $A_n^1(z)$ also represent all solutions of a homogeneous interpolation problem connected to the symbol of the block Hankel moment matrix. On the other hand, they form part of the parameters of an inversion formula for the corresponding block Hankel matrix. An alternative is to use an updating formula for the solution of the block Hankel system in which $X_n(z)$ and $A_n^1(z)$ again play a crucial role. These matrix polynomials $X_n(z)$ and $A_n^1(z)$ can be updated in different ways with or without using the inner FOMPs. In this paper, we give some connections between three of these recurrence relations.

short running title: Look-ahead methods for block Hankel systems

1 Introduction

In this paper, we give a unified framework for the updating formulas used by three look-ahead methods to solve block Hankel systems where the blocks are $p \times p$. The three cases are trivial generalizations of the formulas used for $p = 1$ (scalar case) by Freund and Zha in their look-ahead Trench algorithm [13], by Cabay and Meleshko in their look-ahead algorithm for Padé approximants [6], and by Bojanczyk and Heinig in their multi-step algorithm, which is a look-ahead Schur algorithm [1]. Using these connections, it will become clear that look-ahead algorithms to solve block Hankel systems can be implemented in a lot of ways. Each of the three papers mentioned above uses its own terminology. The unifying framework shows the connection between these different viewpoints, giving the reader more insight into the look-ahead recurrences and allowing him to construct his own (hopefully) weakly stable algorithms using the basic building blocks explained here.

The paper of Freund and Zha uses the notion of formal orthogonality. Define a formal inner product for matrix polynomials as a matrix-valued bilinear form $\langle \cdot, \cdot \rangle : \mathcal{P}[z] \times \mathcal{P}[z] \rightarrow \mathbb{C}^{p \times p}$ with $\mathcal{P}[z] := \mathbb{C}[z]^{p \times p}$. In this paper, only those formal inner products are considered having a moment matrix with respect to the classical basis $I_p z^i, i = 0, 1, 2, \ldots$, which is block Hankel, i.e., $\langle I_p z^i, I_p z^j \rangle = h_{i+j} \in \mathbb{C}^{p \times p}$ (Section 2). In the strongly regular case, i.e.,

*This research was supported by the National Fund for Scientific Research (NFWO), project Lanczos, grant #2.0042.93 and by the Human Capital and Mobility project ROLLS of the European Community under contract ERBCHRXCT930416.

when all leading principal block submatrices $H_{\nu,\nu} := [h_{i+j}]_{i=0,1,\ldots,\nu-1}^{j=0,1,\ldots,\nu-1}$ are nonsingular, there exists a unique system of biorthogonal monic matrix polynomials $\{B_i(z), A_i(z)\}_{i=0,1,\ldots}$ of degree i satisfying $\langle B_i(z), A_j(z)\rangle = \delta_{ij} D_{ii}$ with D_{ii} nonsingular. To compute this system of orthogonal matrix polynomials a block modified two-sided Gram-Schmidt method can be used as for a general strongly regular moment matrix $M := [\langle I_p z^i, I_p z^j \rangle]_{i=0,1,\ldots}^{j=0,1,\cdots}$ (Section 3). In Section 10, we give a more efficient method using the block Hankel structure of the moment matrix. Written in terms of the stacking vectors of the formal orthogonal matrix polynomials (FOMPs), the system of orthogonal matrix polynomials gives an inverse block LDU factorization of the block Hankel matrix where the blocks of the block diagonal D are $(p \times p)$. If the moment matrix is not strongly regular, we define in Section 4 a system of block orthogonal polynomial matrices with respect to so-called block indices ν_i, $i = 1, 2, \ldots$. These block indices indicate (a subset of) the indices of the nonsingular leading principal block submatrices of the moment matrix. It can be a subset in the strict sense to enhance the numerical stability of the updating method of Section 9. Now, the FOMPs are divided in blocks of $\alpha_i := \nu_i - \nu_{i-1}$ ($\nu_0 := 0$) matrix polynomials. The first FOMP of a block is called the *true* FOMP. The other FOMPs of the block, if they exist, are called *inner*. In this case, we also derive an inverse block LDU factorization of the block Hankel matrix but the diagonal blocks of D have bigger size, more precisely, the ith block has dimension $\alpha_i p$. In a similar way, the system of block orthogonal polynomial matrices can be computed by a block modified two-sided Gram-Schmidt method with respect to the block indices ν_i. We refer the interested reader to [4]. In Section 5, we show how to compute the true or first FOMP $A_n^1(z)$ of block n. The paper of Cabay and Meleshko uses the notion of Padé approximation to derive their recurrence formula. By connecting formal orthogonality to a homogeneous interpolation problem in Section 6, which can be viewed as a Padé approximation problem, Section 7 gives an efficient method to compute the inner FOMPs of block n based on an auxiliary polynomial matrix $X_n(z)$ and the true FOMP $A_n^1(z)$. Because these two matrix polynomials $X_n(z)$ and $A_n^1(z)$ represent in a compact way all solutions of a right homogeneous interpolation problem as described in Section 8, they can be updated in a very efficient way into $X_{n'}(z)$ and $A_{n'}^1(z)$ as long as the difference between the block indices $\nu_{n'}$ and ν_n is small. Under this condition, the updating procedure requires $O(\nu_n^2)$ FLOPS to compute $X_n(z)$ and $A_n^1(z)$. This updating procedure is used by Cabay and Meleshko and involves the solution of a block Sylvester system. Section 11 shows that $X_n(z)$ and $A_n^1(z)$ form part of the parameters of an inversion formula for the block Hankel matrix H_{ν_n,ν_n}. Once, all inversion parameters are computed, the inversion formula can be applied in a very efficient way on a given right-hand side. This leads to an $O(\nu_n^2)$ FLOPS algorithm to solve a block Hankel system of linear equations compared to the classical methods like, e.g., Gauss elimination with partial pivoting which destroys the Hankel structure and requires $O(\nu_n^3)$ FLOPS. An alternative is to use the updating formula of Section 12.

At this point, we have only looked at the efficiency of the computations. We have only mentioned that the block indices are choosen in such a way that the corresponding leading principal block submatrices are well-conditioned. This determines the look-ahead strategy. For details concerning the look-ahead strategy, we refer the reader to the literature. For scalar Hankel matrices, several of these look-ahead algorithms have been designed [6, 13, 1]. Also for Toeplitz matrices, i.e. matrices having the same entries on each antidiagonal, such look-ahead schemes have been constructed [8, 7, 12, 11, 14, 20]. Even superfast, i.e. requiring $O(\nu \log^2 \nu)$ operations, look-ahead algorithms were developed [16, 17, 19]. An error analysis was done in [5] for generalized Sylvester matrices, for Hankel matrices in [6] and for block Toeplitz systems in [22]. This last reference also compares three possible look-ahead schemes with numerical examples.

In Section 13, the recurrence relations, involving not only the auxiliary and the first FOMP but also the inner FOMPs, are derived for a look-ahead Trench algorithm very similar to the scalar version of Freund and Zha [13]. Section 14 gives the easy generalization of the scalar recurrences of a look-ahead Schur-type algorithm of Bojanczyk and Heinig [1]

where also the inner FOMPs are explicitly involved. In Section 15, we show that by solving the block Sylvester system in a specific way, the inner FOMPs appear explicitly.

2 Formal inner product of two matrix polynomials

We denote the set of $(p \times p)$ matrix polynomials as $\mathcal{P}[z] := \mathbb{C}[z]^{p\times p}$.

Definition 2.1 (formal inner product) *We define a* formal inner product *of two matrix polynomials as a bilinear form* $\langle \cdot, \cdot \rangle : \mathcal{P}[z] \times \mathcal{P}[z] \to \mathbb{C}^{p\times p}$, *i.e.,*

$$\begin{aligned} \langle P_1(z)A_1 + P_2(z)A_2, Q(z)\rangle &= A_1^T \langle P_1(z), Q(z)\rangle + A_2^T \langle P_2(z), Q(z)\rangle \\ \langle P(z), Q_1(z)A_1 + Q_2(z)A_2\rangle &= \langle P(z), Q_1(z)\rangle A_1 + \langle P(z), Q_2(z)\rangle A_2 \end{aligned}$$

for each $P_1(z), P_2(z) \in \mathcal{P}[z]$ *and for each* $A_1, A_2 \in \mathbb{C}^{p\times p}$.

The (i,j)th block element of the moment matrix M is defined as

$$M_{i,j} := \left\langle I_p z^i, I_p z^j \right\rangle.$$

In the sequel, we consider the formal inner product whose moment matrix is block Hankel, i.e.,

$$H := M = [h_{i+j}]_{i=0,1,2,\ldots}^{j=0,1,2,\ldots}, \qquad h_{i+j} \in \mathbb{C}^{p\times p}.$$

Hence, the inner product of the two monomials $I_p z^i$ and $I_p z^j$ only depends on the sum of the degrees i and j,

$$\left\langle I_p z^i, I_p z^j \right\rangle = h_{i+j}.$$

So, we can also write the inner product as a matrix-valued linear functional

$$\langle P(z), Q(z)\rangle = \mu\left((P(z))^T \cdot Q(z)\right)$$

with $\mu(I_p z^i) = h_i, \qquad i = 0, \pm 1, \pm 2, \ldots$.

3 Formal (bi)orthogonality

Definition 3.1 (formal orthogonality) *The matrix polynomial* $Q(z)$ *is* right orthogonal *to the matrix polynomial* $P(z)$ *or* $P(z)$ *is* left orthogonal *to* $Q(z)$ *iff* $\langle P(z), Q(z)\rangle = 0$.

The stacking vector $\mathbf{Q}$ of a polynomial matrix $Q(z) = \sum_{i=0}^{\alpha} Q_i z^i$ is the block vector of the coefficients of the polynomial matrix, i.e.,

$$\mathbf{Q} := [Q_0^T, Q_1^T, \ldots, Q_\alpha^T, 0, \ldots]^T$$

where the number of additional zero coefficients should be clear from the context. Using the stacking vectors $\mathbf{P}$ and $\mathbf{Q}$ of $P(z)$ and $Q(z)$, we can express formal orthogonality as

$$\langle P(z), Q(z)\rangle = 0 \quad \Leftrightarrow \quad \mathbf{P}^T H \mathbf{Q} = 0.$$

If we denote part of the block Hankel matrix H as $H_{m,n} := [h_{i+j}]_{i=0,1,\ldots,m-1}^{j=0,1,\ldots,n-1}$, the polynomial matrix $Q(z)$ of degree α is right orthogonal to all matrix polynomials of degree $< \beta$ iff

$$H_{\beta,\alpha+1}\mathbf{Q} = 0.$$

Note that in general $H \neq H^T$. Hence, left and right formal orthogonal matrix polynomials (FOMPs) are not directly connected. However, in the scalar case, i.e., $p = 1$, $H = H^T$ and we do not have to make the distinction between left and right orthogonality. In [10, 2, 9], more details can be found about formal orthogonal polynomials with respect to a fixed linear functional.

Definition 3.2 (biorthogonality) $\{B_i(z), A_i(z)\}_{i=0,1,\ldots}$ *is a* system of biorthogonal matrix polynomials *with respect to the formal inner product* $\langle \cdot, \cdot \rangle$ *iff*

- $\deg B_i(z) = i$ *and the highest degree coefficient (hdc) of* $B_i(z)$ *is* I_p *(monic)*
- $\deg A_i(z) = i$ *and hdc of* $A_i(z)$ *is* I_p
- $\langle B_i(z), A_j(z) \rangle = \delta_{ij} D_{ii}$ *with* D_{ii} *nonsingular.*

This system of biorthogonal matrix polynomials can be computed using, e.g., the two-sided modified Gram-Schmidt algorithm.

$$
\begin{aligned}
&B_0 = A_0 = I_p \\
&\textbf{for } k = 0, 1, 2, \ldots \\
&\quad D_{k,k} = \langle B_k, A_k \rangle \\
&\quad A_{k+1} = zA_k - \sum_{i=0}^{k} A_i D_{i,i}^{-1} \langle B_i, zA_k \rangle \\
&\quad B_{k+1} = zB_k - \sum_{i=0}^{k} B_i D_{i,i}^{-T} \langle zB_k, A_i \rangle^T \\
&\textbf{endfor}
\end{aligned}
$$

If we take the stacking vectors of the matrix polynomials $A_i(z)$ as block columns of the matrix $A := [\mathbf{A}_0, \mathbf{A}_1, \ldots]$, we get that A is (block) unit upper triangular. With a similar definition for B and $D := \mathrm{diag}(D_{ii})$, the biorthogonality leads to an inverse block LDU factorization of the block Hankel matrix

$$B^T H A = D.$$

Note that this factorization can only exist if all leading principal block submatrices $H_{i,i}$ are nonsingular. We will drop this condition in the next section.

4 Formal block biorthogonality

Suppose now that not all leading principal block submatrices $H_{1,1}, H_{2,2}, \ldots$ are nonsingular. Denote the indices of the successive ones by $\nu_1, \nu_2, \ldots$ ($\nu_0 := 0$). They are called the ***block indices.*** It turns out that in this case we have to put several FOMPs into blocks. The jth block consists of $\alpha_{j+1} := \nu_{j+1} - \nu_j$ matrix polynomials

$$\overline{A}_j(z) := \left[A_{\nu_j}(z), A_{\nu_j+1}(z), \ldots, , A_{\nu_{j+1}-1}(z)\right], \quad j = 0, 1, 2, \ldots, N.$$

If there are an infinite number of nonsingular leading principal submatrices, $N = +\infty$. Otherwise, N is finite and $\nu_{N+1} = \alpha_{N+1} = +\infty$. The kth FOMP of the jth block is denoted as

$$A_j^k(z) := A_{\nu_j+k-1}(z), \qquad k = 1, 2 \ldots, \alpha_{j+1}.$$

Definition 4.1 (block biorthogonality) $\{B_i(z), A_i(z)\}_{i=0,1,\ldots}$ *is a* system of block biorthogonal matrix polynomials *with respect to the formal inner product* $\langle \cdot, \cdot \rangle$ *and the block indices* $\nu_i, i = 0, 1, 2, \ldots$ *iff*

- $\deg B_i(z) = i$ *and hdc of* $B_i(z)$ *is* I_p
- $\deg A_i(z) = i$ *and hdc of* $A_i(z)$ *is* I_p
- $\langle \overline{B}_i(z), \overline{A}_j(z) \rangle := \left[\langle B_i^k(z), A_j^l(z) \rangle\right]_{k=1,2,\ldots,\alpha_{i+1}}^{l=1,2,\ldots,\alpha_{j+1}} = \delta_{ij} D_{ij}$
  with D_{ii} *a nonsingular* $p\alpha_{i+1} \times p\alpha_{i+1}$ *matrix*

As for the strongly regular case, a block two-sided modified Gram-Schmidt algorithm could be used to compute the system of block biorthogonal polynomials but we will not go into the details here. We refer the interested reader to [4]. Using the matrices A and B whose block columns are the stacking vectors of the right and left biorthogonal matrix polynomials respectively, we get the following block inverse LDU factorization for the matrix H

$$B^T H A = D \tag{1}$$

where the size of the diagonal blocks of D is as small as possible. Note that A and B are again unit upper triangular. In the next sections, we show how to compute the right (bi)orthogonal matrix polynomials in a more efficient way. Note that the previous definition also applies when the block indices are connected to well-conditioned leading principal block submatrices determined by one of the possible look-ahead strategies.

5 Computation of the first monic right FOMP

The first formal orthogonal matrix polynomial of block n $A_n^1(z) := A_{\nu_n}(z)$, also called the *true* FOMP, has degree ν_n, is monic and is right orthogonal to all previous blocks of left orthogonal polynomials. Hence, $A_n^1(z)$ is right orthogonal to all previous left orthogonal polynomials or

$$\left\langle B_i(z), A_n^1(z) \right\rangle = 0, \qquad i = 0, 1, \ldots, \nu_n - 1$$

or

$$\left\langle I_p z^i, A_n^1(z) \right\rangle = 0, \qquad i = 0, 1, \ldots, \nu_n - 1.$$

Using the stacking vector

$$\mathbf{A}_n^1 = \begin{bmatrix} \dot{A}_n^1 \\ I_p \end{bmatrix} \in \mathbb{C}^{(\nu_n+1)p \times p}$$

we get the following equivalent set of homogeneous linear equations

$$H_{\nu_n,\nu_n+1}\mathbf{A}_n^1 = 0 \qquad \text{or} \qquad H_{\nu_n,\nu_n}\dot{A}_n^1 = -H_{\nu_n+1}^c$$

with $H_{\nu_n+1}^c$ the last block column of H_{ν_n,ν_n+1}. Because the block index ν_n indicates the *nonsingular* leading principal block submatrix H_{ν_n,ν_n}, the first monic right FOMP $A_n^1(z)$ is uniquely defined. A more efficient way to compute $A_n^1(z)$ will be given in Section 9.

6 A homogeneous interpolation problem

We can express formal orthogonality as a homogeneous interpolation problem in the following way. We define the symbol $H(z)$ of the moment matrix H as $H(z) := \sum_{k=0}^{+\infty} t_k z^{-k-1}$. A matrix polynomial $Q(z)$ of degree α is right orthogonal to all matrix polynomials of degree $< \beta$ iff $H_{\beta,\alpha+1}\mathbf{Q} = 0$ iff $Q(z)$ satisfies the following homogeneous interpolation problem $H(z)Q(z) = O_-(z^{-\beta-1}) + O_+(z^0)$ where $O_-(z^\gamma)$ represents a series of the form $O_-(z^\gamma) := c_\gamma z^\gamma + c_{\gamma-1}z^{\gamma-1} + \cdots$ and $O_+(z^\gamma)$ represents a series of the form $O_+(z^\gamma) := c_\gamma z^\gamma + c_{\gamma+1}z^{\gamma+1} + \cdots$. For the first monic right FOMP in block n, this leads to the following interpolation conditions

$$H(z)A_n^1(z) = O_-(z^{-\nu_n-1}) + O_+(z^0) = R_a^-(z) + R_a^+(z)$$

with $\begin{cases} R_a^-(z) := r_{a,-1}^- z^{-\nu_n-1} + r_{a,-2}^- z^{-\nu_n-2} + \cdots \\ R_a^+(z) := r_{a,0}^+ z^0 + r_{a,1}^+ z^1 + \cdots + r_{a,\nu_n-1}^+ z^{\nu_n-1} \end{cases}$.

The series $R_a^-(z)$ and $R_a^+(z)$ are called the $(-)$ and $(+)$ residual series of $A_n^1(z)$ respectively. Because the highest degree coefficient of $A_n^1(z)$ is I_p, we get

$$H(z) - R_a^+(z)(A_n^1(z))^{-1} = O_-(z^{-2\nu_n-1})$$

or $R_a^+(z)(A_n^1(z))^{-1}$ is a minimal partial realization of $H(z)$ of McMillan degree $p\nu_n$. $(z^{\nu_n-1}R_a^+(z^{-1}))(z^{\nu_n}A_n^1(z^{-1}))^{-1}$ is a (ν_n-1,ν_n) matrix Padé approximant of $z^{-1}H(z^{-1})$. This is the terminology in which the look-ahead method of Cabay and Meleshko [6] is designed.

7 Computation of the other right monic FOMPs

The FOMPs of block 0 should satisfy

$$H(z)A_0^j(z) = O_-(z^{-1}) + O_+(z^0), \qquad j = 1,2,\ldots,\alpha_1.$$

Hence, any monic matrix polynomial of degree j can be taken as $A_0^j(z)$, $j = 1,2,\ldots,\alpha_1$. All the FOMPs of block $n > 0$ should satisfy the interpolation conditions

$$H(z)A_n^j(z) = O_-(z^{-\nu_n-1}) + O_+(z^0), \qquad j = 1,2,\ldots,\alpha_{n+1}.$$

Because H_{ν_n,ν_n} is nonsingular, this defines $A_n^1(z)$ uniquely (see Section 5). To compute the other right monic FOMPs of block n, called *inner* FOMPs, we introduce the auxiliary matrix polynomial of block n $X_n(z)$ having degree $< \nu_n$ and satisfying

$$H_{\nu_n,\nu_n}\mathbf{X}_n = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I_p \end{bmatrix} \qquad \Leftrightarrow \qquad H(z)X_n(z) = O_-(z^{-\nu_n-1}) + I_p z^{\nu_n} + O_+(z^0).$$

Note that $X_n(z)$ is also uniquely defined. The residual series of $X_n(z)$ are defined as

$$H(z)X_n(z) =: R_x^-(z) + R_x^+(z), \qquad \text{with } R_x^+(z) = O_+(z^0)$$

and

$$R_x^-(z) := I_p z^{-\nu_n} + r_{x,-1}^- z^{-\nu_n-1} + r_{x,-2}^- z^{-\nu_n-2} + \cdots.$$

This allows us to compute a possible choice for the inner FOMPs as

$$A_n^{j+1}(z) = z^j A_n^1(z) + X_n(z)D^{(j)}(z), \quad j = 1,2,\ldots,\alpha_{n+1}-1$$

where the coefficients $d_i^{(j)}$ of the monic matrix polynomial $D^{(j)}(z)$ of degree $< j$ satisfy

$$\begin{bmatrix} I_p & r_{x,-1}^- & \cdots & r_{x,-j+1}^- \\ & I_p & \cdots & r_{x,-j+2}^- \\ & & \ddots & \vdots \\ & & & I_p \end{bmatrix} \begin{bmatrix} d_0^{(j)} \\ d_1^{(j)} \\ \vdots \\ d_{j-1}^{(j)} \end{bmatrix} = - \begin{bmatrix} r_{a,-j}^- \\ r_{a,-j+1}^- \\ \vdots \\ r_{a,-1}^- \end{bmatrix}. \tag{2}$$

Because $D^{(j)}(z) = zD^{(j-1)}(z) + d_0^{(j)}$, $j = 2,3,\ldots,\alpha_{n+1}-1$, we also have the recurrence

$$A_n^{j+1}(z) = zA_n^j(z) + X_n(z)d_0^{(j)}. \tag{3}$$

If we define the residual series for $A_n^j(z)$ as

$$H(z)A_n^j(z) =: R_{a,j}^-(z) + R_{a,j}^+(z)$$

with $R_{a,j}^-(z) = O_-(z^{-\nu_n-1})$ and $R_{a,j}^+(z) = O_+(z^0)$ and matrix polynomial, it is clear that the same recurrence (3) also applies for the residual series, i.e.,

$$\begin{aligned} R_{a,j+1}^-(z) &= zR_{a,j}^-(z) + R_x^-(z)d_0^{(j)} \\ R_{a,j+1}^+(z) &= zR_{a,j}^+(z) + R_x^+(z)d_0^{(j)}. \end{aligned}$$

One could use these recurrences to compute those coefficients of the residual series which are needed during the rest of the computations. This would then be a Schur-type algorithm. On the other hand, only those coefficients could be computed needed in (2) to update the inner FOMPs. This would be a Levinson-type method. Note that the inner FOMPs are not uniquely defined. The nth block $\overline{A}_n(z)$ can be replaced by $\overline{A}_n(z)U_n$ for any (block) unit upper triangular $(p\alpha_{n+1} \times p\alpha_{n+1})$ matrix U_n. In a similar way an nth block of left FOMPs can be constructed. The previous construction for the nth block of right monic FOMPs proves the existence of the inverse LDU factorization (1). This section shows that once we know the auxilary matrix polynomial $X_n(z)$ and the first FOMP $A_n^1(z)$ of block n, the inner FOMPs can be computed in an efficient way. In the next sections, we show how to compute $X_n(z)$ and $A_n^1(z)$ in an efficient way based on $X_{n-1}(z)$ and $A_{n-1}^1(z)$.

8 Right homogeneous interpolation problem

In this section, we show that $X_n(z)$ and $A_n^1(z)$ form a compact representation for all solutions of a right homogeneous interpolation problem. Consider the symbol $H(z) := \sum_{k=0}^{+\infty} t_k z^{-k-1}$ again and fix the value of $\nu \in \mathbb{N}$. For each $\alpha \in \mathbb{N}$, we define the set S_α^ν as

$$S_\alpha^\nu := \left\{ q(z) \in \mathbb{C}[z]^{p\times 1} \,|\, \deg q(z) \leq \alpha \text{ and } H(z)q(z) = O_-(z^{-2\nu+\alpha-1}) + O_+(z^0) \right\}.$$

Note that for $\alpha \geq 2\nu$, S_α^ν consists of all vector polynomials $q(z)$ with $\deg q(z) \leq \alpha$ because there are no interpolation conditions to be fulfilled.
Take $\nu = \nu_n$. The auxiliary matrix polynomial $X_n(z)$ is completely characterized by the following conditions

- $X_n(z)$ has degree $< \nu$,
- the columns of $X_n(z)$ belong to $S_{\nu+1}^\nu$,
- the hdc of the $(-)$ residual series of $X_n(z)$ is $I_p z^{-\nu}$.

The first FOMP $A_n^1(z)$ is completely characterized by the following conditions

- $A_n^1(z)$ is monic of degree ν,
- the columns of $A_n^1(z)$ belong to S_ν^ν.

Lemma 8.1 *The block Hankel matrix $H_{\nu,\nu}$ is nonsingular iff all nontrivial solutions $q(z)$ of S_ν^ν have a nonzero highest degree coefficient.*

proof. The stacking vector $\mathbf{q}$ of a nontrivial solution $q(z)$ of S_ν^ν having degree $< \nu$ should satisfy the set of linear homogeneous equations: $H_{\nu,\nu}\mathbf{q} = 0$. Hence, such a nontrivial $q(z)$ exists iff $H_{\nu,\nu}$ is singular. □

Theorem 8.2 *If $H_{\nu,\nu}$ is nonsingular, i.e., $\nu = \nu_n$, each $q(z) \in S_\alpha^\nu$ can be written in a unique way as*

$$q(z) = \begin{bmatrix} X_n(z) & A_n^1(z) \end{bmatrix} \begin{bmatrix} q_x(z) \\ q_a(z) \end{bmatrix} \tag{4}$$

$$\text{with} \begin{cases} q_x(z), q_a(z) \in \mathbb{C}[z]^{p\times 1} \\ \deg q_x(z) \leq \min(\alpha-\nu-1, \nu-1) \\ \deg q_a(z) \leq \alpha - \nu \end{cases}.$$

proof. The proof is by induction on α. It is clear that the only element of S_α^ν for $\alpha < \nu$ is the zero vector polynomial. For $\alpha = \nu$, the columns of $A_n^1(z)$ are elements of S_α^ν. Because its highest degree coefficients are the columns of I_q, hence linearly independent, any element $q(z)$ of S_ν^ν can be written in a unique way as $q(z) = A_n^1(z)q_a + q'(z)$ with $\deg q_a \leq 0$

(hence, q_a is constant) and $\deg q' < \nu$. However, because $q' \in S_\nu^\nu$ with highest degree coefficient equal to zero, and $H_{\nu,\nu}$ is nonsingular, we know from Lemma 8.1 that $q' = 0$. This proves the theorem for $\alpha = \nu$. Suppose the theorem is true for $\alpha < 2\nu$. Because $H(z)X_n(z) = O_-(z^{-\nu-1}) + I_p z^{-\nu} + O_+(z^0)$, we can write every element $q(z)$ of $S_{2\nu}^\nu$ in a unique way as $q(z) = z^\nu A_n^1(z)q_{a,\nu} + z^{\nu-1}X_n(z)q_{x,\nu-1} + q'(z)$ by making the highest degree coefficient of q and the coefficient of z^{-1} in $H(z)q(z)$ zero. In other words, $q'(z) \in S_{2\nu-1}^\nu$ or $q'(z)$ can be written in a unique way as $q'(z) = \left[X_n(z) \;\; A_n^1(z)\right]\left[q_x(z) \;\; q_a(z)\right]$ with $\deg q_x \leq \nu - 2$ and $\deg q_a \leq \nu - 1$. Hence, we can write $q(z)$ in a unique way as $q(z) = A_n^1(z)(q_a(z) + z^\nu q_{a,\nu}) + X_n(z)(q_x(z) + z^{\nu-1}q_{x,\nu-1})$. This proves the theorem for $\alpha = 2\nu$.
For $\alpha \geq 2\nu$, the elements q of S_α^ν only have to satisfy the degree condition $\deg q \leq \alpha$. Hence, any element $q(z)$ of S_α^ν with $\alpha > 2\nu$ can be written in a unique way as $q(z) = z^{\alpha-\nu}A_n^1(z)q_{a,\alpha-\nu} + q'(z)$ with $q'(z) \in S_{\alpha-1}^\nu$. Applying the same reasoning as above this proves the theorem for $\alpha > 2\nu$. □

For singular $H_{\nu,\nu}$, there exists a similar representation for all solutions. Note that if we allow $\deg q_x(z) \leq \alpha - \nu - 1$, each $q(z) \in S_\alpha^\nu$ can still be written as (4) but not necessarily in a unique way anymore.

9 Efficient updating for $X_n(z)$ and $A_n^1(z)$

Consider $H_{\nu,\nu}$ and $H_{\nu',\nu'}$ nonsingular with $\nu = \nu_n$ and $\nu' = \nu_{n'}$, and $n < n'$. In this section, we efficiently compute $X_{n'}(z)$ and $A_{n'}^1(z)$ based on $X_n(z)$ and $A_n^1(z)$. At level n', the columns of $X_{n'}(z)$ and $A_{n'}^1(z)$ have to satisfy

- the columns of $X_{n'}(z)$ belong to $S_{\nu'+1}^{\nu'} \subset S_{\nu'+1}^{\nu}$,
- the columns of $A_{n'}^1(z)$ belong to $S_{\nu'}^{\nu'} \subset S_{\nu'}^{\nu}$.

Because $X_n(z)$ and $A_n^1(z)$ allow to represent all solutions of S_α^ν, for each value of $\alpha \in \mathbb{N}$, $X_{n'}(z)$ and $A_{n'}^1(z)$ can be written as

$$\left[X_{n'}(z) \;\; A_{n'}^1(z)\right] = \left[X_n(z) \;\; A_n^1(z)\right]\begin{bmatrix} C(z) & E(z) \\ D(z) & F(z) \end{bmatrix} \tag{5}$$

satisfying interpolation conditions with degree and normalization constraints, i.e.,

$$\begin{cases} H(z)A_{n'}^1(z) = O_-(z^{-\nu_{n'}-1}) + O_+(z^0) \\ H(z)X_{n'}(z) = O_-(z^{-\nu_{n'}-1}) + I_p z^{-\nu_{n'}} + O_+(z^0) \\ \deg A_{n'}(z) = \nu_{n'} \text{ and monic} \\ \deg X_{n'}(z) < \nu_{n'} \end{cases}$$

Working this out using the coefficients of the unknown polynomial matrices $C(z)$, $D(z)$, $E(z)$, $F(z)$, we get the following set of linear equations

$$S_{\nu\to\nu'}\left[\begin{array}{cc} c_0 & e_0 \\ c_1 & e_1 \\ \vdots & \vdots \\ c_{\delta-1} & e_{\delta-1} \\ \hline d_0 & f_0 \\ d_1 & f_1 \\ \vdots & \vdots \\ d_{\delta_n-1} & f_{\delta_n-1} \end{array}\right] = \left[\begin{array}{cc} 0 & -r_{a,-\delta_n+\delta-1}^- \\ 0 & -r_{a,-\delta_n+\delta-2}^- \\ \vdots & \vdots \\ 0 & -r_{a,-\delta_n}^- \\ \hline 0 & -r_{a,-\delta_n-1}^- \\ \vdots & \vdots \\ \vdots & \vdots \\ 0 & -r_{a,-2\delta_n+1}^- \\ I_p & -r_{a,-2\delta_n}^- \end{array}\right] \tag{6}$$

with

$$S_{\nu\to\nu'} := \left[\begin{array}{cccc|cccc} 0 & 0 & \cdots & r^-_{x,0} & 0 & 0 & \cdots & r^-_{a,-\delta_n+\delta} \\ 0 & 0 & \cdots & r^-_{x,-1} & 0 & 0 & \cdots & r^-_{a,-\delta_n+\delta-1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ r^-_{x,0} & r^-_{x,-1} & \cdots & r^-_{x,-\delta+1} & 0 & r^-_{a,-1} & \cdots & r^-_{a,-\delta_n+1} \\ r^-_{x,-1} & r^-_{x,-2} & \cdots & r^-_{x,-\delta} & r^-_{a,-1} & r^-_{a,-2} & \cdots & r^-_{a,-\delta_n} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ r^-_{x,-\delta_n+1} & r^-_{x,-\delta_n} & \cdots & r^-_{x,-\delta_n-\delta+2} & r^-_{a,-\delta_n+1} & r^-_{a,-\delta_n} & \cdots & r^-_{a,-2\delta_n+2} \\ r^-_{x,-\delta_n} & r^-_{x,-\delta_n-1} & \cdots & r^-_{x,-\delta_n-\delta+1} & r^-_{a,-\delta_n} & r^-_{a,-\delta_n-1} & \cdots & r^-_{a,-2\delta_n+1} \end{array}\right],$$

$\delta_n := \nu_{n'} - \nu_n$ and $\delta := \min\{\delta_n, \nu_n\}$. Hence, the coefficient matrix $S_{\nu\to\nu'}$ has $\delta + \delta_n$ block rows and columns and is block Sylvester.

Theorem 9.1 *$H_{\nu',\nu'}$ is nonsingular $\Leftrightarrow$ $S_{\nu\to\nu'}$ is nonsingular.*

proof. If $S_{\nu\to\nu'}$ is singular, there exist stacking vectors $\mathbf{e}$ and $\mathbf{f}$ satisfying $S_{\nu\to\nu'}[\mathbf{e}^T\ \mathbf{f}^T]^T = 0$ with $[\mathbf{e}^T\ \mathbf{f}^T]^T \neq 0$. Hence, the nontrivial polynomial vector $q(z) = [X_n(z)\ \ A^1_n(z)]\,[e(z)\ \ f(z)]^T$ has degree $< \nu'$ and satisfies $H(z)q(z) = O_-(z^{-\nu_{n'}-1}) + O_+(z^0)$ or $H_{\nu',\nu'}\mathbf{q} = 0$. Hence, $H_{\nu',\nu'}$ is singular and vice versa. □

For $p = 1$, (5) is the updating formula used in the weakly stable algorithm for computing Padé approximants by Cabay and Meleshko [6].
Note that considering the remark at the end of Section 8 we can also take $S_{\nu\to\nu'}$ with $\delta := \delta_n$ leading to the same solution $X_{n'}(z)$ and $A^1_{n'}(z)$ because these are uniquely defined. In the sequel, we shall take this definition for $S_{\nu\to\nu'}$. Hence, it is a $2\delta_n$ block Sylvester matrix. The same recurrence relation (5) can be used to update the residual series. In the same way as in Section 7, we can distinguish between Schur-type and Levinson-type methods. In this section, we have given an updating method for the auxiliary matrix polynomial $X_n(z)$ and the true FOMP $A^1_n(z)$. When the difference between two successive block indices $\alpha_{j+1} := \nu_{j+1} - \nu_j$ stays small, the computation of all $X_i(z)$ and $A^1_i(z)$, $i = 1, 2, \ldots, n$ requires $O(\nu_n^2)$ FLOPS.

10 The definite or strongly regular case

If all leading principal block submatrices $T_{i,i}$, $i = 1, 2, \ldots$ are nonsingular, we can choose the block indices $\nu_i = i$. This is called the *strongly regular* or *definite* case. There are no inner FOMPs and all the blocks of FOMPs just contain the true FOMPs, i.e., $A^1_n(z) = A_n(z)$. The set of linear equations (6) to update the auxiliary and true matrix polynomial is

$$\left[\begin{array}{cc} I_p & 0 \\ r^-_{x,-1} & r^-_{a,-1} \end{array}\right]\left[\begin{array}{cc} c_0 & e_0 \\ d_0 & f_0 \end{array}\right] = \left[\begin{array}{cc} 0 & -r^-_{a,-1} \\ I_p & -r^-_{a,-2} \end{array}\right].$$

Hence,

$$[X_{n+1}(z)\ \ A_{n+1}(z)] = [X_n(z)\ \ A_n(z)]\left[\begin{array}{cc} C(z) & E(z) \\ D(z) & F(z) \end{array}\right]$$

with $C(z) = 0$, $D(z) = -\left(r^-_{a,-1}\right)^{-1}$, $E(z) = -r^-_{a,-1}$ and $F(z) = I_p z + \left(r^-_{a,-1}\right)^{-1} r^-_{x,-1} r^-_{a,-1} - \left(r^-_{a,-1}\right)^{-1} r^-_{a,-2}$. with the initialization: $[h_0]\,[x_{0,1}\ \ a_{0,1}] = [I_p\ \ -h_1]$ or $[X_1(z)\ \ A_1(z)] = \left[h_0^{-1}\ \ I_p z - h_0^{-1}h_1\right]$. Note that a necessary and sufficient condition to have the definite case is that each residual coefficient $r^-_{a,-1}$ is nonsingular. Considering the matrix Padé table for $z^{-1}H(z^{-1})$ [3], it turns out that $z^n A_n(z^{-1})$ is the comonic denominator at entry $(n-1, n)$ of the Padé table and $z^{n-1}X_n(z^{-1})$ is the denominator at entry $(n-2, n-1)$.

11 Inverse of a block Hankel matrix

The matrix polynomials $X_n(z)$ and $A_n^1(z)$ are also parameters of an inversion formula for the block Hankel matrix H_{ν_n,ν_n}.

Theorem 11.1 (inversion formula [21]) *Let $H_{\nu,\nu}$ be nonsingular. Consider the sets of linear equations*

$$\begin{bmatrix} h_0 & h_1 & \cdots & h_{\nu-1} & h_\nu \\ h_1 & h_2 & \cdots & h_\nu & h_{\nu+1} \\ \vdots & \vdots & & \vdots & \vdots \\ h_{\nu-1} & h_\nu & \cdots & h_{2\nu-2} & h_{2\nu-1} \end{bmatrix} \begin{bmatrix} x_0 & a_0 \\ x_1 & a_1 \\ \vdots & \vdots \\ x_{\nu-1} & a_{\nu-1} \\ 0 & I_p \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ I_p & 0 \end{bmatrix} \tag{7}$$

and

$$\begin{bmatrix} u_0 & u_1 & \cdots & u_{\nu-1} & 0 \\ w_0 & w_1 & \cdots & w_{\nu-1} & I_p \end{bmatrix} \begin{bmatrix} h_0 & h_1 & \cdots & h_{\nu-1} & h_\nu \\ h_1 & h_2 & \cdots & h_\nu & h_{\nu+1} \\ \vdots & \vdots & & \vdots & \vdots \\ h_{\nu-1} & h_\nu & \cdots & h_{2\nu-2} & h_{2\nu-1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & I_p \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{8}$$

Then, the inverse of $H_{\nu,\nu}$ can be written as

$$H_{\nu,\nu}^{-1} = \begin{bmatrix} x_0 & & & \\ x_1 & x_0 & & \\ \vdots & \vdots & \ddots & \\ x_{\nu-1} & x_{\nu-2} & \cdots & x_0 \end{bmatrix} \begin{bmatrix} w_1 & \cdots & w_{\nu-1} & 1 \\ \vdots & \iddots & \iddots & \\ w_{\nu-1} & & & \\ I_p & & & \end{bmatrix} - \begin{bmatrix} a_0 & & & \\ a_1 & a_0 & & \\ \vdots & \vdots & \ddots & \\ a_{\nu-1} & a_{\nu-2} & \cdots & a_0 \end{bmatrix} \begin{bmatrix} u_1 & \cdots & u_{\nu-1} & 0 \\ \vdots & \iddots & & \\ u_{\nu-1} & & & \\ 0 & & & \end{bmatrix}.$$

Note that with $\nu = \nu_n$, $X_n(z) = X(z) := \sum_{i=0}^{\nu-1} x_i z^i$ and $A_n^1(z) = A(z) := \sum_{i=0}^{\nu} a_i z^i$. The other two matrix polynomials $U(z) := \sum_{i=0}^{\nu-1} u_i z^i$ and $W(z) := \sum_{i=0}^{\nu} w_i z^i$ can be updated in a similar way as the updating of $X(z)$ and $A(z)$. Hence, if the difference between two successive block indices stays small, this gives a method to compute the inversion parameters $X(z), A(z), U(z)$ and $W(z)$ of the inversion formula for the block Hankel matrix $H_{\nu,\nu}$ using $O(\nu^2)$ FLOPS. The inversion formula can be applied to a given right hand side by fast polynomial multiplication methods using $O(\nu \log \nu)$ FLOPS by FFT-techniques [15]. Hence, this leads to a fast method to solve block Hankel systems of equations. However, using the inversion formula could lead to a loss of accuracy in the computed result as is the case for Toeplitz matrices. See, e.g., [17, 22]. However, the inversion formula has the advantage that it is easy to perform iterative refinement. The stability of inversion formulas for scalar Toeplitz matrices is studied in [18]. Instead of using the inversion formula, one can apply the alternative updating method of the next section. It is similar to updating methods used in the Toeplitz case where it usually leads to more accurate results than the use of the inversion formula. See, e.g., [17, 12]. In [22], this is shown for the block Toeplitz case.

12 Updating of the solution of a block Hankel system

For each block index ν, the corresponding block Hankel system of linear equations

$$H\mathbf{v} = \mathbf{y}$$

can be written as a nonhomogeneous (right) interpolation problem when v and y are interpreted as the stacking vectors of $v(z)$ and $y(z)$ respectively.

Definition 12.1 (nonhomogeneous interpolation problem) *Let $y(z) \in \mathbb{C}[z]^{p\times 1}$ and $H(z) \in \mathbb{C}[z]^{p\times p}$ be given. For each block index ν of H, we call $v(z) \in \mathbb{C}[z]^{p\times 1}$ the solution of a nonhomogeneous interpolation problem with right-hand side $y(z)$ iff*

$$H(z)v(z) = y(z) + O_-(z^{-\nu-1}) + O_+(z^0)$$

and $\deg v(z) < \nu$.

Note that the solution always exists and is unique.

We construct now the updating formulas to go from the solution $v(z)$ for block index ν to the solution $v'(z)$ for block index ν'. Let us try to find the polynomial vectors $e(z), f(z) \in \mathbb{C}[z]^{p\times 1}$ such that

$$v'(z) = v(z) - [X(z) \;\; Z(z)] \begin{bmatrix} e(z) \\ f(z) \end{bmatrix}. \tag{9}$$

Because the degree of $v'(z)$ should be smaller than ν', we limit the degrees of $E(z)$ and $F(z)$, i.e. $\deg e(z) < \delta$ and $\deg f(z) < \delta$ with $\delta = \nu' - \nu$. The nonhomogeneous interpolation conditions should be satisfied, i.e.

$$\begin{aligned} y(z) &= H(z)v'(z) + O_-(z^{-\nu'-1}) + O_+(z^0) \\ &= H(z)v(z) - H(z)[X(z) \;\; Z(z)] \begin{bmatrix} e(z) \\ f(z) \end{bmatrix} + O_-(z^{-\nu'-1}) + O_+(z^0) \end{aligned}$$

Because $H(z)v(z) = y(z) + O_-(z^{-\nu-1}) + O_+(z^0)$ and $H(z)[X(z) \;\; Z(z)] = [R_x^-(z) + R_x^+(z) \;\; R_z^-(z) + R_z^+(z)]$, we find the coefficients e_i of $E(z)$ and f_i of $F(z)$ from the following system of linear equations with the coefficient matrix $S_{\nu\to\nu'}$ as in (6)

$$S_{\nu\to\nu'} \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{\delta-1} \\ \hline f_0 \\ f_1 \\ \vdots \\ f_{\delta-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \hline r^+_{v,-\nu-1} - y_\nu \\ r^+_{v,-\nu-2} - y_{\nu+1} \\ \vdots \\ r^+_{v,-\nu'-1} - y_{\nu'} \end{bmatrix} \tag{10}$$

with $H(z)v(z) = R_v^-(z) + O_+(z^0) = r^-_{v,-1}z^{-1} + r^+_{v,-2}z^{-2} + \cdots + O_+(z^0)$ and $y(z) = y_0 + y_1 z^1 + \cdots$.

13 A look-ahead Trench algorithm

The recurrence relation (5) only needs the previous auxiliary matrix polynomial $X_{n-1}(z)$ and the first FOMP $A^1_{n-1}(z)$ to compute the next ones. If we also want to compute the inner FOMPs we can use (3). In this section, we generalize the recurrence relations used by Freund and Zha [13] where also the inner polynomials are needed in the recurrence relations.

Suppose that all blocks of FOMPs $\overline{A}_i(z)$, $i = 0, 1, \ldots, n-1$ and all auxiliary matrix polynomials $X_i(z)$, $i = 1, 2, \ldots, n-1$ have been computed. To compute $A^1_n(z)$ and $X_n(z)$, we only need the FOMPs of block $n-1$ and $X_{n-1}(z)$. Indeed, we can determine $q_a, q'_a \in \mathbb{C}^{p\alpha_n\times p}$, $q_x \in \mathbb{C}^{p\times p}$ such that

$$[A^1_n(z) \mid X_n(z)] = [\overline{A}_{n-1}(z) \mid X_{n-1}(z) \mid zA^{\alpha_n}_{n-1}(z)] \left[\begin{array}{c|c} q_a & q'_a \\ \hline q_x & 0 \\ \hline I_p & 0 \end{array}\right] \tag{11}$$

where

$$H_{\nu_n,\nu_n+1}\left[\overline{\mathrm{A}}_{n-1} \mid \mathrm{X}_{n-1} \mid Z\mathrm{A}_{n-1}^{\alpha_n}\right]\left[\begin{array}{c|c} q_a & q'_a \\ \hline q_x & 0 \\ \hline I_p & 0 \end{array}\right] = \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline \vdots & \vdots & \vdots \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 0 & I_p & r'_a \\ \hline R_{n-1} & r_x & r_a \end{array}\right]\left[\begin{array}{c|c} q_a & q'_a \\ \hline q_x & 0 \\ \hline I_p & 0 \end{array}\right]$$

$$= \left[\begin{array}{c|c} 0 & 0 \\ \hline \vdots & \vdots \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ \hline 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & I_p \end{array}\right] \qquad (12)$$

with R_{n-1} nonsingular and Z the downshift operator. Only the last $\alpha_n + 1$ block equations of (12) are needed. By using the left FOMPs $\overline{B}_i$, this can be split up as

$$\left\langle \overline{B}_{n-2}(z), \left[\overline{A}_{n-1}(z) \mid X_{n-1}(z) \mid zA_{n-1}^{\alpha_n}(z)\right]\left[\begin{array}{c} q_a \\ q_x \\ I_p \end{array}\right]\right\rangle = \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 0 & I_p & r'_a \end{array}\right]\left[\begin{array}{c} q_a \\ q_x \\ I_p \end{array}\right]$$

$$= \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ 0 \end{array}\right]$$

or $q_x = -r'_a = -\left\langle B_{n-2}^{\alpha_{n-1}}(z), zA_{n-1}^{\alpha_n}(z)\right\rangle$. Taking the formal inner product with $\overline{B}_{n-1}(z)$, we get

$$\left\langle \overline{B}_{n-1}(z), \left[\overline{A}_{n-1}(z) \mid X_{n-1}(z) \mid zA_{n-1}^{\alpha_n}(z)\right]\right\rangle \left[\begin{array}{c|c} q_a & q'_a \\ \hline q_x & 0 \\ \hline I_p & 0 \end{array}\right] = [D_{n-1} \mid 0 \mid d_a]\left[\begin{array}{c|c} q_a & q'_a \\ \hline q_x & 0 \\ \hline I_p & 0 \end{array}\right]$$

$$= \left[\begin{array}{c|c} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & I_p \end{array}\right].$$

Hence, q'_a is the last block column of $D_{n-1}^{-1} = \left\langle \overline{B}_{n-1}(z), \overline{A}_{n-1}(z)\right\rangle^{-1}$ and $q_a = -D_{n-1}^{-1}d_a = -\left\langle \overline{B}_{n-1}(z), \overline{A}_{n-1}(z)\right\rangle^{-1}\left\langle \overline{B}_{n-1}(z), zA_{n-1}^{\alpha_n}(z)\right\rangle$. Once $X_n(z)$ and $A_n^1(z)$ are computed, we can use (3) to compute the inner FOMPs. This can also be transformed using formal inner products. Hence, this is a Levinson-type method.

14 A look-ahead Schur-type algorithm

In this section, we give the recurrence formulas which allow to design a generalization of the look-ahead Schur-type algorithm of Bojanczyk and Heinig [1]. Note that instead of using (3) to compute inner FOMPs, we can use the more general form

$$A_n^{j+1}(z) = zA_n^j(z) + X_n(z)d_0^{(j)} + \sum_{i=1}^{j} A_n^i(z)y_i$$

with $y_i \in \mathbb{C}^{p\times p}$ arbitrary. In [1], this freedom is used (for $p = 1$) to compute the inner FOMPs such that

$$\overline{A}_n = \begin{bmatrix} \overline{\dot{A}}_n \\ I_{p\alpha_{n+1}} \end{bmatrix}.$$

With this additional condition on the highest degree coefficient, the (inner) FOMPs are uniquely defined. Suppose that we have already computed A_n^i, $i = 1, 2, \ldots, j-1$, then A_n^j can be computed as

$$A_n^j(z) = \left[A_n^1(z) \mid X_n(z) \mid zA_n^{j-1}(z)\right] \begin{bmatrix} q_a \\ q_x \\ I_p \end{bmatrix}$$

where $q_a, q_x \in \mathbb{C}^{p\times p}$ satisfy

$$H_{\nu_n,\nu_n+j} \left[\begin{array}{c|c|c} \dot{A}_n^1 & \mathbf{X}_n & Z\dot{A}_n^{j-1} \\ \hline I_p & 0 & a \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 0 & 0 & I_p \end{array}\right] \left[\begin{array}{c} q_a \\ \hline q_x \\ \hline I_p \end{array}\right] = \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline \vdots & \vdots & \vdots \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 0 & I_p & r_a \end{array}\right] \left[\begin{array}{c} q_a \\ \hline q_x \\ \hline I_p \end{array}\right]$$

$$= \left[\begin{array}{c} 0 \\ \hline \vdots \\ \hline 0 \\ \hline 0 \\ \vdots \\ 0 \\ 0 \end{array}\right]$$

The condition on the highest degree coefficient leads to $q_a = -a$ with a the coefficient of $A_n^{j-1}(z)$ corresponding to z^{ν_n-1}, while the condition on the residual series gives $q_x = -r_a := \left\langle I_p z^{\nu_n+j-2}, zA_n^{j-1}(z)\right\rangle$. To compute $X_n(z)$ and $A_n^1(z)$, we can use the recurrence relation (11). The same recurrence relations can also be used to compute the residual series. This leads to a Schur-type algorithm.

15 Solving the block Sylvester system

In this section, we show that solving the block Sylvester system (6) in a specific way, leads to the introduction of the inner FOMPs in the recurrence relation. We start with

$$\begin{aligned} H(z)X_n(z) &= R_x^+(z) + R_x^-(z) \\ H(z)A_n^1(z) &= R_a^+(z) + R_a^-(z). \end{aligned}$$

Let us partition the block Sylvester matrix $S_{\nu\to\nu'}$ in 4 blocks each having dimension $p\delta_n \times p\delta_n$

$$S_{\nu\to\nu'} =: \begin{bmatrix} H_1(r_x^-) & H_1(r_a^-) \\ H_2(r_x^-) & H_2(r_a^-) \end{bmatrix}.$$

To compute $X_{n'}(z)$, we have to solve

$$S_{\nu\to\nu'} \begin{bmatrix} \mathbf{C} \\ \mathbf{D} \end{bmatrix} = \begin{bmatrix} H_1(r_x^-) & H_1(r_a^-) \\ H_2(r_x^-) & H_2(r_a^-) \end{bmatrix} \begin{bmatrix} \mathbf{C} \\ \mathbf{D} \end{bmatrix} = \begin{bmatrix} 0 \\ E_{\delta_n} \end{bmatrix} \tag{13}$$

with E_{δ_n} a zero block vector except for the last component which is I_p. Using the transformation of variables

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{D} \end{bmatrix} =: \begin{bmatrix} I_{\delta_n} & -H_1(r_x^-)^{-1}H_1(r_a^-) \\ 0 & I_{\delta_n} \end{bmatrix} \begin{bmatrix} \mathbf{C}' \\ \mathbf{D}' \end{bmatrix}$$

solving (13) transforms into

$$\begin{bmatrix} H_1(r_x^-) & H_1(r_a^-) \\ H_2(r_x^-) & H_2(r_a^-) \end{bmatrix} \begin{bmatrix} I_{\delta_n} & -H_1(r_x^-)^{-1}H_1(r_a^-) \\ 0 & I_{\delta_n} \end{bmatrix} \begin{bmatrix} \mathbf{C}' \\ \mathbf{D}' \end{bmatrix} = \begin{bmatrix} 0 \\ E_{\delta_n} \end{bmatrix}$$

$$\begin{bmatrix} H_1(r_x^-) & 0 \\ H_2(r_x^-) & S \end{bmatrix} \begin{bmatrix} \mathbf{C}' \\ \mathbf{D}' \end{bmatrix} = \begin{bmatrix} 0 \\ E_{\delta_n} \end{bmatrix}$$

with S the Schur complement of $H_1(r_x^-)$ with respect to $S_{\nu\to\nu'}$. Hence, it turns out that $\mathbf{C}' = 0$ and $\mathbf{D}' = S^{-1}E_{\delta_n}$. The updating formula to derive $X_{n'}(z)$ transforms into

$$[X_n(z), \ldots, z^{\delta_n-1}X_n(z) \mid A_n^1(z), \ldots, z^{\delta_n-1}A_n^1(z)] \begin{bmatrix} I_{\delta_n} & -H_1(r_x^-)^{-1}H_1(r_a^-) \\ 0 & I_{\delta_n} \end{bmatrix} \begin{bmatrix} \mathbf{C}' \\ \mathbf{D}' \end{bmatrix}$$

$$= [X_n(z), \ldots, z^{\delta_n-1}X_n(z) \mid A_n^1(z), A_n^2(z), \ldots, A_n^{\delta_n}(z)] \begin{bmatrix} \mathbf{C}' \\ \mathbf{D}' \end{bmatrix} = X_{n'}(z)$$

where the inner FOMPs $A_n^j(z)$ are the same as given by (3). Hence, S is nothing else but R_n of (12). We get that $\mathbf{D}' = q_a'$. A similar procedure can be followed to compute $A_{n'}^1(z)$. So, we have shown that when solving the block Sylvester system in a specific way the inner FOMPs show up leading to linear systems of dimension $p\delta_n$ instead of $2p\delta_n$.

16 Conclusion

We have indicated the central role of the polynomial matrices $X_n(z)$ and $A_n^1(z)$ when computing right FOMPs, when representing all solutions of a certain homogeneous interpolation problem, as part of the parameters of an inversion formula of a block Hankel matrix or an updating formula for solving block Hankel systems.

We have given connections between the recurrences used in three look-ahead methods which appeared in the literature. It is clear that several combinations of these recurrence relations can be taken together with several look-ahead strategies to design (hopefully) weakly stable algorithms to solve block Hankel systems.

References

[1] A.W. Bojanczyk and G. Heinig. A multi-step algorithm for Hankel matrices. *Journal of Complexity*, 10(1):142–164, March 1994.

[2] C. Brezinski. *Padé-type approximants and general orthogonal polynomials*, volume 50 of *Internat. Ser. of Numer. Math.* Birkhäuser Verlag, Basel, 1980.

[3] A. Bultheel. *Laurent series and their Padé approximations*, volume OT-27 of *Oper. Theory: Adv. Appl.* Birkhäuser Verlag, Basel-Boston, 1987.

[4] A. Bultheel and M. Van Barel. Linear algebra, rational approximation and orthogonal polynomials, 1995. (In preparation).

[5] S. Cabay, A.R. Jones, and G. Labahn. A stable algorithm for multi-dimensional Padé systems and the inversion of generalized Sylvester matrices. Technical Report TR 94-07, Department of Computing Science, The University of Alberta, Edmonton, Alberta, Canada, March 1994. To appear in SIMAX.

[6] S. Cabay and R. Meleshko. A weakly stable algorithm for Padé approximants and the inversion of Hankel matrices. *SIAM J. Matrix Anal. Appl.*, 14:735–765, 1993.

[7] T. F. Chan and Per C. Hansen. A look-ahead Levinson algorithm for general Toeplitz systems. *IEEE Trans. Sig. Proc.*, 40(5):1079–1090, 1992.

[8] T. F. Chan and Per C. Hansen. A look-ahead Levinson algorithm for indefinite Toeplitz systems. *SIAM J. Matrix Anal. Appl.*, 13(2):490–506, 1992.

[9] T. Chihara. *An introduction to orthogonal polynomials.* Gordon and Breach Science Publishers, New York, 1978.

[10] A. Draux. *Polynômes orthogonaux formels – applications*, volume 974 of *Lecture Notes in Math.* Springer, Berlin, 1983.

[11] R.W. Freund. A look-ahead Bareis algorithm for general Toeplitz matrices. *Numer. Math.*, 68:35–69, August 1994. AT&T Bell Labs. Numerical Analysis Manuscript 93-11.

[12] R.W. Freund and H. Zha. Formally biorthogonal polynomials and a look-ahead Levinson algorithm for general Toeplitz systems. *Linear Algebra Appl.*, 188/189:255–303, 1993.

[13] R.W. Freund and H. Zha. A look-ahead strategy for the solution of general Hankel systems. *Numer. Math.*, 64:295–322, 1993.

[14] K. Gallivan, S. Thirumalai, and P. Van Dooren. A look-ahead Schur algorithm. In *Fifth SIAM Conference on Applied Linear Algebra, Snowbird, Utah*, pages 450–454, June 1994.

[15] K.O. Geddes, S.R. Czapor, and G. Labahn. *Algorithms for computer algebra.* Kluwer, Boston (Mass.), 1992.

[16] M.H. Gutknecht. Stable row recurrences for the Padé table and generically superfast lookahead solvers for non-Hermitian Toeplitz systems. *Linear Algebra Appl.*, 188/189:351–422, 1993.

[17] M.H. Gutknecht and M. Hochbruck. Look-ahead Levinson and Schur algorithms for non-Hermitian Toeplitz systems. Technical Report 93-11, IPS, ETH Zürich, 1993. NUMAT

[18] M.H. Gutknecht and M. Hochbruck. The stability of inversion formulas for Toeplitz matrices. Technical Report 93-13, IPS, ETH Zürich, 1993. LAA

[19] M.H. Gutknecht and M. Hochbruck. Look-ahead Levinson- and Schur-type recurrences in the Padé table. *Electron. Trans. Numer. Anal.*, 2:104–129, 1994.

[20] T. Huckle. A look-ahead algorithm for solving nonsymmetric linear Toeplitz equations. In *Fifth SIAM Conference on Applied Linear Algebra, Snowbird, Utah*, pages 455–459, June 1994.

[21] G. Labahn, D.K. Choi, and S. Cabay. The inverses of block Hankel and block Toeplitz matrices. *SIAM J. Comput.*, 19(1):98–123, 1990.

[22] M. Van Barel and A. Bultheel. A look-ahead algorithm for the solution of block Toeplitz systems. Technical Report TW224, Department of Computer Science, K.U. Leuven, March 1995. Submitted to LAA (32 pages).